NEA/CNRA/R(97)2

LICENSED OF COMPUTER-BASED SYSTEMS IMPORTANT TO SAFETY

Workshop and CNRA Special Issues
Meeting on Technical Support

APPENDIX

COMMITTEE ON NUCLEAR REGULATORY ACTIVITIES
OECD NUCLEAR ENERGY AGENCY
Le Seine Saint-Germain – 12, boulevard des Îles
F-92130 Issy-les-Moulineaux (France)
Tel. (33) 1 45 24 82 00  Fax (33) 1 45 24 11 10
LICENSING OF COMPUTER-BASED SYSTEMS IMPORTANT TO SAFETY

Workshop and CNRA Special Issues
Meeting on Technical Support
Appendix

ORGANISATION FOR ECONOMIC CO-OPERATION AND DEVELOPMENT

Paris

55022

Document incomplet sur OLIS
Incomplete document on OLIS
ORGANISATION DE COOPÉRATION ET DE DÉVELOPPEMENT ÉCONOMIQUES

En vertu de l’article 1er de la Convention signée le 14 décembre 1960, à Paris, et entrée en vigueur le 30 septembre 1961, l’Organisation de Coopération et de Développement Économiques (OCDE) a pour objectif de promouvoir des politiques visant :
— à réaliser la plus forte expansion de l’économie et de l’emploi et une progression du niveau de vie dans les pays Membres, tout en maintenant la stabilité financière, et à contribuer ainsi au développement de l’économie mondiale ;
— à contribuer à une saine expansion économique dans les pays Membres, ainsi que les pays non membres, en voie de développement économique ;
— à contribuer à l’expansion du commerce mondial sur une base multilatérale et non discriminatoire conformément aux obligations internationales.


L’AGENCE DE L’OCDE POUR L’ÉNERGIE NUCLÉAIRE


L’AEN a pour principal objectif de promouvoir la coopération entre les gouvernements de ses pays participants pour le développement de l’énergie nucléaire en tant que source d’énergie sûre, acceptable du point de vue de l’environnement, et économique.

Pour atteindre cet objectif, l’AEN :
— encourage l’harmonisation des politiques et pratiques réglementaires notamment en ce qui concerne la sûreté des installations nucléaires, la protection de l’homme contre les rayonnements ionisants et la préservation de l’environnement, la gestion des déchets radioactifs, ainsi que la responsabilité civile et l’assurance en matière nucléaire ;
— évalue la contribution de l’électronucléaire aux approvisionnements en énergie, en examinant régulièrement les aspects économiques et techniques de la croissance de l’énergie nucléaire et en établissant des prévisions concernant l’offre et la demande de services pour les différentes phases du cycle du combustible nucléaire ;
— développe les échanges d’information scientifiques et techniques notamment par l’intermédiaire de services communs ;
— met sur pied des programmes internationaux de recherche et développement, et des entreprises communes.

Pour ces activités, ainsi que pour d’autres travaux connexes, l’AEN collabore étroitement avec l’Agence Internationale de l’Énergie Atomique de Vienne, avec laquelle elle a conclu un Accord de coopération, ainsi qu’avec d’autres organisations internationales opérant dans le domaine nucléaire.

© OCDE 1997

Les permissions de reproduction partielle à usage commercial ou destinée à une formation doivent être adressées au Centre français d’exploitation du droit de copies (CFC), 20, rue des Grands-Augustins, 75006 Paris, France, pour tous les pays à l’exception des États-Unis, l’autorisation doit être obtenue du Copyright Clearance Center Inc. (CCC). Toute autre demande d’autorisation ou de traduction totale ou partielle de cette publication doit être adressée aux Éditions de l’OCDE, 2, rue André-Pascal, 75775 PARIS CEDEX 16, France.
COMMITTEE ON NUCLEAR REGULATORY ACTIVITIES

The Committee on Nuclear Regulatory Activities (CNRA) of the OECD Nuclear Energy Agency (NEA) is an international committee made up primarily of senior nuclear regulators. It was set up in 1989 as a forum for the exchange of information and experience among regulatory organisations and for the review of developments which could affect regulatory requirements.

The Committee is responsible for the programme of the NEA, concerning the regulation, licensing and inspection of nuclear installations. The Committee reviews developments which could affect regulatory requirements with the objective of providing members with an understanding of the motivation for new regulatory requirements under consideration and an opportunity to offer suggestions that might improve them or avoid disparities among Member Countries. In particular, the Committee reviews current practices and operating experience.

The Committee focuses primarily on power reactors and other nuclear installations currently being built and operated. It also may consider the regulatory implications of new designs of power reactors and other types of nuclear installations.

In implementing its programme, CNRA establishes co-operative mechanisms with NEA's Committee on the Safety of Nuclear Installations (CSNI), responsible for co-ordinating the activities of the Agency concerning the technical aspects of design, construction and operation of nuclear installations insofar as they affect the safety of such installations. It also co-operates with NEA's Committee on Radiation Protection and Public Health (CRPPH) and NEA's Radioactive Waste Management Committee (RWMC) on matters of common interest.
ABSTRACT

Both the NEA Committee on the Safety of Nuclear Installations and the Committee on Nuclear Regulatory Activities believe that an essential factor in ensuring the safety of nuclear installations is the continuing exchange and analysis of technical information and data. To facilitate this exchange the Committee has established Working Groups and Groups of Experts in specialised topics. Based on a request of the Committees, an Organising Committee of Experts on software licensing issues was established to develop a state-of-the-art view of the technological problems and methods for identifying licensing concerns, to formulate plans for an International Workshop and to prepare a state-of-the-art report on safety-critical software.

In April 1995, the Committee met and planned for a Workshop on the Technical Issues of Computer-Based Systems Important to Safety. It was the view of both the CNRA and CSNI that the workshop should include participants from all facets of nuclear industry to obtain as wide a perspective as possible. Further plans were made to incorporate the results of the Workshop into a state-of-the-art report which to be presented at a CNRA/CSNI Special Issue meeting scheduled for June 1996.

This appendix provides the workshop programme and copies of the papers and presentations, and a list of participants.
FOREWORD

The main purpose of the Workshop is to provide a forum for the exchange of information on the technical issues of computer-based systems important to safety (safety-critical and safety-related software). Participants met their counterparts from other countries and organisations and discussed current and future issues regarding the topic. Approximately one hundred (100) participants from seventeen (17) different countries attended the workshop. The countries represented at the workshop included: Belgium, Canada, Czech Republic, Finland, France, Germany, Hungary, Italy, Japan, Republic of Korea, the Netherlands, Norway, Spain, Sweden, Switzerland, the United Kingdom, and the United States.

The Workshop was held in Munich, Germany, from 5 to 7 March 1996 and was hosted by Gesellschaft für Reaktorsicherheit (GRS)/Institut für Sicherheitstechnologie (ISTec). Members of Organising Committee wish to acknowledge the excellent planning and arrangements made by the staff of GRS/ISTec.

The main purpose of the CNRA Special Issues meeting was for both CNRA members and CSNI members to come together and discuss technical support required for licensing issues of computer-based systems important to safety, utilising the results of the workshop. The meeting, which took place 18 June 1996 in Paris, included presentations by members of the Organising Committee, open discussion by CNRA and CSNI members and a final panel session.

The main presentations were as follows:

Mr. Barry Kaufer, OECD/NEA
Introduction and Background

Mr. John M. Gallagher, United States Nuclear Regulatory Commission
Licensing Issues Related to System Architecture

Mr. Pierre-Jacques Courtois, Belgium, AVN
Licensing Issues Related to Software Assessment in Testing & Inspection

Dr. Norman Ichiyen, Atomic Energy of Canada, Ltd (AECL)
Maintenance and Innovation Issues

This report was prepared under the guidance of the CNRA Organising Committee, a group of experts formed by the CNRA. The Group included (in addition to the above mentioned members): Dr. H. Ragheb (AECB), Mr. P. Joannu (Ontario Hydro), Mr. R. Virolainen (STUK), Prof. B. Wahlstrom (VTT), Mr. F. Feron (DSIN), Mr. M. Joly (EDF), Mme. B. Soubies (IPSN), Dr. W. Bastl (ISTec), Dr. T. Schaefer (BfS), Dr. D. Wach (ISTec), Mr. Z. Ogiso (NUPEC), Dr. H.-K. Shin (KAERI), Mr. G. Dahl (Halden Reactor Project), Mr. B. Liwang (SKL), Mr. U. Meyer (HSK), and Mr. N. Wainwright (NII).

The Organising Committee wishes to acknowledge the excellent guidance and support provided by the Secretariat of the OECD Nuclear Energy Agency.
Final Programme

INTERNATIONAL WORKSHOP

on

TECHNICAL SUPPORT

for

LICENSING ISSUES of COMPUTER-BASED SYSTEMS

IMPORTANT TO SAFETY

Hosted by GRS / ISTec

MUNICH, GERMANY

5 – 7 March 1996

COMMITTEE ON NUCLEAR REGULATORY ACTIVITIES (CNRA)

COMMITTEE ON THE SAFETY OF NUCLEAR INSTALLATIONS (CSNI)

Sponsor/Host

OECD NUCLEAR ENERGY AGENCY

The Committee on Nuclear Regulatory Activities (CNRA) of the OECD Nuclear Energy Agency (NEA) is an international committee made up primarily of senior nuclear regulators. It was set up in 1986 as a forum for the exchange of information and experience among regulatory organisations and for the review of developments which could affect regulatory requirements.

The Committee on the Safety of Nuclear Installations (CSNI) of the OECD Nuclear Energy Agency (NEA) is an international committee made up of senior scientists and engineers. It was set up in 1973 to develop, and co-ordinate the activities of the Nuclear Energy Agency concerning the technical aspects of the design, construction and operation of nuclear installations insofar as they affect the safety of such installations. The Committee's purpose is to foster international co-operation in nuclear safety among the OECD Member countries.

INSTITUT FÜR SICHERHEITSTECHNOLOGIE (ISTec) GMBH

The Institute for Safety Technology (ISTec) GmbH was founded in 1991 as a subsidiary of Gesellschaft für Anlagen- und Reaktorsicherheit (GRS) mbH. ISTec carries out application-oriented research, development and consulting in the areas of:

- methodologies and systems for the control of industrial processes
- qualification and quality assurance with the aim of achieving high standards of safety and operability

The main subject areas covered by ISTec are:

- diagnosis technology for industrial processes and systems
- instrumentation and control concepts
- information technology for industrial applications
- qualification and quality assurance for hard- and software
- waste management and recycling
Organising Committee

- P. J. Coutoix  AIB-Vicotte, Belgium
- N. Ichiyon  AECL, Canada
- H. Ragheb  AEGB, Canada
- V.M. Raisa  Out. Hydro, Canada
- P. Joannou  Out. Hydro, Canada
- R. Virroinen  STUK, Finland
- B. Wahlstrom  VTT, Finland
- F. Feron  DSIN, France
- M. Joly  EDF, France
- B. Scubies  IPSN, France
- W. Bauti  ISTec, Germany
- T. Schneiter  BfS, Germany
- D. Wach  ISTec, Germany
- Z. Ogiro  NUPEC, Japan
- A. J. Soehregis  ECN, Netherlands
- G. Dahl  Halden, Norway
- B. Ljung  SKI, Sweden
- U. Meyer  HSK, Switzerland
- N. Vainwright  NHL, United Kingdom
- J. Gallagher  USNRC, United States

MONDAY - 4 March

13:00 - 16:00
→ Facilitator/Recorder Training Session
ISTec Garching Offices
(Organising Committee Members)

19:00 - 20:00
→ Pre-registration Eden-Hotel-Wolff
(Hotel Reception Area)
POSTERS

➢ Hartmann & Braun
➢ TÜV Nord
➢ BfS

TUESDAY MORNING - 5 March

08:30 - 09:00 REGISTRATION - European Patentamt, Erhardstrasse 27
09:00 - 09:15 WELCOMING REMARKS
GRSISTec - Dr. Werner Bastl OECD/NEA - Mr. Barry Kaufor

REQUIREMENTS - PLANNING & SPECIFYING A SYSTEM IMPORTANT TO SAFETY
Session Chairperson - Mr. Paul Joannou, Ontario Hydro (PWGS)

09:15 - 10:15 Prof. Dr. H.-O. FISCHER, Ruhr-Universität Bochum
HOW TO PLAN AND SPECIFY A SYSTEM IMPORTANT TO SAFETY

10:15 - 10:45 COFFEE

10:45 - 11:10 Dr. Urho Pulkkinnen, VTT Automation
Programmable Automation Systems in PSA - Approaches to Reliability Modelling and Quantification

11:10 - 11:30 M. Joan-Michel Nogué, AEROSPATIALE
Qualification Process Applied to Airborne Systems Containing Software

11:30 - 11:50 Mr. Paul Joannou, Ontario Hydro
Framework for Engineering Real Time Software for Nuclear Power Plants

11:50 - 12:15 Mr. Norman Ichihara, AECL
Specification and Verification of Safety Critical Software

12:15 - 13:00 Panel Discussion - Authors of Invited Papers

7
TUESDAY AFTERNOON - 5 March

13:00 - 13:45  LUNCH

13:45 - 14:00  Mr. Bruce M. Cook, Westinghouse Electric Corporation
              Reconciling the Structured Software Design Process to Large Scale Nuclear I&C Project Schedules

14:00 - 14:15  Mr. Yoshio Yamamoto, Kawasaki Electric Power Company
              Planning & Specifying of Digital Based Reactor Protection System for next Stage PWR Plants in Japan

14:15 - 14:30  Mr. Károly Hemar, Hungarian Atomic Energy Commission
              Safety Code Requirements for Computer-Based Systems in Hungary

14:30 - 14:45  Mr. Harri Heimbürger, Finnish Centre for Radiation & Nuclear Safety
              Finnish Regulatory Requirements for Programme Computer-Based Automation (I&C) Systems

14:45 - 15:00  Mr. Ladislav Karpeta, State Office of Nuclear Safety, Czech Republic
              Licensing Requirements & Evaluation Methodology for Safety Critical Software of the NPP Temelin I&C Systems

15:00 - 15:10  Mr. Won-Young Yun, Korea Institute of Nuclear Safety
              Current Status and Licensing Experience of Computer-Based Safety Systems in Korea

15:10 - 15:20  Mr. Tan-i Chi Ogiso, Nuclear Power Engineering Corporation
              Regulatory View of Computer-Based Safety-Related Systems

15:20 - 15:30  Mr. Erik Johansson, KTH, Royal Institute of Technology, Stockholm
              Retrofitting to Programmable Electronics in Nuclear Power Plants - Focus on Requirements Aspects

15:30 - 15:45  COFFEE

15:45 - 17:00  Discussion Groups - 3 to 4 groups with facilitators

17:00 - 18:00  Closing Summaries and Panel Discussion
              Mr. Paul Joanos & Facilitators

WEDNESDAY MORNING - 6 March

DEVELOPMENT - PRODUCING A REVIEWABLE SYSTEM
Session Chairperson - Mr. Gustav Dahl, OECD Halden Project

09:00 - 10:00  Prof. David L. PARNAS, McMaster University
              HIGH INTEGRITY SOFTWARE
              What Should We Do? What Can We Do? Why Don’t We Do It?

10:00 - 10:20  M. Claude Emenjaud, Schneider Electric
              Reviewability Guidelines for Computer-Based Safety Systems

10:20 - 10:40  Mr. Takashi Mishima, Tokyo Electric Power Company
              Design Requirement and Development of Software for the Digital Safety Protection System in Kashiwazaki-Kariwa - Units 6 & 7

10:40 - 11:00  COFFEE

11:00 - 11:20  Mr. Norman Winston, Nuclear Installations Inspectorate (NII)
              Development - Producing a Reviewable System

11:20 - 11:40  Dr. Arnold Graf, Siemens-KWU
              Design for Licensibility - Teleperm XS from Siemens

11:40 - 12:00  Mr. Gustav Dahl, OECD Halden Reactor Project
              Application of Guidelines for Review of Software in a Programmable Reactor Protection System

12:00 - 13:00  Panel Discussion - Authors of Invited Papers
WEDNESDAY AFTERNOON - 6 March

13:00 - 13:45  LUNCH
13:45 - 14:00  Mr. Hiroshi Yatabe, Hitachi Ltd., Omika Works
   Development of the Digital Safety Related System
14:00 - 14:15  Mr. Olle Andersson, Forsmark Kraftgrupp AB
   Requirements for Computer Systems Important to Safety at Forsmark AB
14:15 - 14:30  Mrs. Janette Ann Baldwin, AEA Technology Winfrith
   Development of a Computer-Based Protection and Monitor Systems for Application on Nuclear
   Power Plant, Reactor Applications and Experimental Studies
14:30 - 14:40  Mr. Toshiiro Aoyagi, Japan Atomic Power Company
   Reviusable Software System of Digital Based Reactor Protection System for Next Stage PWR
   Plants in Japan
14:40 - 14:55  M. J.Y. Henry, Institut de Protection et de Sûreté Nucléaire (IPSN)
   Contribution to the Safety Assessment of I & C Software for NPPs - Application to SPIN N4
14:55 - 15:05  Mr. Fausto Zambardi, National Agency for Environmental Protection
   (ANPA)
   Trends and Position of Italian Safety Body on the Application of Computer-Based Protection
   Systems in NPPs
15:05 - 15:15  Mr. Jared S. Wermiel, U.S. Nuclear Regulatory Commission
   Digital Instrumentation and Control Systems in U. S. Nuclear Power Plants
15:15 - 15:30  COFFEE
15:30 - 17:00  Discussion Groups - 3 to 4 groups w/ facilitators
17:00 - 18:00  Closing Summaries and Panel Discussion
   Mr. Gustav Dahl and Facilitators

WEDNESDAY EVENING - 6 March

GALA DINNER
20:00
Featured Speaker
Dipl. -Ing. Günter Reichart
Head of Department, Vehicle Research
Bayerische Motorenwerke AG (BMW)
THURSDAY MORNING - 7 March

TOOL VALIDATION - MAINTENANCE & OPERATIONAL FACTORS
Session Chairperson - Dr. H. Ragheb, AECB (PWG1)

09:00 - 10:00    Prof. Nancy LEVESON, University of Washington

10:00 - 10:20    Mr. Jeffrey M. Voas, Reliable Software Technologies Corp.
Experimental Results from Applying Inverse Input Distributions to a Digital Control System for a Small Nuclear Reactor

10:20 - 10:40    Mr. Günter Göse, TÜV Nord
Tooling of Computer-Based Systems - Method, Tools and Results

10:40 - 11:00    COFFEE

11:00 - 11:20    Dr. Johannes Brummer, ISTec
Validation of Transformation Tools

11:20 - 11:40    Mr. Hiashi Funakoishi, Mitsubishi Electric Corporation
Tool Validation, Maintenance and Operational Facets of Digital Based Reactor Protection System for Next Stage PWR Plants in Japan

11:40 - 12:00    Dr. Helmy Ragheb, AECB, PWG1
Operating & Maintenance Experience with Computer-Based Systems in OECD Member Countries

12:00 - 13:00    Panel Discussion - Authors of Invited Papers

THURSDAY AFTERNOON - 7 March

13:00 - 13:45    LUNCH

13:45 - 14:00    M. Fabien Feron, Direction de la Sureté des Installations Nucléaires
Overview of N4 Series I&C System

14:00 - 14:15    Mr. Kazuhiro Tanaka, Toshiba Corporation
Development of Support Tools for the Safety Digital System

14:15 - 14:30    Mr. Pentti Haaparanta, Technical Research Centre of Finland
Validation of Programmable Automation Systems for Safety Critical Applications

14:30 - 14:45    Mr. Willi Bucher, Hartmann & Brown
Digital Neutron Flux Instrumentation

14:45 - 15:00    Mr. William D. Ghrist, Westinghouse Electric Corporation
Sizeve8 B Reactor Protection System - Software Design Goals

15:00 - 15:45    Discussion Groups - 3 to 4 groups w/ facilitators

15:45 - 16:00    COFFEE

16:00 - 16:45    Closing Summaries and Panel Discussion
Dr. Helmy Ragheb & Facilitators

16:45 - 17:00    BREAK
CLOSING SESSION - 7 March

17:00 - 18:00  CLOSING PANEL SESSION
Guest Speakers and Session Chairpersons
Moderator: Mr. John M. Gallagher, U.S. Nuclear Regulatory Commission

FRIDAY - 8 March

09:30 - 16:00
Organising Committee Meeting for 1996 CNRA Special Issue Meeting - ISTec Garching Offices
SESSION 1 - MORNING

HOW TO PLAN AND SPECIFY A SYSTEM IMPORTANT TO SAFETY - Prof. Dr. H.-D. FISCHER, Ruhr-Universität Bochum

PROGRAMMABLE AUTOMATION SYSTEMS IN PSA - APPROACHES TO RELIABILITY MODELLING AND QUANTIFICATION - Dr. Urho Pulkkinen, VTT Automation

QUALIFICATION PROCESS APPLIED TO AIRBORNE SYSTEMS - M. Jean-Michel Nogué, Aérospatiale

FRAMEWORK FOR ENGINEERING REAL TIME SOFTWARE FOR NUCLEAR POWER PLANTS - Mr. Paul Joannou, Ontario Hydro

SPECIFICATION AND VERIFICATION OF SAFETY CRITICAL SOFTWARE - Mr. Norman Ichiyen, AECL

SESSION 1 - AFTERNOON

RECONCILING THE STRUCTURED SOFTWARE DESIGN PROCESS TO LARGE SCALE NUCLEAR &C PROJECT SCHEDULES - Mr. Bruce M. Cook, Westinghouse Electric Corporation

PLANNING & SPECIFYING OF DIGITAL BASED REACTOR PROTECTION SYSTEM FOR NEXT STAGE PWR PLANTS IN JAPAN - Mr. Yoshihiro Yamamoto, Kansai Electric Power Company

REGULATORY APPROACH IN FUTURE LICENSING OF COMPUTER-BASED SYSTEMS - Mr. Károly Hamar, Hungarian Atomic Energy Commission

FINNISH REGULATORY REQUIREMENTS FOR PROGRAMMABLE COMPUTER-BASED AUTOMATION (&C) SYSTEMS - Mr. Harri Heimbürger, Finnish Centre for Radiation & Nuclear Safety

LICENSED REQUIREMENTS FOR SAFETY CRITICAL SOFTWARE OF THE NPP TEMELIN &C SYSTEMS - Mr. Ceslav Karpeta, State Office of Nuclear Safety, Czech Republic

CURRENT STATUS AND LICENSING EXPERIENCE OF COMPUTER-BASED SAFETY SYSTEMS IN KOREA - Mr. Won-Young Yun, Korea Institute of Nuclear Safety

REGULATORY ASPECT OF DIGITAL SAFETY PROTECTION SYSTEM IN JAPAN - Mr. Zen-ichi Ogiso, Nuclear Power Engineering Corporation

RETROFITTING TO PROGRAMMABLE ELECTRONICS IN NUCLEAR POWER PLANTS - REQUIREMENTS ASPECTS - Mr. Erik Johansson, KTH, Royal Institute of Technology, Stockholm
DEVELOPMENT - PRODUCING A REVIEWABLE SYSTEM

SESSION 2 - MORNING

HIGH INTEGRITY SOFTWARE - WHAT SHOULD WE DO? WHAT CAN WE DO? WHY DON'T WE DO IT? - Prof. David L. PARNAS, McMaster University

REVIEWSABILITY GUIDELINES FOR COMPUTER-BASED SAFETY SYSTEMS - M. Claude Esmenjaud, Schneider Electric

DESIGN REQUIREMENT AND DEVELOPMENT OF SOFTWARE FOR THE DIGITAL SAFETY PROTECTION SYSTEM IN KASHIWAZAKI-KARIWA - UNITS 6 & 7 - Mr. Takaki Mishima, Tokyo Electric Power Company

SAFETY CASES - PRODUCING A REVIEWABLE SYSTEM - Mr. Norman Wainwright, Nuclear Installations Inspectorate (NII)

DESIGN FOR LICENSIBILITY - TELEPERM XS FROM SIEMENS - Dr. Arnold Graf and Dr. Heinz-Wilhelm Bock, Siemens/KWU

APPLICATION OF GUIDELINES FOR REVIEW OF SOFTWARE IN A PROGRAMMABLE REACTOR PROTECTION SYSTEM - Mr. Gustav Dahl, OECD Halden Reactor Project

SESSION 2 - AFTERNOON

DEVELOPMENT OF THE DIGITAL SAFETY RELATED SYSTEM - Mr. Hiroshi Yatabe, Hitachi Ltd., Omika Works

REQUIREMENTS FOR COMPUTER SYSTEMS IMPORTANT TO SAFETY AT FORSMARK AB - Mr. Olle Andersson, Forsmark Kraftgrupp AB

DEVELOPMENT OF COMPUTER-BASED PROTECTION AND MONITORING SYSTEMS FOR APPLICATION ON NPP, REACTOR APPLICATIONS AND EXPERIMENTAL STUDIES - Mrs. Janette Ann Baldwin, AEA Technology Winfrith

REVIEWABLE SOFTWARE SYSTEM OF DIGITAL BASED REACTOR PROTECTION SYSTEM FOR NEXT STAGE PWR PLANTS IN JAPAN - Mr. Toshihiro Aoyagi, Japan Atomic Power Company

CONTRIBUTION TO THE SAFETY ASSESSMENT OF I & C SOFTWARE FOR NPPS - M. J.Y. Henry, Institut de Protection et de Sûreté Nucléaire (IPSN)

TRENDS AND POSITION OF ITALIAN SAFETY BODY ON THE APPLICATION OF COMPUTER-BASED PROTECTION SYSTEMS IN NPPS - Mr. Fausto Zambardi, National Agency for Environmental Protection (ANPA)

DIGITAL INSTRUMENTATION AND CONTROL SYSTEMS IN U. S. NUCLEAR POWER PLANTS - Mr. Jared S. Wermiel, U.S. Nuclear Regulatory Commission

TOOL VALIDATION - MAINTENANCE & OPERATIONAL FACTORS

SESSION 3 - MORNING

HIGH INTEGRITY SOFTWARE - IS IT SAFE? COULD IT BE SAFER? - Prof. Nancy LEVESON, University of Washington

EXPERIMENTAL RESULTS FROM APPLYING INVERSE INPUT DISTRIBUTIONS TO VARIOUS SOFTWARE CONTROL APPLICATIONS - Dr. Jeffrey M. Voas, Reliable Software Technologies Corp.

TESTING OF COMPUTER-BASED SYSTEMS - METHODS, TOOLS AND RESULTS - Mr. Günter Glöe, TÜV Nord

VALIDATION OF TRANSFORMATION TOOLS - Dr. Johannes Brummer, ISTec

TOOL VALIDATION, MAINTENANCE AND OPERATIONAL FACTORS OF DIGITAL BASED REACTOR PROTECTION SYSTEM FOR NEXT STAGE PWR PLANTS IN JAPAN - Mr. Hisashi Funakoshi, Mitsubishi Electric Corporation

OPERATING & MAINTENANCE EXPERIENCE WITH COMPUTER-BASED SYSTEMS - Dr. Halmy Ragheb, AECB, PWG1

SESSION 3 - AFTERNOON

TOOL VALIDATION MAINTENANCE AND OPERATIONAL FACTORS - M. Fabien Feron, Direction de la Sûreté des Installations Nucléaires

DEVELOPMENT OF SUPPORT TOOLS FOR SAFETY DIGITAL SYSTEM - Mr. Kazuhiko Tanaka, H. Yatabe, Toshiba Corporation

VALIDATION OF PROGRAMMABLE AUTOMATION SYSTEMS FOR SAFETY CRITICAL APPLICATIONS - Mr. Pentti Haapanen, Technical Research Centre of Finland

DIGITAL NEUTRON FLUX INSTRUMENTATION - Mr. Willi Bucher, Hartmann & Brown

SIZEWELL B REACTOR PROTECTION SYSTEM - SOFTWARE DESIGN GOALS - Mr. William D. Ghrist, Westinghouse Electric Corporation
SESSION 1 - MORNING

HOW TO PLAN AND SPECIFY A SYSTEM IMPORTANT TO SAFETY - Prof. Dr. H.-D. FISCHER, Ruhr-Universität Bochum

PROGRAMMABLE AUTOMATION SYSTEMS IN PSA - APPROACHES TO RELIABILITY MODELLING AND QUANTIFICATION - Dr. Urho Pulkkinen, VTT Automation

QUALIFICATION PROCESS APPLIED TO AIRBORNE SYSTEMS - M. Jean-Michel Nogué, Aérospatiale

FRAMEWORK FOR ENGINEERING REAL TIME SOFTWARE FOR NUCLEAR POWER PLANTS - Mr. Paul Joannou, Ontario Hydro

SPECIFICATION AND VERIFICATION OF SAFETY CRITICAL SOFTWARE - Mr. Norman Ichiyen, AECL
How to Plan and Specify a System Important to Safety

H. D. Fischer

Abstract
The main contributors to a proper high quality of a digital instrumentation and control system important to safety particularly in the early phases of a corresponding development are identified and described.

H. D. Fischer
Lehrstuhl für Nachrichtentechnik
Ruhr-Universität Bochum
D-44780 Bochum
GERMANY
Introduction

The system important to safety under consideration is a reactor protection system of a nuclear power plant [1]. Since corresponding functions are recently seen in a slightly broader sense we refer to such a system as to the safety instrumentation and control system (I&C) [2]. Specifically we deal with a development of a digitalized safety I&C. In this framework we treat particularly the early phases in the development, often called the phases of planning and specification. It is useful for an exhaustive description of all recommended activities to add a preparation phase prior to the planning phase. The contents of the preparation phase vary to a large extent with the pre-existing environment for such a development work. In future it is seen that the environment described hereafter already exists, so that a considerable shortening of this time interval will take place.

In favour of a short but embracing paper all activities with important influence on the quality of the product are mentioned. The interface between plant system engineers and I&C engineers is described in more detail, because this interface will contribute to common-cause failures of the I&C application software to a considerable extent as investigations of the Chalk River Labs [3] reveal.

The Preparation Phase

The preparation phase is opened by the decision in principle to work towards a digitalized safety I&C system. The first activity in the preparation phase might be a feasibility study which gives the answer to the question if the development to be considered can technically and economically be successful or not. The corresponding investigations might already go into such a detail that parts of the intended system are implemented to obtain a better estimation of the risk and of the financial budget necessary. Just the step from an analogue hard-wired safety I&C system to a digitalized software driven system requires a thorough investigation of the workload for licensing proofs of the software for the responsive system. Since the licensing requirements are by no means standardized, the extent of such a feasibility study varies from country to country. In the Federal Republic of Germany for instance a four-times redundant Core Protection System, physically separated in four rooms, was installed in the Grafenrheinfeld NPP, 1981 [4]. The computerized equipment run on-line to the plant but open-loop. The main question at that time was, could the software be written in a high-level
language or do the licensing proofs require only a low-level assembler language. The answer favoured the high-level language.

During the preparation phase the installations for a computer-aided development are made. Only such an environment promises the fulfilment of the stringent licensing requirements for digital responsive systems. The appropriate environment consists of a net of distributed workstations. In relation to the salaries for the engineers a one to one correspondence between workstations and engineers is nowadays recommended. Along with the installation of such a workstation net its security and safety must be guaranteed. A specific guideline must arrange data savings and the depository of the saved data in such a way that fire, flood or actions of third parties could not cause a total loss of data with all the dramatrical consequences on the time schedule, the financial budget and the psychological impact on the motivation of the development team.

On the distributed net of workstations the documentation tool is implemented. It consists of a powerful data-base system in combination with a desktop-publishing system. The vast amount of technical reports or organizational papers which are written during the long-term development is stored in the computers' memories and is managed by them. In order to save storage capacity the latest version of each report or paper is stored in full length together with only the first sheet of each version on which the summary of the report is written. This should facilitate to check back the cause-consequence course of the total development process. The prior versions of each report are stored on paper in the archives of the development. In this way updating of reports is done quickly and with better quality than in the past. In order to guarantee fast access to all reports a documentation system like an I&C subject classification should be implemented on the workstation net, a general I&C subject classification does not exist since it depends on the company's I&C background and environment. It should be developed together with the obligatory set of descriptors or key-words for the characterization of the contents of the various reports. Searching for one report is then easily done by using the documentation tool with descriptors specifying the contents wanted. The tool is reporting all relevant articles together with their location in the data-base and in the archives as well. In the next step the user could read the summaries of the reports found and could decide if he needs the complete article for his present work.
A reporting system should be installed by which concise progress-reports on the development process as well as the corresponding actual economic data are available at any time. For problem management later on detailed time schedules for each work-item must be provided. Therefore the basic computer-aided tools are selected and implemented on the workstation cluster.

In parallel to the installation of the appropriate working environment other activities like the selection of adequate personnel and the planning of the offices for them take place. The development team consists of both experienced engineers and young creative engineers coming directly from university. Whereas the latter are mainly working on software-related issues the former are mainly specifying and designing the safety I&C system. To enable knowledge transfer a few persons of the other category are always assigned to the respective working groups. The stimulating offices of the development team must fit to the requirements of the specific work in an optimum way. Absolute quiet rooms for those who will write computer codes and larger rooms for those who are involved in system specification and design in order to facilitate a quick solution in case of contradictory requirements.

Another main task of the preparation phase is the identification of relevant standards, rules, and guidelines, national as well as international ones. The most important international standards for the development of a digital safety I&C are the IEC 880, IEC 987, IEC 1226, and ISO 9000, part 3 [5].

During the preparation phase the goals of the development are formulated and an agreement on the goals is achieved. There are technical goals as well as quality goals. Among them could be

i. the digitalized safety I&C should be used in operating plants for backfitting and modernization as well as in new evolutionary plants,
ii. the digitalized safety I&C has to be as high reliable as the hard-wired system at least,
iii. the potential for common-cause failures has to be reduced efficiently,
iv. the electromagnetic compatibility is proven under the environmental conditions to be assured,
v. off-the-shelf products should be used rather than nuclear specific hardware,
vi. I&C application software should be portable,
The licensing proofs of the software are facilitated by choosing a development process as formalized as reasonable achievable.

The total set of goals represent the main concepts of a safety I&C development. Starting with these goals the important work-items are identified and the organizational framework grows when working-groups are assigned to the main subjects of the development. After that the global research and development (R&D) project including hardware and software resources is planned with both technical work-items and time schedules for each of them. Milestones are defined to enable easy controlling of the project's progress. Financing and the necessary payment of interest for investments are considered. These activities result in a detailed R&D proposal. The preparation phase is successfully finished when the long-standing R&D project is launched.

**The Planning and Specification Phases**

During these parts of the development adequate methodologies to assure high quality of the product are investigated and those are implemented which fit to the way of working according to the company's culture. The last statement is the more important as the existing personnel will perform the corresponding engineering once the new system will be available on the market.

The relations of the project's quality group to the company's quality division is clarified. Necessary audits even on the computer-aided documentation system and the archives of all reports are settled with the company's quality experts who provide a first quality assurance group independent of the development project. A contract with a third party is made for independent reviews of the development's technical results. This constitutes a complete independency of the company which is carrying out the development. Later on these activities result in so-called type-tests, in which the new system is proved in meeting the technical specifications mentioned in the corresponding data-sheets. For this reason, the participation of official licensing experts is highly appreciated, because in this way the information on the characteristics of the new system is obtained by these institutions as early as possible. The new system may even win broad agreement among those who are responsible for the suitability-tests carried out before the new system or parts of it will be installed on a nuclear power plant. With the third party the obligatory phase model is agreed upon. Additionally, guidelines
specific to the safety I&C development are written and discussed with the third party. They concern, for instance, the subjects given in table 1.

<table>
<thead>
<tr>
<th>planning-administrative</th>
<th>final documents acc. to phase model</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1 phase model for safety I&amp;C development</td>
<td>3.1</td>
</tr>
<tr>
<td>1.2 system quality assurance plan (WR)</td>
<td>3.2</td>
</tr>
<tr>
<td>1.3 software quality assurance plan (WR)</td>
<td>3.3 table of requirements for SW- and HW components</td>
</tr>
<tr>
<td>1.4 hardware quality assurance plan</td>
<td>3.4 table of specifications for SW components</td>
</tr>
<tr>
<td>1.5 configuration management plan</td>
<td>3.5 table of design documents for SW components</td>
</tr>
<tr>
<td>1.6 verification &amp; validation plan (WR)</td>
<td>3.6 table of implementation documents</td>
</tr>
<tr>
<td>1.7 information security and safety</td>
<td>3.7</td>
</tr>
<tr>
<td>1.8 list of open topics for hardware licensing</td>
<td>3.8 table of system integration documents (WR)</td>
</tr>
<tr>
<td>constructive</td>
<td>3.9</td>
</tr>
<tr>
<td>2.1 programming guidelines</td>
<td>3.10 documentation of the functional requirements for the safety I&amp;C</td>
</tr>
<tr>
<td>2.2 documentation guidelines</td>
<td>analytical</td>
</tr>
<tr>
<td>2.3</td>
<td>4.1 tests</td>
</tr>
<tr>
<td>2.4</td>
<td>4.2 reviews</td>
</tr>
<tr>
<td>2.5 I&amp;C formal documents</td>
<td>WR = work report</td>
</tr>
<tr>
<td>2.6 application of plant identification system (WR)</td>
<td></td>
</tr>
</tbody>
</table>

Table 1 Specific Guidelines for Safety I&C

For every fiscal year a detailed working-programme for each working-group is set up. The working-programme consists of the individual work-items together with its time-schedules. Here it is distinguished between setpoint-time, actual-time, and probable time for finishing the work-item.

For problem exchange and broad information on the progress of the project so-called development dialogues are performed every fortnight. During the dialogues open questions arise. They are written into a list of open topics together with the names of the persons who are assigned to solve the problems in a prescribed time interval. So, the fortnight dialogues also provide for an efficient control of the work on open topics. The corresponding list is audited by the company's internal quality assurance group.

All those who are responsible for the development project in the company come together about four times a year to control the technical progress and to check if the development process is on schedule. They control the resources of personnel and investment, so an effective problem management is already performed when small deviations from the time-schedule are detected. To enable a development in time, it is necessary that the work-items are described as complete as possible. The title of the work-items is by no means sufficient, the goal of the investigation and the way to be followed throughout the treatment are formulated, too. Additionally, the expected result is mentioned, for instance, a work-report, a list or a catalogue. In particular for

6 30.01.1996
the young members of the development team the outstanding experts of the company are named who are to be consulted during treatment of the work-item.

In the planning phase the computerized tools for the verification and validation phases are already designed, because the tools' implementation may last many years. Among them are so important ones like a checker for the actual response-time of automatic countermeasures of the I&C system once required or a part-task simulator for validation purposes of those I&C functions which could not be tested on-site or which are recommended to be tested at the laboratory instead on-site to reduce costs. With the part-task simulator the adequacy of I&C functions is tested or the tuning of their parameters is done. To achieve this goal small parts of the I&C model must be replaced by the new intended I&C functions. This should happen right after an I&C function has been specified by the I&C expert. So there must be an automatic device which adds the code of the new I&C function to the existing I&C model of the simulator whereas simultaneously the replaced I&C function is switched off. In this way one could handle even complicated and large I&C systems: The modularity of the whole system can much better be preserved with a software-driven system instead of a conventional hardwired system, since each individual I&C function can be implemented separately from plant input signal to output signal without increasing costs.

<table>
<thead>
<tr>
<th>symbol</th>
<th>meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="Symbol 1" /></td>
<td>( u = Ri )</td>
</tr>
<tr>
<td><img src="image2" alt="Symbol 2" /></td>
<td>( i = C \frac{du}{dt} )</td>
</tr>
<tr>
<td><img src="image3" alt="Symbol 3" /></td>
<td>( u = L \frac{di}{dt} )</td>
</tr>
<tr>
<td><img src="image4" alt="Symbol 4" /></td>
<td>( u = e )</td>
</tr>
</tbody>
</table>

Fig. 1 Graphical Specification Language in Electrical Engineering
For specification of the I&C functions the I&C experts use a computer-aided specification and Coding Environment (SPACE). This tool provides a formalized visual language - like that of an electrical engineer (fig. 1) - whose syntax and semantics are easily learned because they are domain-specific and are oriented at the way in which safety I&C systems have been specified so far. Thus, the language is already known to the company's I&C experts as well as to the plant system engineers. This facilitates precise understanding when members of both groups talk together. The specification of each I&C function is done with SPACE using about 50 different elementary blocks of prescribed functionality (fig. 2). Since each block provides elementary functionality the corresponding code consists of a few statements only. In consequence, the code can be verified completely. The syntax consists mainly of a single rule: only signals of the same type (binary on/off or analogue) are to be connected. Connections are done in a CAD manner: use a mouse and pull it from source to sink!
Fig. 2 Basics for SPACE

With SPACE a formal specification method of broadest scale is applied with reusable software elements. It is a very effective approach for avoiding software errors due to wrong interpretation of the I&C task specification prepared by the plant system engineers. It seems to be the best way to systematize the mental process of the software developers and to free them from the effects of changing subjective dispositions. At the same time, this method allows the software production process to be expediently divided into discrete steps including stepwise
verification of the corresponding outcome. That is why this method generates software which is test-friendly to a large extent. The procedure model for generation of the I&C application software is given in fig. 3.

![Diagram of Procedure Model for Software Production]

Fig. 3 Procedure Model for Software Production

The I&C functional requirements are drawn up mainly by mechanical engineers, process engineers and physicists - as usual - in the form of plain text, diagrams, equations or tables. After quality assurance, the functional requirements for the digital safety I&C system are then passed on to the instrumentation and control experts in form of separated I&C functions, a prerequisite for a modularized system.

These instrumentation and control experts - comprising control engineers, communication engineers, physicists, data processing experts and mathematicians - read and digest the I&C functions. Any questions they may have as to how functional requirements are to be interpreted are discussed in so-called "system clarification meetings". The instrumentation and control experts then prepare all the data and information in a formal and thus always identical manner in order to be able to draw up the software specification. This work is performed graphically with the aid of SPACE in a form
- which is rigorously formalized
- which can be read by the functional requirement team for verification purposes and
- which also serves as documentation for purchasers.

In Germany the specification follows a format which is recommended by the German power plant owners' association.
When the specification has been verified by the functional requirement team, it is automatically prepared for code generation. This preparation step involves converting the two-dimensional graphic format into a sequential format. The output from this sequentialization step is already a high level language code. Because of the rule-based generation, this code can also be verified automatically against the specification by an independent verification tool. The achievable depth (verification quality) thereby is much greater than for any manual checking method.

This method affords the indisputable advantage that the requirements for qualification of the tools used for graphic specification, sequentialization and generation of the code can be kept quite low.

The procedure just described is one major step towards the goal to reduce the probability of common-cause failure to an extent which is as low as reasonably achievable. A frequent cause of common-mode failures are errors in the I&C functional requirements. This is not a specific feature of digital systems but applies just as much to conventional hard-wired systems. For this reason, no further action is necessary over and above the intensive quality assurance efforts that are already standard practice. The accident sequence calculations which produce the input used as the basis for the functional requirements are already performed by a different group of people from those who formulate the functional requirements themselves. There is a natural interface between these two groups which is utilized for a verification step. With a view to improving the efficiency and quality of this work, the various documentation is now produced using computer-aided systems together with a more systematic approach as in the past. Five levels of documents have been defined for classification and verification of the I&C functions requirements.

The most superior goal is to protect men and environment against nuclear radiation. This goal is achieved if so-called safety goals are assured. Their number differs from the grade of detailing. To prove the completeness of the defined I&C functions, they all are attached to the set of safety goals. Each safety goal should be assured by at least two functionally diverse I&C functions. The resulting graphical presentation is characterized as the first level of document for the I&C functional requirements. The instrument of functionally diverse I&C tasks is now being used more than ever before. Functionally diverse I&C functions represent a natural way to prevent the coincidence of identical data constellations at the same time in independent computers which, as all who are concerned agree, is the most frequent condition for common-
cause failures in computers. Thus, functional diversity is the most effective way to cope with software errors.

The second level of document only consists of an ordered list of all I&C functions, which contains the name, the identifier and the category according to the I&C functional classification scheme in each row. For the safety system of an evolutionary PWR there exist about 180 different I&C functions.

The I&C functions are described for the purpose of specification by the I&C experts in the third level of document. For each I&C function only two pages are provided. On one page
i) the functional task description,
ii) the category according to the I&C functional classification scheme,
iii) the required maximum response-time,
iv) the relevant events on which the I&C function responds,
v) the necessary independency of other I&C functions
are given.

A signal-flow diagram is drawn on the second page; the diagram visualizes the processing of input signals to output signals.

With the third level of document the input for the I&C specification is completely provided. The I&C expert can specify the I&C structure, while he is comparing the required response-times with their actual measured values. He can distribute the different I&C functions to the various computers, while he is observing the independency requirements together with application of the deterministic failure criterion resulting from rules, guidelines or standards. The I&C expert can comprehend the intended performance of each I&C function by analysing the corresponding signal-flow diagram, which is part of his technical language, too. The plant system engineer can explain the function to the I&C engineer by using the signal-flow diagram.

The fourth level of document gives a detailed reasoning for the definition of each I&C function, while the calculations for the postulated initiating events are evaluated.

The fifth level of document provides the proofs, that all relevant events - disturbances, incidents, and beyond design basis accidents - are positively controlled by the set of all I&C functions. Thus, the proofs are carried out with an event-oriented approach, whereas a
symptom-oriented approach is followed for the definition of the I&C functions. By this methodology of applying two complementary approaches the probability of common-cause failures in the I&C functional requirements can be reduced to a sufficient extent. Particularly, for nuclear developing countries it seems to be more adequate to use the event-oriented approach for the definition of I&C functions and the symptom-oriented approach for their verification.

Conclusions
The main contributors to obtain proper high quality of a digital safety I&C reach far back to the beginning of the development process. If no or only small experience with responsive systems is available then it is highly recommended to add a so-called preparation phase prior to the development itself. Such a time interval should be used for installation of an appropriate infrastructure which fits on the goals of the intended development in an optimum manner. The infrastructure does not only consist of a workstation net but of all things which guarantee an efficient development later on both with highest quality of the product and in good time. The latter decides on the economic success of the product nearly to the same extent as the proven high quality does. In the planning and specification phase emphasis is laid on the correct arrangement of the interface between the two groups which work together in a safety I&C development: the plant system engineering group and the safety I&C engineering group. In order to avoid misinterpretation of the I&C functional requirements by the I&C experts a graphical language for the specification of the I&C functions is chosen which is already used by both groups for the design of the conventional hard-wired reactor protection system. This leads directly to domain specific software with reusable I&C application software in a strong modular manner. Since this graphical specification language is familiar to both groups of engineers, the functional specification of the safety I&C is most suited for verification by the plant system engineers. Thus, the potential of common-cause failures for this part of the development is the same as that in case of a hard-wired system. The great extent of computerized tools and the efficient documentation and reporting system assure the participation of several senior engineers in the development. Just this represents a certain diversity in experts compared to times gone away where nearly a very small group of engineers were involved in such a task. The use of computer aid not only for I&C specification and for automated code generation but also for verification purposes together with a sufficiently redundant topology of the multi-computer system will reach reliability figures still better than for conventional hard-wired reactor protection systems.
Acknowledgement

Many thanks to Siemens KWU for the kindness to use some figures concerning their digital safety I&C development.

References


IEC 1226, Power Plants - Instrumentation and control systems important for safety-classification, first edition 1993-05
ISO 9000, part 3, Guidelines for the application of ISO 9001 to the development, supply and maintenance of software, 1990-08
PROGRAMMABLE AUTOMATION SYSTEMS IN PSA
- approaches to reliability modelling and quantification

Urho Pulkkinen
VTT Automation
P.O.Box 1301
FIN-02044 VTT, Finland

ABSTRACT

Safety authorities often require plant specific PSAs, and quantitative safety goals are set on different levels. The reliability analysis is more problematic when critical safety functions are realized by applying programmable automation systems. Conventional modeling techniques do not necessarily apply to the analysis of these systems, and the quantification seems to be impossible. However, it is important to analyze contribution of programmable automation systems to the plant safety and PSA is the only method with system analytical view over the safety.

The paper discusses the applicability of PSA methods (fault tree analyses, failure modes and effects analyses) in the analysis of programmable automation systems. The problem of how to decompose programmable automation systems for reliability modeling purposes is discussed. In addition to the qualitative analysis and structural reliability modeling issues, the possibility to evaluate failure probabilities of programmable automation systems is considered. One solution to the quantification issue is the use of expert judgements, and the principles to apply expert judgements is discussed in the paper. A framework to apply expert judgements is outlined. Further, the impacts of subjective estimates on the interpretation of PSA results are discussed.
1 INTRODUCTION

Safety authorities require plant specific PSAs in many countries, and quantitative safety goals are set either on core melt frequency or safety function reliability level. The reliability modeling gets difficult when critical safety functions are realized by applying programmable automation systems. Conventional modeling techniques do not necessarily apply to the analysis of these systems, and the quantification seems to be impossible. However, it is important to analyze contribution of programmable automation systems to the plant safety and PSA is the only method with system analytical view over the safety.

The paper discusses the applicability of PSA methods (fault tree analyses, failure modes and effects analyses) in the analysis of programmable automation systems. The problem of how to decompose programmable automation systems for reliability modeling purposes is also dealt with. In addition to the qualitative analysis and structural reliability modeling issues, the possibility to evaluate failure probabilities of programmable automation systems is discussed. Due to lack of operational experience and due to the nature of software faults, the conventional reliability estimation methods can not be applied. One solution to the quantification issue is the use of expert judgements, and the principles to apply expert judgements is discussed in the paper. A framework to apply expert judgements is outlined. Further, the impacts of subjective estimates on the interpretation of PSA results are discussed.

2 PROBABILISTIC SAFETY ASSESSMENT (PSA)

The purpose of probabilistic risk assessment (PSA) is to give an overall view on plant safety by identifying the initiating events of accidents, describe the event sequences beginning from initiating events and leading to various plant damage states and radioactive releases. Further, PSA evaluates the plant risk quantitatively in terms of accident frequencies. PSA is usually divided in three levels. The level 1 PSA concentrates on the reliability modeling of emergency functions and accident sequence models, level 2 PSA considers the physical accident processes and the source term, and level 3 PSA evaluates the effects of radioactive releases on the environment. The focus of this paper is at level 1 PSA.

PSAs are used for several purposes. The can be applied to compare different designs from the safety point of view as well as to evaluate design modifications and backfittings. Further, they may be used to develop and evaluate procedures, such as safety related technical specifications and emergency procedures. PSA models can also be used as a common model between plant designers and users as well as between power companies and safety authorities.

On possibility to use PSA in licensing is to pally quantitative safety guidelines or safety goals, which approach has been adopted by some safety authorities. Safety goals are rules which specify acceptable risk for example by setting a limit for core damage frequency or safety function failure probability. If safety goals are set, then all types of hazardous events must be modeled and their probability must be estimated in order to check whether the requirements of the safety goal are met.

A recent way to apply PSA is so called living PSA, which means that the safety of the plant is monitored continuously by PSA. At the same time the PSA models are updated by using the information obtained from the plant.

PSAs are based on models. The reliability structure of the emergency functions are described by applying fault trees, which explain the failures of emergency functions as consequences of component failures, human errors and other basic events. The propagation of accidents beginning from initiating events and ending with various core damage states is described by event trees. The accident sequence frequencies are estimated by using statistical data on the basic event probabilities and initiating event frequencies. The fault tree and event tree models on one hand and statistical estimates for the basic event probabilities on another are the most important models of PSA level 1.
Conventional safety automation systems are usually included in PSA models. The degree of detail of the automation system models depends on the applications of PSA, and in this respect PSAs differ much from each other. Although the fault trees of automation systems may be rather extensive there are no philosophical difficulties in the modeling process: the automation systems do not differ from the other hardware systems.

The failures of programmable automation systems (as conventional automation systems or other hardware systems and human errors) are connected to accidents in several ways. They may cause initiating events and even so called common cause initiators, which cause both initiating event and simultaneous unavailability of redundant safety functions. On another hand they can be system and component failures, which are intermediate or basic events of the fault tree models. These events cause the unavailability of safety functions or their components. This failures may be also latent, i.e. they have actually occurred in past before initiating event. As the failures of hardware components, also the failures of programmable systems may be dependent on each other. The common cause failures (CCFs) of programmable systems form a very difficult problem area.

The difficulties to include programmable automation into PSA.-models are connected with the decomposition of the systems structure and with the problems of determining quantitative reliability estimates. The components or subfunctions of a programmable systems are not easily identified or described in such a way that they could be described easily as events of fault tree. Also the dynamic features of programmable functions cause problems. The quantitative reliability estimates needed in PSA, e.g. the probability of failure to operate properly when demanded, are not results of traditional software reliability models. In order to estimate these quantitative estimates statistically a large number of tests is needed.

In the following the possibilities to identify the failures of programmable systems by applying more or less conditional reliability engineering methods are discussed. Later, the reliability modeling and quantitative analysis are considered.

3 QUALITATIVE RELIABILITY ANALYSES AND RELIABILITY MODELS

Qualitative reliability analysis methods are applicable in identifying the failure modes of the system. They provide information useful for developing reliability modes and documenting the systems failure behavior. One of the most wellknown reliability engineering method is the failure mode and effects analysis (FMEA), which is a standardized method (see IEC 812 (1985)). It has been modified for the analysis of software by Reifer (1979) and developed further for example by Darricau and Hourtolle (1992). Software FMEA has been applied mainly in nonnuclear industry, e.g. in space applications. FMEA is used to identify the component failures, their causes and effects on the systems function. FMEA has been mainly applied to hardware systems, but it is applicable for the analysis of for programmable systems. To perform FMEA, the systems function must be described first by identifying the functions of the component. The objective of the analysis is to identify potential failures in the components, and the consequences these may have on the system. In this way it is possible to identify the potential failures which are most safety critical, and therefore require the most attention. FMEA is a bottom up analysis, starting with the components, and gradually increasing the scope to analyze total functions.

The application of FMEA for analysis of programmable systems requires some modifications. The modified FMEA form includes in addition to the usual information a more exact description of the failure mode, which may be divided into hardware and software failure modes. The hardware failures are described separately from software failure modes in order to emphasize the different nature of the software faults. The dynamic software functions require more attention, and it is not clear whether FMEA can really identify all failures.
It is not practical to apply FMEA to the analysis of software structures. The analysis is most effective on the level of functions of the different modules of software based systems. These modules may not be identified without preliminary fault tree analysis, which suggest that it is advantageous to apply it iteratively with reliability modeling. This kind of iterative approach degrease also the resources needed in the analysis.

Sneak Circuit Analysis, SCA (Taylor (1992)), is a combination of FMEA-type approach and fault tree. In SCA the output of the software, which may lead to hazardous events are identified. Further the causes of errors are studied. The findings of the identification are described by fault trees.

Methods like Preliminary Hazard Analysis, PHA, Software Requirements Hazard Analysis, SRHA, Subsystem Hazard Analysis, SSHA, System Hazard Analysis, SHA, and Operating & Support Hazard Analysis, O&SHA, are based on the conventional hazard analysis methods (see McKinlay (1991)). The method developed by Toola.(1992) for the analysis of accidents and disturbances of process automation belongs to the same family of methods. The above methods have been applied in nonnuclear industry, where extensive risk analyses are not usually made. In fact, it is likely that the analyses made in connection to plant specific PSA provide the same information as the above mentioned methods. However, these approaches reveal weaknesses in the interface between the software and the other parts of the system (see Pulkkinen, 1993).

In PSA environment, FMEA and other qualitative analyses are applied in order to form a basis for fault tree models. Further, it is a documentation of identified failure modes which are included into fault trees as basic events or fault tree gates. In the case of conventional systems, it is rather easy to identify the failure modes at each level of functionality (i.e. at system, subsystem, and component levels). This means that the conventional hardware systems may be decomposed into components or subsystems with respect to the failure effects. This is more difficult in the case of programmable systems. From the hardware point of view, it is rather easy to describe the failures of each electrical unit and to identify the consequences of failures. However, the application of software based systems makes it possible to integrate many functions into one electrical unit (e.g. CPU), and the functionality of software doesn't correspond directly to the hardware units.

When a programmable system is decomposed for reliability modeling purposes, it should be done with respect to the systems emergency functions. This means that such failure modes, which cause the loss of the emergency functions are included into the model. Problems may arise when the system consists of redundant channels. Usually redundant channels can be seen as independent and they are modeled as inputs to an "AND"-gate in the fault tree. This requires that one must be able to judge how independent the channels really are. If the channels are dependent, they may utilize common processors or common service functions. In principle this kind on dependencies can be described by fault tree models: they correspond to the shared equipment (or shared function) dependencies usually encountered in PSA-modeling. The other dependencies, i.e. common cause failures caused by simultaneous failures of either software located in redundant channels or hardware can only be modeled as parametric common cause failures, i.e. they are taken into account only as parameters, which increase the probability of multiple failures.

There are many functionalities, which are specific to programmable systems. These systems may utilize extensively selfdiagnostic functions and "software based redundancy". The latter means that some functions (within a redundant channel) are made more reliable by implementing them with several redundant software units, which perform more or less independently the same task. Both of the above functionalities increase the reliability, but it is not clear how this can be modeled in fault trees. However, since this redundancy is implemented in a special or even dynamic way, it seems feasible not to describe it in fault trees. From the practical point of view, the modeling of this redundancy would make the fault trees too complicated. From more theoretical point of view, it is not possible to model dynamic features of software by static models, such as fault trees. The same principles apply partly also for the description of the diagnostic functions.
The software of programmable systems may be divided to system software and application software. The system software includes for example the operating system, and the application software is designed to produce the functions of the system. In PSA, the most important issue is to model the loss of emergency functions and its dependency on component failures. At the level, which is important for PSA, high degree of detail of the programmable systems is not needed. The division to system and application software is not always needed from that point of view. As a conclusion, it is recommendable to model only the most important functions of the programmable systems.

4 QUANTITATIVE RELIABILITY ANALYSES

If the reliability structure of a programmable system is established and described by a fault tree model, it is possible to determine the minimal cut sets and thus find the most vulnerable points of the system. The information contained by the minimal cutset list is most useful: it gives a qualitative view over system in a systematic way and it gives guidelines to direct the more extensive analyses to most critical issues.

However, in many cases it is advantageous to evaluate quantitatively the systems reliability. This is necessary, when the safety authority has set quantitative safety goals. Further, it is often important to know what is the importance of certain programmable systems for the accident frequency. It is worth noticing that this may imply also that one must determine the failure rate of such programmable automation systems, which are not emergency systems but the failures of which may cause initiating events.

The reliability models of PSA require various types of statistical reliability data. Depending on the system and its components estimates for operating time failure rates, standby time failure rates, repair times, component unavailabilities, probabilities to fail at demand or parameters of common cause failure models are needed. Often the evidence from operational experience is not sufficient for the statistical estimation of the above mentioned parameters, even in the case of commonly used components. This is a problem especially when the PSA is required to correspond the plant specific operational experience. To solve the problems due to lack of reliability data, many approaches are applied. One possibility is to used generic data bases, which may not actually correspond to the plant in question. Another way is to use expert judgements, for which purpose there are methods. Independently from the source of reliability data, PSA-methodology makes it possible to analyze the uncertainty of the risk estimates cause by the uncertainty of the reliability data and other parameters. The possibility of consistent uncertainty analyses makes PSA transparent also in this respect.

The reliability data for programmable components seems at the first sight rather similar to that of conventional systems: the problem seems to be only on the sufficiency of statistical evidence. However, this is not the whole truth. Some of the above mentioned reliability parameters may not be relevant for programmable systems. For example, the probability of failure when demanded is a common parameter for the reliability of emergency system. In the case of conventional system, the demand is exactly defined for example as manual start signal. The programmable systems operate cyclically, and they measure their inputs at certain time points. If there is a demand, i.e. the input corresponding to manual trip signal is true, the system should operate. However, at each time point where the system reads its inputs there is also a demand, which requires certain function. One may ask is the definition of the event "failure at demand" really similar for programmable system and conventional system. Same kind of problems may be actual also for other parameters.

Another issue, which must be noticed is that the reliability of programmable systems is a property of the systems operation environment as well as that of the system itself. Although there are errors in the software, these errors may cause a loss of safety function only in the case of inputs which occur with very low probability during demands. In other words, the reliability of programmable systems depend on the operational profile, which as the probability distribution of input sequences, varies from one
environment to another. This restricts the use of generic operational experience in determination of reliability parameters.

Quantitative reliability estimates are always based on certain evidence, which is most often operational experience statistics. For programmable systems this evidence is either very limited or not applicable due to differences between the operational profiles of the data source and the actual system. Another source of evidence is obtained from the dynamic testing of the system. If high reliability is required, the number of tests is very large, and the it may be practically impossible to test the system so extensively. Thus evidence from other sources must be used in order to estimate the reliability parameters.

The statistical information gathered during the development process of programmable systems is related with the systems reliability, and in principle it may be applied in reliability estimation. In order to use this information, software reliability growth models can be applied. However, it is not directly possible to evaluate the weight these measurements with respect to the PSA estimates.

The abovementioned development process follows certain quality assurance and quality control principles, which are based on applicable standards, but which may vary from one developer to another. More strict principles are believed to result to more reliable products. Thus the quality assurance process provides evidence on reliability. Some may apply to other tools and principles followed during the development process. To use this kind of evidence in determining the reliability estimates requires models and ability to weight the evidence.

The use of several kinds of evidence in determining reliability estimates is possible only through expert judgements, since validated models and extensive empirical observations are not available. As already mentioned, experts judgements are almost routinely applied in PSA to determine for example human error probabilities or describing uncertainty on some critical physical phenomena. The methodology for use of expert judgements is, however, under development and research on this topic is still going on. A framework for applying expert judgements in quantitative reliability analysis of programmable systems is outlined in the following section (see e.g. Pulkinen, 1994, Cooke, 1991).

Since expert judgements are based partly subjective assessments and weightings, it is not possible to interpret the results of PSA as objective. Thus the subjective interpretation of probability must be accepted. The implications of using subjective probabilities are discussed also in the following section.

5 USE OF EXPERT JUDGEMENTS IN QUANTITATIVE RELIABILITY ANALYSIS

To obtain better estimates for reliability of programmable systems, all possible evidence should be applied in the analysis. As discussed earlier, this require extensive applications of experts opinions about the weight of various pieces of evidence. A most suitable approach for using analyzing experts judgements is based on Bayesian models. To outline and illustrate such model, the failure probability (θ) of a programmable system is considered. It is assumed that θ depends on certain factors, which are denoted by \( X = (X_1, ..., X_n) \), in a stochastic way. The exact dependence on these factors is unknown, and it is modeled by a conditional distribution of θ given X, or \( p(\theta | X) \). Another possibility to describe the relation between θ and X is to model θ and a function of X and random disturbance or noise. This kind of model leads, however to the conditional distribution \( p(\theta | X) \). The stochastic nature of the relation between θ and X corresponds to the fact that the same development procedures do not lead automatically to same reliability. The factors \((X_1, ..., X_n)\) describe for example the development process, quality control principles, etc. Part of the factors may be directly observable and measurable, part of the may be unobservable and qualitative characterization of the system and its development process.
The model relating $\theta$ and $X$ is usually parameterized, i.e., the distribution $p(\theta \mid X)$ depends on some parameters

$$p(\theta \mid X) = p(\theta \mid X, \alpha),$$  \hspace{1cm} (1)

in which $\alpha$ is a parameter vector. The form of the model (1) and the interpretation of the parameters should be such that experts can give their assessments with respect to the model. Simple and transparent model is for example the logistic regression

$$\ln \left( \frac{\theta}{1 - \theta} \right) = \alpha_0 + \sum_{i=1}^{n} \alpha_i x_i + \omega,$$  \hspace{1cm} (2)

in which $\omega$ is a random (unknown) noise. It is possible to deduce the conditional distribution (1) from (2), when the statistical properties of the noise term are modeled by suitable distribution.

The experts are asked to evaluate the parameters ($\alpha$), and if necessary, the factors ($X_1,...,X_n$). These assessments form the evidence obtained from expert judgements. The assessments concerning $X$, are denoted by $Z_i, i=1,...,m$ and the assessments concerning $\alpha$ by $Y_i, i=1,...,m$, in which $m$ refers to number of experts. In addition to the experts evidence, there may be results from the testing of the system, for example in the for “$k$ failures occurred in $r$ tests”.

These evidence form tests and expert judgements can be combined by applying a Bayesian model. To do this, one has to specify the conditional probability of having $k$ failures in $r$ test given $\theta$ and the conditional (joint) distributions of $Z_i$ and $Y_i$ given $X$ and $\alpha$. The interpretation of these distributions is important. They are models of experts’ ability to produce estimates for unknown variables. They should be as simple as possible to keep the model transparent. Models like the additive or multiplicative error models discussed by Mosleh and Apostolakis (1988) are suitable for this purpose. The whole expert judgement model can be expressed as Bayesian network. For the above simple model the Bayesian network has the form presented in Figure 1.

Figure 1 specifies the joint distribution of all variables of the model, and describes the dependencies between the variables. When the test results and experts’ judgements become available, it is possible to determine the posterior distribution of $\theta$. The posterior distribution, $p(\theta(\mid k,r), Z_i, Y_i, i=1,...,m)$ combines the whole evidence on $\theta$ together. The numerical procedures for calculation of posterior distribution can be solved for example by applying Monte-Carlo methods (see Pulkkinen, 1994).

Generally, it is possible to combine various assessments of the system reliability to obtain better reliability estimates. The model structure depends on the reliability parameter and the form of evidence. The Bayesian approach provide a transparent way to model and determine the distributions of unknown variables. The modeling task is without doubt very difficult, and one must be aware of the nature of interpretation of the model predictions.

When the reliability parameters for all events corresponding to programmable systems are determined, they can be used as input data for fault tree calculations. Since the Bayesian approach automatically leads to distributions of parameters instead of point estimates, the uncertainty of the estimates is expressed consistently. The posterior distributions can also be applied in the uncertainty analyses of PSA-models.
Figure 1. The Bayesian network model for using expert judgement.

The reliability estimates based on expert judgements are always subjective, which may restrict the use of PSA in licensing and decision making. As subjective assessments of plant safety, the results of PSA express the uncertainty of the PSA team about the accidents and their consequences. The probabilities and accident frequencies are rather analysts degree of beliefs on the plant that real properties of the plant itself. If the assumptions and models are realistic, not conservative, and the uncertainty is expressed in a proper way, then the results provide guidance for decision making. They help in identification of the most important accident contributors, and both the safety authority and the licensee can base their decisions on these results.

The PSA results reflect the modeling assumptions and the strength of evidence behind the risk estimates. Given that the assumptions and evidence is in some sense essentially subjective, it is not recommendable to base the licensing decisions totally on the quantitative PSA results. A more proper and practical approach is to consider whether the modeling assumptions are realistic and whether the different sources of evidence are weighted in acceptable way, which implies also that the it is evaluated whether the uncertainty about different phenomena is taken into account.

The quantitative safety goals have specific role when subjective (Bayesian) interpretation of PSA is followed. When the analyst try to meet the safety requirement, they at the same time try to find such assumptions or such pieces of evidence which lead to results compatible with safety goals. The next step is to analyze if these assumptions are credible or realistic in the case of the plant under analysis.

6 CONCLUSIONS

This paper considers the possibilities to include programmable systems into PSA models. Although the programmable systems are in many respects fundamentally different from conventional systems, it seems feasible to analyze them by usual PSA methods. Qualitative methods, suchs as FMEA, are useful in identifying the failure modes of programmable system. However, it is not advantageous to go into too deep detail. It is not probable that the dynamic aspects of software based systems are tackled by such, basically static analysis methods. Similarly, it is possible to create fault trees describing programmable systems, but is not possible to model them in deep detail. A suitable decomposition seems to be at electrical unit level. this means that, for example, redundant channels of emergency signals may be modeled.
The qualitative results of reliability models and qualitative analysis are included in the minimal cutset list, which actually identifies the weak point of the system with respect to system failure of accident sequence. This list helps in identifying the features of the systems, for which deeper analyses are needed. These deeper analyses should be made by applying methods, which are more suitable for programmable systems (i.e. formal methods, etc.).

The quantitative analysis requires reliability data for components, which usually do not exist in experimental of statistical form for programmable systems. In estimation of reliability, evidence from systems test and expert judgements concerning the system and its design must be used. To combine this evidence, the application of Bayesian methods seems to be a feasible approach. However, it leads to subjective interpretation or PSA results, which has implications on the use of PSA in decision making and licensing. Keeping in mind the limitations of PSA methodology, the interpretation of more or less subjective results and the special features of programmable system, it is useful to apply PSA although part of the plants emergency systems are based on programmable technology.

REFERENCES


QUALIFICATION PROCESS APPLIED TO AIRBORNE SYSTEMS

JEAN MICHEL NOGUÉ  Aerospatiale

☎ 33 61 93 57 95  FAX: 33 61 93 80 90
E-mail: jean-michel.nogue@avions.aerospatiale.fr
1 GUIDANCE DOCUMENTS vs PROCESSES

2 AIRCRAFT, SYSTEM SAFETY ASSESSMENT PROCESS

3 SYSTEM DEVELOPMENT PROCESSES

4 SOFTWARE vs SYSTEM / SAFETY PROCESSES

5 ED12B / DO178B FEATURES
GUIDANCE DOCUMENTS vs PROCESSES

System Development Processes
(ARP 4754)

Hardware Development
Life-Cycle
(DO-178B)

Software Development
Life-Cycle
(Do-178B)

Safety Assessment Process
Guidelines & Methods
(ARP 4761)

Function, Failure
& Safety Information

System Design

Intended Aircraft
Function

Aircraft System Development Process

Hardware Life-Cycle Process

Software Life-Cycle Process

Implementation

Functional System

Certification Guidance Documents Covering System, Safety, Software and Hardware Processes
HAZARDS CONSIDERED:

* EXTERNAL HAZARDS:
  - Natural external environment: rain, lightning, wind, birds...
  - Other aircrafts
  - High Intensity Radiated Fields
  -

* INTERNAL HAZARDS
  - Functions failures
  - Hardware and Software design errors
  - Installation aspects
  - Maintenance, operational procedures errors
  - Human Factors
  -
AIRCRAFT, SYSTEM SAFETY ASSESSMENT PROCESS

Relationships with System Development Process

Qualification of Software in Airborne Systems

© AEROSPATIALE 1996
SOFTWARE vs SYSTEM / SAFETY PROCESSES

SYSTEM LIFE CYCLE PROCESSES

- System Safety Assessment Process
- System Requirements Allocated to Software
- Software Level(s)
- Design Constraints
- Hardware Definition
- Fault Containment Boundaries
- Error Sources Identified/Eliminated
- Software Requirements & Architecture

SOFTWARE LIFE CYCLE PROCESSES
MAIN CHARACTERISTICS

* "Product in a Process" - oriented document

* Guidelines for Confidence Level Assessment (no quantitative aspects)

* "Processes objectives" - oriented (no technique, no organisation)

* All life cycle allowed ("V-model", Prototyping, Re-use...)

* Experience based document
SPECIAL TOPICS

* User modifiable software

* Commercial Off The Shelf Software (COTS)

* Tool Qualification

* Service History

* Specific techniques: Formal methods, Reliability Models
Table A-7
Verification Of Verification Process Results

<table>
<thead>
<tr>
<th>Objective</th>
<th>Applicability by SW Level</th>
<th>Output</th>
<th>Control Category by SW level</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description</strong></td>
<td><strong>Ref.</strong></td>
<td><strong>A</strong></td>
<td><strong>B</strong></td>
</tr>
<tr>
<td>1 Test procedures are correct.</td>
<td>6.3.6b</td>
<td>●</td>
<td>O</td>
</tr>
<tr>
<td>2 Test results are correct and discrepancies explained.</td>
<td>6.3.6c</td>
<td>●</td>
<td>O</td>
</tr>
<tr>
<td>3 Test coverage of high-level requirements is achieved.</td>
<td>6.4.4.1</td>
<td>●</td>
<td>O</td>
</tr>
<tr>
<td>4 Test coverage of low-level requirements is achieved.</td>
<td>6.4.4.1</td>
<td>●</td>
<td>O</td>
</tr>
<tr>
<td>5 Test coverage of software structure (modified condition/decision) is achieved.</td>
<td>6.4.4.2</td>
<td>●</td>
<td></td>
</tr>
<tr>
<td>6 Test coverage of software structure (decision coverage) is achieved.</td>
<td>6.4.4.2a</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>6 Test coverage of software structure (decision coverage) is achieved.</td>
<td>6.4.4.2b</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7 Test coverage of software structure (statement coverage) is achieved.</td>
<td>6.4.4.2a</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>7 Test coverage of software structure (statement coverage) is achieved.</td>
<td>6.4.4.2b</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8 Test coverage of software structure (data coupling and control coupling) is achieved.</td>
<td>6.4.4.2c</td>
<td>●</td>
<td>●</td>
</tr>
</tbody>
</table>

Qualification of Software in Airborne Systems

OECD Munich 5-7 March 1996

© AEROSPATIALE 1996
FRAMEWORK FOR ENGINEERING REAL-TIME SOFTWARE
FOR NUCLEAR POWER PLANTS

Paul Joannou
Ontario Hydro
Paul.K.Joannou@hydro.on.ca

ABSTRACT

The use of software as a component of nuclear power plant protective, control and monitoring systems is steadily increasing. This includes usage within systems that have impacts on nuclear safety and production reliability. Being a relatively new technology, software engineering has not matured to the point of having industry standards that reflect a widely adopted set of practices that produce software that can be relied upon for a range of applications in nuclear power plants.

This paper describes a framework that has been developed by Ontario Hydro and AECL that is being used to define requirements for the engineering of protective, control and monitoring applications in CANDU nuclear power plants. The framework distinguishes the requirements based on the software's impact on nuclear safety using a risk based approach. The requirements for engineering new software, qualifying predeveloped software and modifying existing software are dealt with separately.

The framework has been actively applied for several years. Experience with its application, along with the lessons learned are described.

OVERVIEW

Atomic Energy of Canada and Ontario Hydro have utilised digital systems for nuclear power plant control from the first CANDU plant at Douglas Point in 1957. We have used this experience to establish a philosophy for digital systems that has extended to the use of computers for safety systems. The 600 MW class of CANDU plants have used computers in the shutdown systems from the early 1980s. Recently in the Ontario Hydro Darlington Nuclear Generating station, 2 fully computerised, diverse shutdown systems were implemented. The functionality includes trip decision logic, displays to the operators, computer-aided testing, and monitoring.

AECL and OH have been evolving the safety critical software development methodologies right from the first digital protection systems installed in the early 1980s. Here the emphasis was on keeping the design simple and using relatively "standard" software development techniques. The results have been outstanding. These systems have recorded over 300 unit-years of operation without a single unsafe failure recorded. However, in spite of these results, in the course of design evolution, a number of software issues have arisen. The main issues are:

- There is no agreed upon, measurable definition of acceptability for the engineering of safety critical software within the technical community.
- There is no widely accepted and adopted practices for the specification, design verification and testing of safety critical software.
- It is not possible to quantify the achieved reliability of the software component of a safety system.
- It is not possible to quantify the benefits of using diverse software.
ONTARIO HYDRO/AECL SYSTEM DESIGN APPROACH

Ontario Hydro and AECL have evolved a approach that addresses these issues. This approach begins at the system level first since if system decisions are not made correctly, the achievement of the appropriate system reliability cannot be accomplished via software design. The system design must have the following characteristics:

1) SIMPLICITY:
   The functionality of the safety critical portion of the system must be limited to only what is essential to carry out the safety action. Non-safety functions should be displaced to other auxiliary systems to the extent practical.

2) CONTEXT:
   Safety design must be done on an overall system basis, not just on the computers or the software. This means that protection (such as redundancy, diversity, fail safe states, etc.) must be built in at a system level and mitigate against the failure of a particular hardware or software component. The concept of "defence-in-depth" must permeate through the entire design, at every level, and also extends to automated operational testing throughout the lifetime of the system.

In order to ensure a comprehensive solution we have established a framework of software standards that covers ALL levels of safety criticality. The framework consists of the following components:

i) a guideline defining the procedure for categorising software with respect to the effect of its failure on nuclear safety (with safety critical as the most stringent category).

ii) a high level standard addressing the overall software engineering process for each category.

This concept of applying different levels of rigour for design and QA according to safety impact is consistent with corresponding non-digital design methods. It allows us to utilise high levels of rigour for safety critical systems which in our system design philosophy have relatively simple functionality. Our designs are such that systems with higher degrees of functionality (and hence complexity) are those with lower safety criticality.

iii) for each category, sets of standards, procedures, and guidelines to be used to perform specific activities within the software engineering process.

iv) a guideline defining how to qualify pre-developed software for use.
CATEGORISATION

Software is categorised to select software engineering practices that achieve sufficient assurance of the reliability and safety of the software. The categorisation process is based on the effect of software failure on nuclear safety, and can be used to direct the system design process towards design decisions that minimise unnecessary reliance on software.

Software categorisation is usually initiated by system and software designers, who must decide on appropriate software engineering standards for their software, or who are in the process of making system design decisions, such as allocation of functions to hardware and software, or the architecture of a computer system.

The approach is consistent with a risk-based approach to nuclear safety, where the risk associated with the failure of a system is a function of both the probability of failure and the consequences of failure. The higher the risk associated with system failure, the higher must be the assurance that the software will not contribute to that failure. The underlying premise of software categorisation is that, as the safety significance of the software decreases, less effort needs to be expended to demonstrate that the software meets its quality objectives.

Categorisation is performed in two phases. The first phase identifies the Safety Significance of the plant system in which the software exists. The significance can be High, Medium, Low or None. The significance is based on the degree to which the system is counted upon to perform a safety related function as reflected in the systems safety related reliability requirement.

The reliability of mitigating systems is determined based on their assigned unavailability requirements. Table 1 shows the unavailability ranges associated with each safety significance. The reliability of process systems is based on their assigned initiating event frequency limit. Table 2 shows the limits associated with each safety significance.

Table 1 - Safety Significance for Mitigating Systems

<table>
<thead>
<tr>
<th>SAFETY SIGNIFICANCE</th>
<th>SYSTEM UNAVAILABILITY (Q)</th>
</tr>
</thead>
<tbody>
<tr>
<td>High Significance</td>
<td>$Q \leq 10^{-3}$ y/y</td>
</tr>
<tr>
<td>Medium Significance</td>
<td>$10^{-3} &lt; Q &lt; 10^{-1}$ y/y</td>
</tr>
<tr>
<td>Low Significance</td>
<td>$Q \geq 10^{-1}$ y/y</td>
</tr>
</tbody>
</table>

Table 2 - Safety Significance for Process Systems

<table>
<thead>
<tr>
<th>SAFETY SIGNIFICANCE</th>
<th>INITIATING EVENT FREQUENCY LIMIT (f in occurrences/y)</th>
</tr>
</thead>
<tbody>
<tr>
<td>High Significance</td>
<td>$f \leq 10^{-3}$</td>
</tr>
<tr>
<td>Medium Significance</td>
<td>$10^{-3} &lt; f &lt; 10^{-2}$</td>
</tr>
<tr>
<td>Low Significance</td>
<td>$Q \geq 10^{-2}$</td>
</tr>
</tbody>
</table>
The second phase takes into account the software failure modes and their effects on the system safety related functions. This is based on the possibility that the software may not actually implement or affect functions that dictate the systems safety significance. The software’s Failure Impact Type is assessed to be Type 1 - directly impacts safety function, Type 2 - indirectly affects safety function or Type 3 - no affect on the safety function. The Failure Impact Type is used to decrease the category of the software relative to the significance of the system.

Table 3 shows how safety significance and software failure impact are used to determine the software category.

**Table 3 - Software Category Determination**

<table>
<thead>
<tr>
<th>Plant System Safety Significance</th>
<th>SOFTWARE FAILURE IMPACT TYPE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>IMPACT TYPE 1</td>
</tr>
<tr>
<td>High</td>
<td>1</td>
</tr>
<tr>
<td>Medium</td>
<td>2</td>
</tr>
<tr>
<td>Low</td>
<td>3</td>
</tr>
</tbody>
</table>

**SOFTWARE ENGINEERING STANDARDS**

The high level standards for software engineering define the requirements on the software engineering process, define the outputs from the process, and define the requirements that must be met by each output. The requirements are specified to be as measurable as possible, but do not unnecessarily constrain the methodology used to produce the output. For example, in the standard for safety critical software the requirements on the software requirements specification output specify that the specification must define the required behaviour of the software using mathematical functions but does not specify which notation or format should be used.

A set of specific standards, procedures, and guidelines have been developed for each of the categories. These specify the detailed methodology to be used in producing the outputs specified by the corresponding high level standard for the category.

Any of the specific standards, procedures, and guidelines may consist of, or reference, industrial and international standards provided that they conform with the appropriate high level standard.

Stepwise refinement is an important concept not only because it allows the developer to tackle several more manageable problems instead of one large one, but also because it allows the verifier to review more effectively. It is very difficult to review software listings to determine if they represent a solution to the right problem. It is much more manageable to first verify that the requirements are correct, then that the design description satisfies the software requirements, and then, finally, that the code satisfies the design description.
Figure 1 shows a generic lifecycle that typifies this concept of stepwise refinement.

![Software Engineering Lifecycle Diagram](image)

**Figure 1 - Software Engineering Lifecycle**

For safety critical software we have produced a high level standard that is methodology independent but which imposes requirements on the software engineering processes as well as the outputs of these processes. Among other attributes of this standard, the following are the key ones that establish the fundamental basis of our approach:

**Formal specification of requirements:** Documentation must be prepared to describe clearly the required behaviour of the software using mathematical functions written in a notation which has a well-defined syntax and semantics.

**Review and verification:** The outputs from each development process must be reviewed to identify that they comply with the requirements specified in the inputs to that process. In particular, those outputs written using mathematical functions must be systematically verified against the inputs using mathematical verification techniques.

**Reliability testing:** Reliability of the safety critical software must be demonstrated using statistically valid, trajectory-based random testing.

It is accepted that one cannot exhaustively test all possible combinations of inputs and outputs (except for extremely simple programs). As a result, the software is to be placed in service in the knowledge that it may encounter a combination of input conditions never tested for and for which it may fail to meet its requirements.

As far as possible, this degree of uncertainty must be quantified so that it can be shown to be consistent with the reliability requirements of the overall system.
It is possible to use random testing as a means of determining the probability that a software product will encounter an input sequence that will lead to errors. The number of test cases needed to demonstrate a specified reliability within a defined confidence level can be calculated, and has been found to be a practical number of test cases for the typical reliability required of nuclear safety system applications.

QUALIFICATION OF PREDEVELOPED SOFTWARE

Reducing costs is a key priority for all nuclear power plants. To help achieve this, the use of predeveloped software products is being considered more frequently, over traditional custom software engineering. The aim is to obtain the best value without sacrificing quality. Whenever software is used, it is incumbent on the designer to ensure that its quality is acceptable before placing it in service.

Since the standards in our framework are not international standards that are widely adopted by many equipment vendors, it not realistic to expect that any particular software product was developed in a manner that met all of our requirements. Hence it is necessary to evaluate the product using a number of techniques that will indicate whether confidence can be obtained that the software meets its quality objectives.

Confidence in a product can be achieved through a) assessment of the process that was used to engineer the product originally and the process used to maintain it subsequently, b) assessment of the product itself to determine if it has the attributes expected of reliable software, and c) assessment of the degree to which the software has demonstrated itself to be reliable via its operating history and failure information.

Using these techniques, the necessary level of confidence may be achieved, or it may only be achieved if the product is used within certain constraints that restrict its usage to only those where there is confidence, or the results of the assessment may be that the necessary degree of confidence cannot be achieved, and it may be necessary to perform additional verification and resolve any problems discovered before the necessary level of confidence is achieved.

MODIFICATION OF EXISTING SOFTWARE

Many software based systems exist that were not originally developed in accordance with the current framework. When changes are made to these systems it is not practical to enforce conformance to the standards for the changes. This leaves the maintainer with the issue of what requirements should be met in the process of making a change to existing systems. A guideline has been produced to assist with the decision making of what degree of conformance to the standards makes sense for any particular system and change. The degree of conformance to the standards is affected by the existing state of the software with respect to structure and documentation, and is affected by the size and type of change.

Modifying existing software also raises issues that are not explicitly addressed in the standards such as the degree of re-verification and testing required after a change, the assessment of impact of a change and the degree to which the system will be re-specified as a result of a change. The guideline provides a checklist of considerations to be taken into account when making the decisions.
EXPERIENCE TO DATE

Over the last five years the framework has been applied to numerous projects of all categories. Three safety critical software projects have been completed and the experience from these projects have been input into revisions of the standard and the supporting procedures.

For the safety critical projects the standard and the supporting methods have proven themselves to be effective at reducing the introduction of errors into the design process and effective at detecting errors early in the design process. Figure 2 shows the errors detected by each process and the highest level document affected by the error. The figure shows clearly that the number of errors detected later in the project by testing were very few.

Experience has also demonstrated that the cost of using mathematically precise specifications and the cost of using mathematical verification techniques is not excessive relative to the effort expended on other verification and testing activities. The chart in Figure 3 shows that the efforts for each process. As is expected, the verification and testing activities consume the large majority of the effort but the mathematical verification effort (SDV - systematic design verification and SCV - systematic code verification) only took about one third of the total verification and testing effort.

The framework has been embedded within the QA program at our Bruce A nuclear generating station for several years and hence has been applied to a wide variety of projects ranging in criticality and in technology. This has resulted in 5 software product qualifications having been completed covering both category 2 and 3 applications. The framework has been useful in raising awareness of the concerns with software and ensuring that the appropriate requirements are met for any software component used in the plant. The framework is being used by our other stations on an as-needed basis.
The framework has provided a mechanism for dealing with procurement of software and systems with software components. Currently there are no widely adopted software engineering standards that, if conformed to, provide the necessary degree of confidence in the resulting software. Because of this, it is difficult to procure software and systems with the necessary level of confidence. The framework has provided a baseline set of acceptance criteria against which the practices of a vendor can be assessed, and then the qualification guideline provides the tools to take other factors into account to assess if the necessary level of confidence can be achieved. Without the framework, these decisions would have to be made on a case by case basis using the particular sets of experience and knowledge of each project engineer. The framework has provided a consistent approach based on a common set of principles. As experience shows that a particular principle was wrong or ineffective, the framework provides the vehicle for capturing that fact and feeding it forward into future work.

Figure 3 - Cost Distribution

**SUMMARY**

The framework has been in use for over five years and has proven to be effective at providing a baseline for assessment of software engineering practices. The framework provides a mechanism for ensuring that a common set of principles are applied consistently across all our plants, and provides a mechanism for capturing best practice and feeding it forward to future projects. The framework is far from perfect yet, but through the processes of continuous improvement it will continually improve over time.
Specification and Verification of Safety Critical Software

Norman M. Ichiyen
Atomic Energy of Canada Limited

ABSTRACT

Atomic Energy of Canada Limited (AECL) first installed digital shutdown systems in CANDU 6 stations in the early 1980s. We have continued to evolve the software development methodologies with Ontario Hydro (OH) and over the last 4 years have developed a "framework" approach for all categories of criticality of software. Four categories are defined with safety critical software being category 1 and software having no effect on nuclear safety being category 4.

We have also developed two methodologies that comply with our safety critical standard (category 1). One is called the Rational Design Process (RDP). It can be characterized as a methodology based on state machines where the required behaviour of the software is defined using mathematical functions written in a notation which has a well defined syntax and semantics. The second methodology is called the Integrated Approach (IA). It differs from the RDP in that it uses a graphical functional notation to specify the functional software requirements.

The RDP methodology is being used by OH in the development of a digital trip meter for the Pickering Nuclear Generating Station while AECL is implementing these methodologies for the Wolsong Nuclear Generating Stations 2/3/4 - RDP in Shutdown System Number 2, and IA in Shutdown System Number 1. Both of these implementations are now complete and the results from all phases of testing show a remarkably low number of errors. These results demonstrate that the new methodologies we have developed do indeed lead to a higher demonstrable level of software reliability.

INTRODUCTION

AECL and OH have been evolving the safety critical software development methodologies right from the first digital protection systems installed in the early 1980s. This historical evolution and the progress made with respect to an overall framework for software engineering are described in the paper by Paul Joannou [1]. This paper describes the two specific software design methodologies that AECL and Ontario Hydro have developed that meet the requirements of the Standard for Software Engineering of Safety Critical Software [2].

SOFTWARE LIFECYCLE

Stepwise refinement is an important concept not only because it allows the developer to tackle several more manageable problems instead of one large one, but also because it allows the verifier to review more effectively. It is very difficult to review software listings to determine if they represent a solution to the right problem. It is much more manageable to first verify that the requirements are correct, then that the design description satisfies the software requirements, and then, finally, that the code satisfies the design description.
Figure 1 shows a generic lifecycle that typifies this concept of stepwise refinement.

SYS = English language computer system functional specification
SRS = Software Requirements Specification (mathematically precise notation)
SDD = Software Design Description (mathematically precise notation)

Figure 1: AECL and Ontario Hydro generic safety critical software lifecycle

For safety critical software we have produced a high level standard [2] that is methodology independent but which imposes requirements on the software engineering processes as well as the outputs of these processes. Among other attributes of this standard, the following are the key ones that establish the fundamental basis of our approach:

*Formal specification of requirements:* Documentation must be prepared to describe clearly the required behaviour of the software using mathematical functions written in a notation which has a well-defined syntax and semantics.

*Review and verification:* The outputs from each development process must be reviewed to identify that they comply with the requirements specified in the inputs to that process. In
particular, those outputs written using mathematical functions must be systematically verified against the inputs using mathematical verification techniques.

**Reliability testing:** Reliability of the safety critical software must be demonstrated using statistically valid, trajectory-based random testing.

It is accepted that one cannot exhaustively test all possible combinations of inputs and outputs (except for extremely simple programs). As a result, the software is to be placed in service in the knowledge that it may encounter a combination of input conditions never tested for and for which it may fail to meet its requirements.

As far as possible, this degree of uncertainty must be quantified so that it can be shown to be consistent with the reliability requirements of the overall system.

It is possible to use random testing as a means of determining the probability that a software product will encounter an input sequence that will lead to errors. The number of test cases needed to demonstrate a specified reliability within a defined confidence level can be calculated, and has been found to be a practical number of test cases for the typical reliability required of nuclear safety system applications.

**SPECIFIC SOFTWARE DEVELOPMENT METHODOLOGIES DEVELOPED**

We have developed 2 specific software development methodologies that are consistent with our high level standard for safety critical software engineering. They are:

1) Rational Design Process (RDP)
2) Integrated Approach (IA)

The RDP has evolved from the techniques used for the Darlington computerized shutdown systems. It can be characterized as a methodology based on finite state machines where the required behaviour of the software is defined using mathematical functions written in a notation which has a well defined syntax and semantics. The input/output behaviour is defined in tabular format.

The IA proceeds from a system description through hardware design specification and software requirement specification to software design and automatic generation of executable code. A characteristic of the process is the use of a graphical functional notation to specify the functional software requirements. This permits ease of review of the software requirements by people of many disciplines. Further, in performing the software design, the structure of the software requirements is maintained. Only local logic additions and small transformations are performed. Finally, since the notation of the software design is formal, executable code is generated directly from it without the need for a manual coding step.

While both methodologies are quite diverse, they both meet the requirements of our high level standard for safety critical software.

**METHODOLOGY EXAMPLES - SRS**

In order to give the reader some insight into our two methodologies, this section shows an example of the mathematical notations used in the two methodologies for a typical function - behaviour of a reactor trip parameter (PHT Pressure) for high and low going trip setpoints with hysteresis. Such a trip parameter can be considered to have 5 regions of operation as shown in
Figure 2. In the hysteresis regions, the output depends on whether the input signal was previously in the tripped regions or the "normal" region.

![Diagram](image)

*Figure 2. An example trip function*

RDP Methodology:

The RDP methodology uses mathematical tables to describe the input-output relationship. Figure 3 shows a fragment of the SRS data flow diagram that describes the relationship. The notation is described in Table 1.

For simplicity, we will show the condition tables defining the part of figure 3 shown as dotted. The mode table that defines the 7 modes based on all possible states of the input signal and its history is represented as Table 2. The function table that defines the PHT sensor trip function status based on all monitored to controlled variable relationships of the above example, is represented as Table 3.

![Diagram](image)

*Figure 3. Fragment of a software requirements specification data flow diagram (Rational Design Process methodology)*
Table 1. Description of signal names used in Fig. 3

<table>
<thead>
<tr>
<th>Signal Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>F_PHT_HL, F_PHT_LL</td>
<td>These are the PHT pressure High and Low Level setpoint functions (functions are indicated by the F_ preceding the signal). We have used the general function notation since some of the setpoints are functions of reactor power, handscrew position, etc while others are constants.</td>
</tr>
<tr>
<td>M_PHT</td>
<td>This designates the &quot;monitored&quot; variable; the PHT pressure</td>
</tr>
<tr>
<td>F_PHT_ST</td>
<td>This is the PHT sensor trip function status</td>
</tr>
<tr>
<td>S_PHT_ST</td>
<td>This is a state variable describing the previous state of the function F_PHT_ST</td>
</tr>
<tr>
<td>F_PHT_Ptrip</td>
<td>This is the function describing the PHT &quot;parameter&quot; trip status</td>
</tr>
<tr>
<td>C_PHT_Ptrip</td>
<td>This designates a &quot;controlled&quot; variable; the PHT pressure output parameter trip status</td>
</tr>
</tbody>
</table>

Table 2. Sample software requirements specification function notation, mode table

<table>
<thead>
<tr>
<th>Mode</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>HiTrip</em></td>
<td>M_PHT ≥ F_PHT_HL</td>
</tr>
<tr>
<td><em>HiHysNorm</em></td>
<td>(F_PHT_HL &gt; M_PHT ≥ F_PHT_HL-K_PHTHys) ∩ (S_PHT_ST=K_High)</td>
</tr>
<tr>
<td><em>HiHysTrip</em></td>
<td>(F_PHT_HL &gt; M_PHT ≥ F_PHT_HL-K_PHTHys) ∩ (S_PHT_ST=K_High)</td>
</tr>
<tr>
<td><em>LoTrip</em></td>
<td>M_PHT ≤ F_PHT_LL</td>
</tr>
<tr>
<td><em>LoHysNorm</em></td>
<td>(F_PHT_LL &lt; M_PHT ≤ F_PHT_LL+K_PHTHys) ∩ (S_PHT_ST=K_Low)</td>
</tr>
<tr>
<td><em>LoHysTrip</em></td>
<td>(F_PHT_LL &lt; M_PHT ≤ F_PHT_LL+K_PHTHys) ∩ (S_PHT_ST=K_Low)</td>
</tr>
<tr>
<td><em>Normal</em></td>
<td>F_PHT_LL + K_PHTHys &lt; M_PHT &lt; F_PHT_HL-K_PHTHys</td>
</tr>
</tbody>
</table>

Table 3. Sample software requirements specification function notation, function table.

<table>
<thead>
<tr>
<th>Partition/Variable</th>
<th><em>HiTrip</em></th>
<th><em>Normal</em></th>
<th><em>LoTrip</em></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>∪</td>
<td></td>
<td>∪</td>
</tr>
<tr>
<td></td>
<td><em>HiHysTrip</em></td>
<td><em>HiHysNorm</em></td>
<td><em>LoHysTrip</em></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F_PHT_ST</td>
<td>K_High</td>
<td>K_Norm</td>
<td>K_Low</td>
</tr>
</tbody>
</table>

IA METHODOLOGY

The IA methodology uses graphical notation to accomplish the same task. Figure 4 shows the "Monitored" to "Controlled" variable relationship of the above example. It uses three function blocks to describe this relationship. The definitions of the COMPH (Comparator High), and COMPL (Comparator Low) blocks are given on Figures 5&6. The third block is simply an OR block; that is, the controlled variable (//PHT_Ptrip/) is tripped if either the COMPH or COMPL blocks are in the tripped state. The hysteresis functions are handled in the COMPH and COMPL blocks.
Figure 4. Software requirements specification representation of the example trip function (Integrated Approach methodology).

**DESCRIPTION:**

```
<table>
<thead>
<tr>
<th>Out₁</th>
<th>In₁</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>⬡</td>
</tr>
<tr>
<td>0</td>
<td>⬛</td>
</tr>
</tbody>
</table>
```

**ICON:**

```
X COMPH
L
HYS X>L
```

**DEFINITION:**

<table>
<thead>
<tr>
<th>In₁, &gt; In₂</th>
<th>In₂-In₃ &lt; In₁ ≤ In₂</th>
<th>In₁ ≤ (In₂-In₃)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Out₁</td>
<td>'Out₁</td>
<td>False</td>
</tr>
</tbody>
</table>

Figure 5. Integrated Approach Definition of the COMPH block
DESCRIPTION:

ICON:

DEFINITION:

APPLICATIONS:
1) Pickering Digital Trip Meter (Ontario Hydro):
The Heat Transport High Temperature (HTHT) trip parameter in Pickering Nuclear Generating Station 'B' is intended to provide trip coverage for certain slow loss of regulation accidents, for partial loss of station Class IV electrical power accidents and for some feedwater and steam supply failures.

The present Shutdown System No. 1 (SDS1) HTHT trip system consists of Resistance Temperature Detectors (RTDs) in six different inlet headers providing signals to six analog indicating alarm units, also referred to as trip meters. The trip meters are channelized, two per channel, into three channels. Additional protection for this trip parameter is also provided by a second independent and diverse reactor protective system, Shutdown System No. 2 (SDS2).

The present SDS1 analog trip meters rely on operator interaction in the safety performance of the HTHT trip system. The operating procedures require the operator to periodically monitor both the process temperature and the trip setpoint from the analog trip meter display, and to maintain the trip setpoint within a pre-determined margin of the process value, by manually adjusting the trip setpoint. The maximum permissible margin is determined by the plant safety analysis, and if exceeded, the HTHT trip will be impaired.

Ongoing problems with monitoring the process deviation from the trip setpoint of the HTHT trip system has led to an investigation and ultimately the initiation of the Digital Trip Meter development project. When no acceptable method of expanding the deviation value was possible, the use of an accurate digital trip meter was deemed to be a suitable solution for monitoring and maintaining the maximum permissible process deviation. The trip meter is also responsible for initiating the protective reactor trip action via contact outputs.
Due to the unavailability of a qualifiable commercial device, a decision was made to develop a digital trip meter for use in the HTHT trip system. In order to make the design cost effective, the development of a generic digital trip meter that addressed the needs of the HTHT trip system improvements and a number of potential future safety system upgrades, was launched. The design emphasized the selection of a hardware vendor with considerable experience in design and manufacturing of digital panel meters with some software expertise. Ontario Hydro's Nuclear Technology Services Division was responsible for the high level hardware design and for engineering of the software. A contract was awarded in November 1992 for the detailed hardware design and manufacturing of the digital trip meter. The RDP methodology is used for this application.

2) Wolsong Units 2,3&4 Shutdown Systems No. 1 and 2 PDCs (Programmable Digital Comparators) (AECL):

Shutdown systems in a CANDU nuclear power plant have a special safety function to monitor the state of the reactor and to initiate a rapid shutdown of the nuclear reaction when it detects an unsafe condition. CANDU plants use 2 independent, diverse, equally capable shutdown systems, called SDS1 and SDS2. Each shutdown system uses triplicated, channelised, logic to sense a reactor trip condition being required. The PDCs take the place of the analog comparators (used in previous designs) for the majority of the process trip parameters; e.g. low and high PHT (Primary Heat Transport) pressure, low steam generator level, low pressurizer level, low flow, etc. These trip parameters are generally functions of reactor power, and are conditioned out below certain powers (i.e. they are not required to trip the reactor if the reactor power is below certain levels). Figure 8 shows a typical trip parameter - Pressurizer Level where the trip setpoint is a function of reactor power. Included in the design is the use of software self-checks, rationality checks, and hardwired "watchdogs" to ensure that any failures are converted to the "safe" or tripped state. SDS2 uses the RDP software development methodology while SDS1 uses the IA approach.

![Figure 7. Pressurizer level trip setpoint as a function of the compensated average flux detector power P (in percent of full power).](image)
RESULTS:

FORMAL SPECIFICATIONS:
The use of the "formal" or mathematically precise notation for both SRS and SDD has been extremely successful in addressing the most critical area where "software errors" have historically been made. That is, at the software requirements specification. Typically in the past, most errors have occurred here due to ambiguities in the specification. However, now with either our RDP (tabular mathematical notation) or IA (functional graphical language) methodologies, since they are mathematically precise, there are no ambiguities. The SRS is now more complete since input domain coverage can be checked to determine if the required behaviour of the outputs has been specified for the complete, valid range of each input and for all combinations of inputs that affect each output.

Since the SRS was mathematically precise and since it was also very reviewable by the system designers, we were able to ensure correct 'translation' from the system level English language specification at this stage. In fact, for the Wolsong SDS1 design process (using the IA) there were no changes required to the SRS. The time spent at this stage turned out to be well worth the initial effort making the following stages of the design and testing process much simpler with few changes required. In fact, the results of the Unit and Subsystem Testing have been that virtually no errors have been found relating to the formal specifications.

VERIFICATION:
The two formal verification stages, the SDV (Software Design Verification); i.e. verification of the SDD to the SRS, and Code Verification; i.e verification of the Code to the SDD, were successfully carried out but did not turn up a significant number of errors. We attribute this to the nature of the process itself. That is, the combination of the stepwise refinement design process together with the clarity of the notation in the SRS and SDD led to the designers being able to produce the subsequent stages of the design with very high reliability. Another important factor contributing to the small number of errors found is that in all our applications we have used small teams of highly qualified staff. The teams are made up for the most part of the original "architects" of these methodologies which means they are extremely familiar with the concepts and hence can implement them effectively.

A by-product of the verification process was that the verifiers were able to identify portions of the design that were complex to analyze. In many instances this has led the designers to further simplify the design thereby making the design more robust.

RELIABILITY DEMONSTRATION:
Reliability demonstration is the application of probabilistic reasoning, via statistical models based on random sampling, to gain quantifiable confidence that software meets its reliability requirements. The reliability demonstration technique used is based upon a Bernoulli experiment. Investigations have been conducted to fully understand the theoretical basis of the approach [3].
This technique of gaining confidence in software has been applied on three, safety critical applications within Ontario Hydro nuclear stations, in addition to the Pickering Trip Meter and the Wolsong shutdown systems projects. The experience gained to date has confirmed the need for good tools to make the technique cost effective and to minimize the number of anomalies needing manual analysis. The technique consists of the following basic steps:

1. Generate input trajectories that match the expected profile of the inputs during design basis events that the safety system is protecting against. Each input trajectory generated uses a randomly selected initial value, final value and trajectory shape.

2. Input the trajectories to the system under test and record the actual output trajectories.

3. Produce a test oracle, and use it to generate the expected output trajectories.

4. Compare the actual results to the expected results and flag any actual results that do not match the expected results, within allowable tolerance, as anomalies.

Experience has shown that tooling of all four steps is necessary to make the large number of trajectories affordable. It has been found that producing a high fidelity test oracle has been an important step in reducing the number of anomalies that require manual analysis from step four.

The software reliability requirements for our systems has typically been $10^{-4}$ probability of failure upon demand. We have been using 7000 trajectories to demonstrate the $10^{-4}$ with a confidence interval of 50%. We have found 7000 tests to be an achievable number of tests for the types of systems that have undergone reliability demonstration to date.

PRACTICALITY:
The methods have proven to be quite practical. Figure 8 shows a pie chart representation of the relative distribution of costs for the Wolsong 2,3,4 SDS1 PDCs.

![Figure 8. Wolsong units 2,3,4 SDS1 PDC Cost Breakdown](image)
There are several points that are significant about the cost breakdown. Systems engineering, hardware engineering and software engineering were about equal in cost. This shows that even with the high degree of rigour involved in the design as well as the formal verification and testing, software development is not a high portion of the cost. A set of testing tools were developed for the IA which allowed the testing to be carried out in a very efficient and cost effective manner. This re-inforces the need for tools for cost effectiveness in any software engineering process. The software costs were low because there was virtually no re-work required. The mathematically precise graphical language allowed review by both system designers and software designers which resulted in a smooth and error-free development.

CONCLUSIONS:
The AECL and Ontario Hydro approach of having a top level methodology independent standard for each category of software has permitted us to establish a consistent framework for all software. Specifically for safety critical software, our principles for systems design; i.e. emphasis on simplicity and defense-in-depth have led to a simplification of the solutions to the issues surrounding software engineering for safety critical software. The results to date of the applications of the new safety critical software engineering methodologies developed by AECL and Ontario Hydro have demonstrated both the practicality and benefits of the two software development methodologies, the RDP and the IA.

REFERENCES


SESSION 1 - AFTERNOON

RECONCILING THE STRUCTURED SOFTWARE DESIGN PROCESS TO LARGE SCALE NUCLEAR I&C PROJECT SCHEDULES - Mr. Bruce M. Cook, Westinghouse Electric Corporation

PLANNING & SPECIFYING OF DIGITAL BASED REACTOR PROTECTION SYSTEM FOR NEXT STAGE PWR PLANTS IN JAPAN - Mr. Yoshihiro Yamamoto, Kansai Electric Power Company

REGULATORY APPROACH IN FUTURE LICENSING OF COMPUTER-BASED SYSTEMS - Mr. Károly Hamar, Hungarian Atomic Energy Commission

FINNISH REGULATORY REQUIREMENTS FOR PROGRAMMABLE COMPUTER-BASED AUTOMATION (I&C) SYSTEMS - Mr. Harri Heimbürger, Finnish Centre for Radiation & Nuclear Safety

licensing requirements for safety critical software of the npp temelin i&c systems - Mr. Ceslav Karpeta, State Office of Nuclear Safety, Czech Republic

Current status and licensing experience of computer-based safety systems in Korea - Mr. Won-Young Yun, Korea Institute of Nuclear Safety

Regulatory aspect of digital safety protection system in Japan - Mr. Zen-ichi Ogiso, Nuclear Power Engineering Corporation

Retrofitting to programmable electronics in nuclear power plants - Requirements aspects - Mr. Erik Johansson, KTH, Royal Institute of Technology, Stockholm
ABSTRACT

Westinghouse is in the midst of their second large scale, short schedule nuclear instrumentation and control installation project. The principal key to success in these programs is the application of distributed digital I&C systems based on standard product lines. By decoupling the specification of the hardware from the details of the functional design, the manufacturing process, with its long lead times, can be put on a schedule that meets both regulatory safety standards and the project needs. US and international nuclear standards have recognized that when digital systems are applied to functions that are important to nuclear safety, special care must be taken to ensure the integrity of the implementation. They mandate structured design processes with the product of each design step being verified against the requirements that are inputs to that step. While there is little doubt that such processes can produce high quality software products, the length of time required to implement them can potentially impact the schedule in projects such as those described above. Therefore, managing the software development process in large nuclear projects becomes a balancing act, pushing to meet schedular commitments, adapting plans to buy precious time, demonstrating regulation compliance that ensures high quality. Risk management is an essential component of the daily planning in these cases. This paper describes some methods used, and lessons learned, to achieve success.

I. HISTORY OF DIGITAL I&C IN NUCLEAR POWER

In the mid 1970's, two development programs converged at Westinghouse that shaped the way that nuclear I&C would be dealt with in the company for the future. The first of these was a program to design the next generation, large capacity standard plant model, known as the 414. The other program was one that was viewing the emerging technology of microprocessors to see what impact they might have on the future of the industry. The combination of the programs lead to the development of an integrated I&C architecture that offered solutions to the known problems of the day. These solutions focused on the application of fiber optic communications to improve isolation of various parts of the systems from one another, integration of surveillance test features to reduce the operational effort required for periodic testing, and enhanced fault tolerance to reduce the number of forced outages caused by spurious actions that result from I&C failures. The progression of milestones of the Westinghouse digital I&C development activity is shown in Figure 1.

The functions performed by the early versions of the digital I&C systems were simply simulations of those in the analog systems they replaced. Later, as the capabilities of the technology became apparent, more sophisticated algorithms were used in both the protection and control functions that were implemented in these systems. Nevertheless, there is still a strong basis in the conventional functional design approach followed for these systems.

Following the design and implementation of a full protection system prototype, including the establishing of the software development process, the Westinghouse technology was selected for the Primary Protection System at the Sizewell B Nuclear Power Station in Suffolk, England. Following the awarding of the contract for this system in 1984, work began on the design of the second generation prototype system. In addition to bringing the technology in line with the rapidly evolving industry, through the application of 16 bit microprocessors and high performance
communications highways, the focus of this generation was on enhancing the "fail safe" aspects of the design to meet established British practices for nuclear safety systems. Also, the software development process was further refined and design verification methods were established. The experience gained from the design of this generation of the technology formed the basis for the successful application of digital I&C to nuclear safety systems in the United States with the installation of five upgrade protection systems. These upgrade products were known as Eagle 21, a name which has now been adopted to refer to all of the Westinghouse nuclear specific I&C products. In 1991, while the Sizewell PPS was still on the manufacturing floor, the Nuclear Electric Company, Sizewell's owner, made the hard decision to abandon the vendor for much of the remaining I&C systems for the plant, and awarded that scope to Westinghouse on the condition that the original plant construction schedule be maintained. That left just two years to design, manufacture, install and test the largest digital I&C system that Westinghouse had been involved in to date. These remaining I&C systems are collectively known as the Integrated Systems for Centralized Operations (ISCO), and include safety related and non-safety controls, and the distributed computer system which collects information for display to the operations staff. The project was successful, the Sizewell B plant is now in commercial operation.

Now, Westinghouse is in the midst of our second, large scale nuclear digital I&C system design. This one is for the Temelin Nuclear Power Plant in the Czech Republic. The two units on this site are Russian designed VVER's that have been under construction for a number of years. The buildings and fluid systems were well progressed, but the owner decided that modern, western I&C technology was necessary to operate the plant into the next century. Although there are many similarities between the I&C systems for Temelin and Sizewell, there are equally many new aspects of the design brought on by the merging of Russian design concepts with those of the US and western Europe. Also, the scope of the I&C systems for each Temelin unit is larger than the combined PPS and ISCO efforts on Sizewell. For Temelin, the Turbine Control System is included, as is a comprehensive Monitoring and Diagnostic System. One aspect, however, that is very similar between the
two projects is the short schedule.

II. BACKGROUND OF INDUSTRY STANDARDS

Of course, Westinghouse was not the only organization who was looking at the application of digital I&C technology to nuclear power plants. Not long after Westinghouse started their development program, the US Nuclear Regulatory Commission began to formulate regulatory guidance on the application of digital computers to safety systems. For a basis, the NRC looked to the aerospace industry because they felt that many of the related issues had been dealt with there. From the regulatory view point, the principal issue that needed closer scrutiny was the computer software. This was due to the inherent design flexibility provided by software and the chance of it introducing design errors. The US nuclear industry also became active in establishing design rules. A joint working group of ANS and IEEE was formed to address this topic. They also found that, while existing standards such as IEEE 603 provided all necessary requirements for the system design, the issues of software quality and qualification (verification) needed to be expanded. The product of this joint working group is the standard ANS/IEEE 7-4.3.2. 1 This standard has been endorsed by the USNRC as the sufficient set of requirements applying to the application of digital computers to nuclear safety systems. This standard has since been revised, and endorsement of the new version by the USNRC is anticipated in 1996. Following the initial publication of the nuclear specific standard, the IEEE Computer Society published a series of standards that apply to safety critical software regardless of industry. These standards present a view of the Quality Assurance organization that is inconsistent with the established practice in the nuclear industry, and are therefore difficult to apply without liberal interpretation. This inconsistency was further aggravated by the issuance of ASME NQA-2a-Part 2.7 2 in 1990, which deals specifically with the quality assurance aspects of software that is to be used in nuclear power plants.

International organizations were also taking up the topic of establishing requirements for safety critical software. In particular, the International Electrotechnical Commission (IEC) published IEC 880 3 after many years of deliberation. This standard is more prescriptive than its US counterpart, and provides requirements on the design of the computer system and software in addition to those placed on the process followed to design these elements. Three years after the publishing of the safety system software standard, IEC published standard 987 4 which deals with requirements for the hardware portion of computer based safety systems. In the midst of all of this standard writing activity, the International Organization for Standardization (ISO) published ISO 9000-3. 5 Although not specific to the nuclear industry, nor limited to safety critical software, this software quality standard is being adopted by many companies as the basis for their software development process.

Computer software has drawn codes and standards like moths to a flame. While the intent of these standards is honorable, to ensure the quality of the product, the burden they place on large projects is costly, and needs to be considered.

III. THE NATURE OF LARGE PROJECTS

As the design codes and standards were being written, the authors undoubtedly had a mental image of the work involved in the application of computers to the safety systems. These mental images were probably bounded to the functions that would be performed in a single computer. If asked whether the principles apply to large systems, they would no doubt answer in the affirmative, saying that it is just a matter of scale, and the amount of resource you apply to get the job done. However, in large distributed digital systems such as Sizewell and Temelin, there is a network of interfaces, both in the systems and in the organizations, that make what seem to be straightforward concepts expressed in the standards difficult to deal with.

One of the more prevalent problems on large scale projects is the incompleteness of the requirements specification at an early enough stage of the project. As the responsibilities for design activities are divided among various groups in different organizations, the coordination of the organizational interfaces becomes a substantial effort. Managing conflicting priorities is the key role of the project integration team. While a design group may recognize their responsibility to provide input requirements to another group, it may not be high on their priority, and in fact may require a significant amount of their own design effort requiring input from a third group, and so on. As the number of organizational interfaces increases, the potential for conflict increases exponentially. It is possible to create gridlock situations that bring progress to a halt. For example, the control system designer needs a list of all electrical distribution breakers to be controlled so that he can assign the loads to cabinets and thereby determine the total number of control cabinets needed. On the other hand, the electrical system designer needs to know the number of cabinets so that he can
determine the requirements for the distribution breakers. Issues such as this are routinely solved by working together and results in the production of several design iterations.

Both the Sizewell and Temelin projects had a short schedule because of a change in the I&C vendor rather late in the construction schedule. One would think that this should at least provide a better situation in regard to the finalization of the fluid systems design and hence the requirements place on the I&C systems by those systems designs. In fact, this was not the case in either project. Despite the relatively late start of the I&C design process, the continuous evolution of the functional design requirements was considerable. This may lead one to consider whether any large integrated I&C system can be considered to have anything other than a short schedule. If this is the case, it needs to be factored into the planning for the commissioning of the systems.

The increase in scale of large projects has another effect that can be easily overlooked. That effect is the need to account for resource limitations in the planning of any activities, but especially the integration and testing activities on site. It may only take two weeks to install and test a particular cabinet of equipment. However, if the number of cabinets to be installed and tested is large, Temelin has over 450 cabinets per unit, the lead time to start the testing can easily exceed a year in time. If delays in the up front activities delay the shipment of the earliest cabinets, a "bow wave" of work to be done gets stacked up to the point that recovery by adding resources becomes impractical.

IV. ROLE OF STANDARD PRODUCT LINES

The key to success when implementing large scale projects is the application of standard product lines that are configured to perform the functions for the specific plant. This should not come as a surprise because it was the way of doing business for nuclear I&C during the decades of the use of analog technology. All of the reactor protection systems for Westinghouse plants are virtually the same. Before a power plant even had a name, the majority of the modules of its protection system were assigned positions in racks. However, the modern design standards lead us to a different approach. First the plant is analyzed, then the requirements for protection and control are determined. Only when this step is complete are computer elements to be selected and the software prepared. Working in such a serial fashion is a time consuming process, and would generally not provide interface information for plant layout and cabling concerns at an early enough stage to support the plant schedule.

In practice, the standard approach is also followed for modern digital systems. At least from a general sense, the architecture is established long before the requirements for the systems are collected, often before any contract to do work is even signed. This is necessary so that the I&C vendor can count the cost of the commitments he is making. By using standard product lines, the specification of the hardware to be used can be separated from the functional design. The number of cabinets to be used in any segment of the architecture can be determined from an approximation of the number of loads and sensors to be processed by that segment. The details of the functional design do not need to be established until later in the process. This approach also provides early information on layout space required and other support requirements such as power and HVAC. These estimates are only good if they are based on experience gained with standard products. If they were mere guesses based on conceptual designs, there would a large margin for error.

Of course, the standard product used must meet the strict requirements of the nuclear industry. Many commercial grade digital control systems would not meet the strict design verification requirements imposed by the industry codes and standards. Therefore, it is essential that the standard product is selected based on its experience in similar applications, i.e. nuclear power plants and, where applicable, important to safety applications within those plants. This is generally stated in bid specifications as a requirement to use proven technology.

Standard product does not only apply to the computer hardware that is used. Too often, the computer based I&C systems are viewed as empty boxes that can perform any function that is programmed into them. A significant benefit can be realized in the reduction of software design effort through the use of standard software modules as well. For Temelin, a significant portion of the system software used in the safety system and safety related control system was recovered from the software developed and verified for the Sizewell project, as is shown in Figure 2. The software that needed modification was due to evolution of the standard technology, and not due to application differences in the plant. This reuse of software provides additional savings in the design verification effort. It also enhances the reliability of the system due to the broader application experience gained for the software.
V. ADAPTATION OF INDUSTRY STANDARDS

While there is a clear benefit provided by the publication of design codes and standards to provide a uniform basis for the regulatory review of computer based safety systems, one must realize that these codes have been prepared from one particular point of view, that being the "clean sheet" design of a nuclear power plant. On projects where the I&C is being applied later in the project cycle, such as upgrade systems or the late phase construction efforts like Sizewell and Temelin, consideration of the real world conditions needs to be taken in the application of those standards. During the writing of the standards, thoughts were expressed along the lines that the quality process should not be constrained by schedule or budget. While this ideal may be the basis for the guidelines contained in the standard, it can create difficulties in application to the real world. All contracts have delivery schedules and costs that are constrained in one way or another. Inevitably, slippage on the up front phases, where functional requirements are formed, lead to mandated recovery plans in the later stages, and compression of the original plans.

One of the principal tenets of the US and international safety critical software standards is the application of a "top down" design methodology. Following this approach, the requirements for a given design process step are determined before that step is taken. In software parlance, this is known as the traditional waterfall approach to software development. In actual practice, it presents some problems. First, if the safety system architecture is based on the application of a standard I&C product line, much of the design will be established prior to a full consideration of the requirements. Perhaps stated more precisely, the requirements emerge in a piecemeal fashion. Early in the project life cycle, often before contracts are signed, a sufficient set of requirements exist to establish the general architecture of the system. Details, such as what signals will be processed may only be known in gross generalizations, such as estimated input and output counts. Nevertheless, based on these sketchy requirements, the basic selection of the hardware to be used is made. This selection is usually closely coupled to the selection of the vendor who will provide the system. Contrast this fact with the design standard approach that mandates a complete documentation of the requirements for the computer based safety system prior to selection of the hardware and software elements that will be used to implement the design. Therefore, the design standards should be adapted to allow for incomplete and changing requirements.

There is also a need for latitude in the time phasing of the design process steps that allows for progress to be made in parallel activities. IEC 880 specifically states that each step is to be ended by a verification activity that must be completed before starting the next step. This is impractical if the requirements are not complete. Frequently, it becomes necessary, in order to maintain progress on the project schedule, to proceed based on assumptions or best estimates of what the requirements will be when they become final. This of course introduces an element of risk that should be recognized, and
shared, by both the system designer and the utility.

Management of risk becomes the dominant factor in large projects that are constrained by schedule. Because of this, engineering judgement plays a significant role in the design process. From the view point of the standards, however, there seems to be little room for judgement. A requirement is written, design alternatives developed, and the ultimate solution selected based on rigorously documented thought process. All of this subject to review by an independent verifier. If the requirements are not complete at the time a particular design decision must be made, the collective experience of the designers is generally used to make assumptions that permit progress to be made. Often, these judgements are difficult to document. The ultimate system validation must ensure that in the end, an acceptable design has been delivered to the plant.

If the documentation of the requirements is difficult in the early stages of a large project, the traceability of design features to those requirements is even more so. Requirements stated at the top level for I&C systems tend to general statements of design bases. Extracting specific requirements that can be traced through the successive levels of design documentation is a difficult process for small systems, and is overwhelming for large systems. Further, the basis for some of the design features will come from the definition of the standard product line, and cannot be readily traced to the specific project requirements documents. Industry standards, and the licensing agencies who apply them, should evaluate the true worth of this method of design justification.

The design processes mandated by the safety critical software standards should allow a mechanism for applying limitations, imposed by the use of standard system, on the functional requirements. As the standards presently reflect, the design requirements are held as a given, and the selection of hardware and development of the software proceed from that base. The authors of the standards would argue that an iterative mechanism does exist, and that requirements can be changed as a result of decisions made in down stream processes. However, this misses the point. First, in short schedule projects, there is usually not a sufficient amount of time available for numerous iteration of the functional design documents. Second, if a functional design is characterized by the selection of alternatives, then early input from the implementation process, through the suggestion of standard design modules, could shorten the process and avoid unnecessary iteration. Finally, if a limitation of the final equipment presents a real constraint, then the functional designers may need to consider alternative approaches rather than simply modify their requirements. For example, suppose a display system is limited in the number of graphic pages it can produce. To provide more might require abandoning the standard product in favor of new, non-proven equipment. It would be better for the functional designer to be made aware of this so that he can optimize the design of his displays within the constraint rather than be told at a later date that he has exceeded the limit and needs to reduce the required displays. In the view point of the present standards, the computer system does not yet exist when the requirements are established, so it is difficult to consider such limitations in the up front design.

Another area were the value of the conventional methods should be examined is the design verification, particularly for the software portions of the system. For the system software, detailed path analysis and exhaustive testing is appropriate. While the standards call for independence of this verification, it should also be recognized that expertise is also important if the depth of the review is going to be adequate. This is especially true for standard system software. While it is desirable to have thorough documentation of the requirements and specifications for these software modules, there is generally additional foundation knowledge that is necessary to really understand the workings of the lower levels of the system software. Giving excessive stress to the independence of the verification process will generally preclude the access to this foundation knowledge. A better approach would blend the independent verification with in depth structured self review by the designers who know the system best. This was the basis for the traditional approach to product quality in the nuclear industry, but has been spurned by the computer science discipline.

At the application software level, the methods applied to verification need to be examined. This will become even more evident as the industry progresses to graphical, tool based application generators. There will need to be more focus on the correctness of the functional design and its translation into the implemented software system, and less on the structure of the software itself. In this regard, a more rational approach needs to be taken in the coordination of system testing, both at the factory and after installation at site, and the design verification process. These two activities share the common goal of demonstrating that the installed systems perform the desired functions. However, they have been thus far treated as disjoint activities. On both Sizewell and Temelin, Independent Verification &
Validation teams have been hired by the plant utility, at the request of the regulators. These teams have focus on structural analysis of the safety software and dynamic simulation to validate the system against the design basis analysis. The effort expended might have been better applied in the confirmation of the adequacy of the functional design through participation in the definition and execution of the site test activities. In other words, such design verification activities should become more involved in the practice of the system implementation, and less in the theoretical areas.

VII. CONCLUSIONS

It has been shown that, through the application of standard computer products, short design schedules for larger nuclear power plant I&C applications can be achieved. The application of standard products, however, is not strictly in line with the design process that is called for by the published industry standards. Future revisions to these standards, and the interpretations given to them when invoking them on future projects, should take into account the dual path development that is inherent in standard product systems, one path being the development of the standard product to generic requirements and the second being the application design to plant specific requirements. The top down design approach should be modified to incorporate consideration of design constraints that result from the selection of a standard element of a computer system. Finally, the methods applied to design verification should be examined to ensure that the highest value is realized from this activity, and that excessive effort is not spent on reworking the standard product on each successive project, but rather focused on the new aspects of the design that the application software represents. As the experience base grows in the practical application of digital I&C technology to the nuclear industry, the standards should be amended to reflect that experience, and have less focus on theoretical issues.

VIII. REFERENCES


OECD/NEA  CSNI-CNRA
INTERNATIONAL WORKSHOP
on
 LICENSING ISSUES of COMPUTER-BASED SYSTEMS
 IMPORTANT to SAFETY

Munich, GERMANY
5 - 7  March  1996

Planning and specifying of digital based
reactor protection system
for next stage PWR plants in Japan

* Y. Yamamoto  Kansai Electric Power Co., Inc.
   T. Aoyagi  Japan Atomic Power Co.
   M. Utsumi  Mitsubishi Heavy Industries, Ltd.
   H. Funakoshi  Mitsubishi Electric Corporation
1. Introduction

As the instrumentation and control system has the role of the center nerve in the nuclear power plant, I&C system was designed to have high reliability and safety from the beginning of the nuclear history. At the time of construction of the newly designed PWR plant in Japan, we are trying to upgrade total I&C system design.

Digital technology, which is the major upgrading issue in the I&C system, have been applied step by step to Japanese PWR plants from portions of I&C system not important to safety to portions important to safety shown in Fig.-1.

In the next stage PWR plant, totally digitalized I&C system including safety system and soft-operation control room are planned to apply to improve safety, reliability, maintainability and testability, and to reduce plant cost.

This newly designed I&C system, especially the digital reactor protection system is carefully designed, and a large scale qualification test was carried out by NUPEC with cooperation of Japanese PWR/BWR utilities and venders.

2. Design of the reactor protection system and V&V

When we apply the digital technology into the reactor protection system, we have to consider not only conventional design criteria but also newly stated design criteria for safety software into design of the system.

Basically, the functional design of the digital reactor protection system is almost the same as the design of the conventional system and conventional design criteria is applied.

But, concerning about the software, recently stated domestic design guideline (JEAG-4609) is applied to the design process and the V&V activity.

In this JEAG-4609, design and manufacturing process shown in Fig.-2 is defined and the V&V activity should be applied to each stage of design and manufacturing. (process in JEAG-4609 is similar to the process in ANSI/IEEE-7.4.3.2)
3. Consideration in the design of the system architecture

4 channel - 4 train system configuration is applied to the digital reactor protection system as shown in Fig.-3 and this configuration is same as the conventional system.

Hardware components, software modules and software tools used in the digital reactor protection system are same as the non-safety system which have sufficient operating experience, therefore reliable safety software is able to produce.

About hardware portion of the system, separation and isolation among redundant channels or trains is obtained by physically separated cabinets or power sources same as the conventional system, in addition unidirectional optical fiber data links also provides means for isolation.

About software portion of the system, the same software is installed among redundant channels or trains because the safety software has sufficient reliability through V&V activity and does not affect redundancy of the system.

Distributed processor architecture is applied to the design of each channel or train to reduce effects of a common mode failure of hardware or application software.

For example,

(1) Reactor trip function is separated from ESF actuation function, by the use of deferent micro-processor in signal processing and voting logic, so as to both safety functions do not lose at the same time.

(2) Signal processing for reactor trip functions are divided into 2 groups of micro-processors. And 2 out of 4 voting logic for reactor trip function is composed of 2 different micro-processors and dynamic trip logic circuit which has high capability for fail-safe.

So even in a common mode failure of redundant microprocessors, entire loss of reactor trip function is avoided.

(3) Functionally distributed micro-processors described above can operate asynchronously to each other, so they can execute its operation regardless of the operation of the other micro-processor.
4. Consideration in the design of the software

As mentioned above, reliability of the safety software is obtained by tough V&V activity.

But some design criteria shown below are applied to the software design itself to carry out the V&V activity effectively.

(1) Safety software should have single task architecture and execute the task in a fixed time interval. Interrupt operation should be inhibited.

(2) The system software which controls or manages the system operating mode should be separated from the application software which describes the function of each system.

(3) The system software should only have required features for the I&C system in the nuclear power plant. (Dedicated system software for nuclear application should be used)

(4) The system software should have modular and structural architecture, and each module should be written in high level language. (PL/M language is used)

(5) The application software should be written as combination of functional modules (sub-routines) which represent functional element of the I&C system of the nuclear power plant.

(6) Each functional modules should be written in high level language. (PL/M language is used)

(7) Combination of the functional modules should be visible in a graphical image like CAD. Graphical symbols which represent functional modules are put on the CRT of software tool, and each module is connected together with directional lines on the CRT. (We call this method as Problem Oriented Language (POL))
5. **Consideration in the environmental qualification**

Environmental withstand capability of the digital reactor protection system is considered in the design according to some domestic standard on seismic, radiation etc.

In addition, some conditions as seismic design are individually checked site by site.

There are no domestic standard for EMI in the nuclear power plant, but based on some investigation and analysis for magnitude of the noise or surge in the nuclear power plant, we choose some standards about thunder impulse and electro-magnetic surge, and applied them in the qualification test.

There are no domestic standard for RFI in nuclear power plant also, but we carried out qualification test with some handy talkies used in operating nuclear power plants.

These EMI and RFI qualification is same as the qualification for the non-safety digital I&C system in operating nuclear power plants, so judging from satisfactory operating experiences we think that these qualification is applicable to the digital reactor protection system.
### Fig. 1 History of the I&C Technology

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>- Signal Processing</td>
<td>• Analog (Box Type)</td>
<td>• Analog (Card Type)</td>
<td>• Digital (Non-Safety)</td>
<td>• Digital</td>
</tr>
<tr>
<td>- Protection Logic</td>
<td>• Magnetic Relay</td>
<td>• Solid-State Circuit</td>
<td>• Solid State Circuit</td>
<td>• Digital</td>
</tr>
<tr>
<td>- Control Sequence</td>
<td>• Magnetic Relay</td>
<td>• Magnetic Relay</td>
<td>• Digital (Non-Safety)</td>
<td>• Digital</td>
</tr>
<tr>
<td>Application of Digital Technology</td>
<td>• Rad Waste System</td>
<td>• Main Control System (Control rod, Pressurizer, etc.)</td>
<td>• Main Control Board</td>
<td>• Safety System</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• Non-Safety System</td>
</tr>
</tbody>
</table>
System Requirements of the Digital Safety System

Requirement Specification of the System Design

Requirement Specification of the Hardware & Software Design

Hardware Design & Production

Software Design

Software Production

Hardware & Software Integration

Validation

Software V&V plan

Verification 1

Verification 2

Verification 3

Verification 4

Verification 5

Final Products

Verification 1: Specification verification of system design requirement
Verification 2: Specification verification of hardware & software design requirement
Verification 3: Software design verification
Verification 4: Software production verification
Verification 5: Verification of hardware and software integration

Note 1) indicates the scope of design and production activity.

Note 2) indicates the scope of V&V activity.

Fig. 2 Verification & Validation Flow
Fig. - 3 System Architecture of the Digital Reactor Protection System
Regulatory approach in future licensing of Computer Based Systems

Hungarian practice

by

K. Hamar

Hungarian Atomic Energy Commission
Nuclear Safety Inspectorate (HAEC NSI)


Introduction

I&C Reconstruction design activities had already started in the Hungarian 4 unit NPP. The properties of the considered new digital technology are challenging the Hungarian Nuclear Safety Inspectorate to decide in the near future on licensing structural and global modifications, including the - so beleived - "unchangeable" 4 level Protective Action System of the WWER440/213 units.

The needs of privatisation in the energy industry lead to the elaboration of new Law on Atomic Energy, which will be followed by a new series of Safety Codes, in 1996.

Authors of the I&C Volume are facing at the same time the requirements of presently ongoing reconstruction design activities. One of the greatest challenges is to handle the Computer Based Systems. The topic of software handling and the required test efforts in the manufacturing and commissioning phase seems to be the subject of hard negotiation due to the lack of developed national standards.

The living international examples are demonstrating high sums in budget and many engineer years have to be spent to manage licensing procedures. Due to the limited resources there is no possibility to prescribe strict rules. The intention is that, the Hungarian Regulator has to provide guidances for licensing, and at the same time to represent the so called knowledge base of the Regulator.
In the new structure of the Hungarian Safety Code only the

- Procedural Rules
- Design Principles
- Quality Assurance Rules

are mandatory.

The rest of the Code remain in the Guide category. This fact can relieve the burden of the authors, and the responsible regulatory staff, and at the same time this solution is conform with the legal frame.

In the Hungarian legal frame Applicant has the right to prove the fulfillment of any requirements on the basis of standards and guides are different from the Hungarian Code. This is beneficial for the applicants, but they have to face that fact, a different basis from that, the regulatory Evaluator is prepared, can lead to significantly longer time and higher cost in the evaluation procedure. However the new Law on Atomic Energy limits the duration of licensing procedures in 6 months.

I&C Safety Code
New Volume

Safety Categories

According to December 1995 Regulatory Decree, the Hungarian NPP units Machinery, Instrumentation and Control, and Electrical Systems and equipment should be assigned to safety classes. According to the principles, which are very closed to the IAEA guides, the classes have to be established in accordance to the importance of the safety functions and the associated systems and their equipment. PSA support is planned in the future to verify classification.

All the systems and their equipment, including I&C and Electrical Systems have to be classified into four (4) safety classes.

The No.1. class is occupied by the primary circuit machinery pressure elements, like reactor wessel and pipes, inside the primary circuit boundaries.

There is no need to categorize I&C or Electrical System, and their safety functions to safety class No.1.

In accordance with the above statement, the categories A, B, and C of IEC1226 standard are assigned to the Safety Classes 2., 3., and 4.
The representation of the IEC1226 standard in the I&C Safety Code Volume provides a so called "bridge" to the IEC set of standards. On the stabilized technical areas the Code lists IEC standards, on the quickly developing areas the interpretation of minimal requirements is intended. This interpretation is planned to be realized in appended handbooks to the I&C Code Volume.

The regulatory staff studied IAEA, US NRC documents, had a short overview on French regulations, and learned a lot from German practices, and recognized the need of the next appended Handbooks.

- Software Quality Assurance Handbook
- Handbook on Aging and I&C Equipment qualification
- Type Licensing of "from-the-self" equipment

The new legal frame is going to establish the "Type License", which hasn't been existing before.

Software Quality Assurance Handbook

On the basis of IEC1226, IEC880, IAEA-IWG-NPPCI-95/14 draft on V&V, and IEEE SW standards the Handbook describes minimal requirements to guide applicants what can be the common basis of understanding in the reconstruction activities we are looking forward.

As we have learnt from German and French practice, the IEC basis is accepted, the IAEA draft is very wise relating to the safety categories of IEC1226, and expressing the need of different handling of New, Existing, Proprietary and Configurable software. For practical advices one can scan the IEEE SW standards.

As a conclusion we can state, the translation, interpretation and citation do not work without a harmonization effort with mandatory part of the Safety Code, and the legal frame.

This harmonization effort has to resolve also the problem of the financial and human resources, if the Regulator has to calculate with limits.

As a practical conclusion we see, in a class "A" System reconstruction the primary SW components, which are assigned also to class "A", can be only Existing Accessible or Configurable SW. (Definitions are in the IAEA draft)

Otherwise there is no chance in Hungary to proceed with classic V&V in limited time in the frames of a purely national project.
The Body of the SW QA Handbook

This Handbook at the present is the first document on SW to be published by State Authority Organisation in Hungary. On the nuclear field it also fills gaps, because up to now the old licensing rules did not handle the programmed or digital matter of I&C, if it had. If there will be a translation of the Handbook from Hungarian language, Seniors of I&C software can recognize easily the roots it wants to be conform with.

Chapters

- Description and necessity of V&V, with definitions and references
- Competence, V&V organization aspects, responsibilities
- Safety Classes and Types of Software
- Vendor Qualification
- Classic SW Life Cycle Phases and Corresponding V&V activities
  input documents and output documents by phases
  coding conventions and restrictions
  modification and maintenance
  configuration management
- Safety Critical SW Additional Requirements and Verification of them
- Validation and Acceptance in safety categories A,B,C
  New SW
  Existing Accessible SW
  Proprietary SW
  Configurable ??? (further discussions)

Appendixes

- SW Quality Models (Boehm and McCall)
  list of qualifying aspects
  list of SW quality features
  list of SW characteristics
  SW Metrics (Hungarian tool Qualigraph)
• Program Design Language short description
top-down design method
structural programming
language elements, keywords
control structures
examples
• Coding conventions and rules in 3 classes of importance
• Check Lists (examples) against
  System Requirements Specification
  Computer System Specification
  SW Specification and Design
  ??? (further discussions)

Safety Critical SW Additional Requirements

In addition to classic SW Life Cycle Phases and Corresponding V&V activities we need prescriptions on the conscious behavior of the system developers, and V&V personnel. The Regulator needs that confidence, the developer and V&V team know clearly the consequences of safety system errors. Authority inspected examinations on NPP technology and Safety Code can partly guarantee that level of knowledge, and despite of the non-popular matter of authority controlled examinations, the examination can help to maintain the confidence.

In addition to the System Requirements Specification in the frames of Hazard and Risk Analysis designer have to define what is error, what is malfunction, what is catastrophe. Prescriptions needed on the accepted frequencies of prohibited system states, while ALARA have to limit the required efforts in the countermeasures.
In consequence, error detection and internal and external (to the computer based safety system) event handling functions should be realised. In the highest safety class considering the external events the system designer have to calculate with the list of PIEs, and the frequencies of them are described in the plant specific PSA Level-1 study.
In the case of internal events after a preliminary system design a Fault Tree Analysis and a Failure Mode and Effect Analysis (IEC 812,1025), and the same time feeding back of the discovered failure possibilities and sensitivities are
required. This loop have to be well documented, in order to maintain the confidence in the final structure of hardware and software.
The visibility of this kind of iteration maintains better the confidence than the submittal of "bullet-proof" final versions.

Conclusion - Present Status

The SW QA Handbook now is in the final phase of elaboration. The existing text represents the above mentioned international guides and standards, and the opinion of a little group of experts from the aspect of requirement minimalization, what we practically need.

The elaboration project was prolonged once, with 3 months, originally the submittal of the final draft was scheduled in November 1995.
The final draft doesn't exceed the 200 hundred pages yet, however the harmonization with the mandatory part of the Safety Code is a near future task, and the wide negotiations can lead to additions done by regulatory staff.

The wide acceptance can be a function of the statements on the required test efforts, due to that fact, the nuclear operator and cooperating Hungarian and international suppliers see not only the benefits, but the costs of testing also.
To achieve consensus, the intended regulation doesn't speak about that question, who and where should provide test facilities, so the operator can calculate also with the test laboratories of his suppliers, planning the cost effective way in the licensing procedure.

The measure of the required test efforts in the different safety classes hasn't been fixed yet. In this question the guide status of the I&C Safety Code gives the benefit to state approximate test coverage values of acceptance without very serious legal consequences.
The final answer will be influenced by the future ranking of probabilistic approach. In the present frame PSA is a subsidiary tool for the deterministic evaluation, so it concludes, good test coverage values are requested in the highest safety class.
WWER440/213 units 4 * 440 MW

I&C Safety Systems based on relay logic
In error conditions power restrictions

Safety class No.2. IEC1226 "A"
Spurious reactor scrams: 0.5-1/year/unit
Practically safe, test period 6 week
Due to the management of aging and the human effort
in maintenance degradation tendency can not be discovered
up to now

I&C Data Acquisition
upper level
older generation IBM and PDP-11 clones
USSR "CM-2" and Hungarian TPA-11
lower, technological level
Z80 cabinets
Intel 80186 cabinets
In error conditions no power operation limitations

Safety class No.3. IEC1226 "B"
Small software modules
Serial line communication
Despite of obsolescence good operational experience
In commissioning phase poorly tested, there was not such
like requirement

Advantage of obsolescence long time operated software
components
Disadvantage decreasing capability for
modifications
younger generation of sw engineers have
are not skilled in old sw systems
there is no policy to maintain the human
knowledge
Forecast upper level will be substituted
I&C In-core monitoring
  upper level micro-VAX
  lower level Motorola 68000 cabinets
  In error conditions power operation limitations
    after 2 hours home consumption power

Safety class No.3. IEC1226 "B"
New systems 1993-96
Large software
  sophisticated operational systems, visualisation and communication
    OS-9, VMS, X-Windows, TCP/IP
  complicated user software
    core calculation and graphic visualisation
    50,000 source lines in Fortran and C
In commissioning phase    functional testing
                          hardware testing
Confidence in the off-the-shelf software components

No capability on the regulatory side to inspect in details
Authority requirement user modules and factory-test
  have to be documented in accordance to IEC1226 "B"
  category and IEC880
Upgrading up to now on piecemeal basis
sensors, transmitters,
electronic limit sensors
local proof cabling
power breakers
etc.
Changing the non-qualified to qualified equipment
No structural modifications

1990-94 nearly 400 permissions to replace I&C equipment in the scope of inspection activity
due to the replacement the NSI issued more then 100 import licences for various types of equipment

Due to the safe operation the NSI doesn't enforce global or structural I&C reconstruction.

In a national project,
Advanced General and New Evaluation of Safety (AGNES)
supported by PSA level 1 study, review by IAEA

I&C was not evaluated as most vulnerable or main risk contributor

Despite of the above evaluation
the NPP management of the Paks NPP recognised the necessity of upgrading
giving a high priority to the operational safety sparing the manpower in maintenance

Problematic fields

- Specification is based on questionable quality original design documentation.
- Operational experience in large digital systems.
- There is no national standard for software engineering.
- Digital controllers reliability data is confidential
FINNISH REGULATORY REQUIREMENTS FOR PROGRAMMABLE COMPUTER-BASED AUTOMATION (I&C) SYSTEMS


ABSTRACT

Finnish Centre for Radiation and Nuclear Safety (STUK) continuously updates its YVL-guides concerning the safety of nuclear power plants. The guide YVL 5.5 "Electrical and Automation Systems and Components at Nuclear Power Plants" has been revised and updated to give the design requirements for programmable computer-based systems and components. General software design requirements for computer-based safety classified systems are presented. Also more detailed design requirements for control room systems and equipment are laid down. The guide YVL 5.5 describes in more details the design requirements of I&C systems to monitor severe accidents following the principal requirements of the guide YVL 1.0 "Safety Criteria for design of Nuclear Power Plants".

The phases of supervision by STUK has also been updated and revised to include the supervision of programmable computer-based systems and components. The required independent analysis to demonstrate and prove the safety and reliability of programmable computer-based systems depending on the safety class is described. The requirements concerning programmable computer-based components and equipment are also discussed in the paper.
1. INTRODUCTION

Finnish Centre for Radiation and Nuclear Safety (STUK) issues detailed regulations concerning the safety and physical protection of nuclear power plants and safeguards by virtue of the Nuclear Energy Act (990/87) and several Decisions of the Council of State. The YVL-guides are rules any individual licensee or any other organization concerned shall comply with unless some other acceptable procedure or solution is presented to STUK by which the safety level laid down in an YVL guide is achieved.

This paper describes those design requirements and regulatory activities, which are presented in the guide YVL 5.5 Draft 3 concerning programmable industrial automation and computer-based systems. The guide is planned to be applied as well for existing nuclear power plants as for possible new plants.

STUK is following the international standardization work on NPP I&C for systems important to safety. For example the general requirements for computer-based systems (Draft IEC 1513) proposed by IEC/TC45/SC45A/WGA3 may be the missing system level document which is needed as well as the supplements to the widely accepted standard IEC 880.

The updating work started at STUK in the beginning of 1993 with several parallel activities. For example the working group collected most of the relevant international standards. STUK also conducted and supervised research work to develop regulatory guidance eg. related to the feasibility of assessment methods for programmable computer-based automation systems /1/. This was possible only with "real" pilot systems. The case studies were supported by both Finnish utilities IVO INTERNATIONAL LTD and TEOLLISUUDEN VOIMA OY (TVO) and the pilot systems were loaned from ABB ATOM AB and SIEMENS AG. The results of the feasibility of the selected methods are reported in STUK-YTO-TR-report series /2, 3/.
As an on-going activity, Technical Research Centre of Finland; VTT Automation and VTT Electronics, are preparing three reports for STUK concerning "Diversity requirements for safety critical programmable automation systems" /4/, "Assessment of Software Reliability on the Bases of Dynamic Test Results" /5/ and "Programmable Automation Systems in PSA–approaches to reliability modelling and quantification" /6/.

Within nuclear industry programmable technology is used for safety–related and safety automation systems. For existing nuclear power plants in Finland, this technology has been proposed to replace parts of the control and protection systems. In Finland there is long experience, more than 15 years, on programmable systems at tens of conventional power plants and chemical industries. The Finnish utilities use microprocessor–based systems also at nuclear power plants in several safety class 3 and NNS (non nuclear safety) systems and applications.

It has widely been recognized that the programmable systems deviate by their properties and behaviour from the conventional non–programmable systems in such extent that their verification and validation (V&V) for safety critical applications requires new methods and practices /7/. The safety assessment can not be based on traditional probabilistic methods due to the difficulties in the quantification of the reliability of computer–based systems. The proposed qualitative arguments are based on sound engineering judgement and represent deterministic rather than probabilistic criteria.

One important advantage of programmable system is their ability to perform continuous self diagnostics at eg. every 10 ms so the failures are detected and can be corrected instantaneously if required. The systems can even correct some type of errors by themselves (eg. bit errors in data transfer). Important is that safety related events in programmable systems like in any other system are informed to the main control room operators.
2. REGULATORY CONCERNS AND GUIDE YVL 5.5

Requirement categories

The nuclear safety requirements could be divided into three categories /8/. It has to be noted that no technical requirements are contained in the Nuclear Act nor in the Nuclear Energy Degree, but those legislative documents provide the administrative framework for nuclear plant licensing and regulation.

The mandatory requirements (first category) are given in the "Decision of the Council of State on the general regulations for the safety of nuclear power plants (395/91), /9/.

The next level of requirements (second category) consists of the YVL Guides issued by the regulatory body (STUK). Those requirements are not mandatory. Alternative solutions can be accepted if supported by firm evidence which demonstrates achievement of equal level of safety. The requirements contained in the YVL Guides and the formal decisions by STUK expressed in the letters to the utilities belong to the same second requirement category.

The new requirements written in guide YVL 5.5 represent the current position taken by the regulatory staff and are communicated to the utility and vendor representatives in discussions on modernizing existing nuclear power plants in Finland. The third category requirements which were written for future advanced NPP's and are discussed in more detail in /14/, are now included in new YVL-guides.

The new requirements for the automation (I&C) systems which are included in the new guide YVL 5.5:

It is evident that the future plants will have digital control and protection systems. The different vendors offer systems with different development histories and
operating experience. Therefore, the requirements to be set for demonstration of adequate reliability have to be adopted individually to each licensing case.

A recommended approach is a well documented system development process which has used formal methods for avoiding programming errors and for facilitating detection of possible errors. Also, it is important that the systems have been developed to tolerate both system and operator errors. Reason for this is based on widely accepted experiences, that the written accurate system specification is the key document from safety point of view. The most risky phase, where most of the errors or deviations introduce themselves, appears when the application software requirements are derived from the system level descriptions. This means that the validation phase is one of the most important from safety point of view.

**Requirements for the control room functions and systems available for the operators are included in guide YVL 5.5.**

In the main control rooms most monitoring and control takes place with CRT's or equivalent monitors. However, it is required that an overview picture of the state of the main systems be permanently provided in diverse displays.

In the Decision of the Council of State on the general regulations for the safety of nuclear power plants (395/91), it is stated concerning the monitoring and control of a nuclear power plant that the control room shall contain equipment which provide information about the plant's operational state and any deviations from normal operation as well as systems which monitor the state of the plant's safety systems during operation and their functioning during operational transients and accidents.

Also the Guide YVL 1.0, "Safety criteria for design of nuclear power plants", issued by STUK requires eg. that the control room shall be equipped with devices that at all times give enough information about the operational state and safety functions of the plant. Furthermore, the control room shall be equipped with alarm equipment which indicate deviations from the normal operating condition, and with appropriate,
properly assured data collecting, processing and display equipment assisting the operators during operational occurrences and under accident conditions.

3. GENERAL SAFETY DESIGN PRINCIPLES ACCORDING TO YVL 5.5.

The following general safety design principles apply for both electrical and automation systems and components and do not depend on the applied technology. The principles follow from the requirements issued in Decision of the Council of State (395/91) and guide YVL 1.0. The requirements concerning safety design principles, which are discussed in more detail in YVL 5.5, are as follows:

- Levels of protection (Defence in depth—principle)
- Monitoring and control of a nuclear power plant (Disturbance and accident prevention and mitigation)
- Ensuring safety functions (Redundancy, physical separation and diversity principles)
- Safety classification of systems and quality assurance
- Electromagnetic compatibility (EMC)
- Protection against overvoltages
- Environmental qualification
- Identification system of process equipment and controlled variables.

In guide YVL 5.5 STUK is not directly referring to requirements or recommendations in any standard or guide but is requiring the licensee to present those national and international standards which the design is based on. Formal requirements for the documents are given in guide YVL 1.2 /10/.

4. DESIGN REQUIREMENTS FOR I&C SYSTEMS IMPORTANT TO SAFETY

The requirements can be derived from the requirements issued in the above mentioned top level documents. The following design principles and requirements
are for protection systems, which are classified as safety class 2 systems according to guide YVL 2.1 /11/.

The quantitative design goals for safety functions are given in guide YVL 2.8 /12/. These goals have to be taken into account in design of protection systems and it has to be shown how these goals have been obtained. The guide YVL 2.8 is under revision partly due to programmable technology to make it possible to use qualitative methods and expert judgement. If they are used the analysis has to be conservative.

The more detailed design principles for protection systems are as follows:

- redundancy (failure criteria, guide YVL 1.0)
- diversity (common mode failure probability, guide YVL 2.7, /13/)
- performance of protection function to end if activated
- physical separation of protection and control systems
- preventive periodic testing/on–line diagnostics
- physical separation electricity supplies
- application of relevant international standards.

These general design principles must also be fulfilled if programmable computer-based systems are used as protection systems. The guide YVL 5.5 includes requirements on system level and application software level. Also the requirements for reliability of the communication bus are presented.

**Design requirements for programmable systems**

The system level specification shall give answers eg. to the following main questions:
- How the proposed system architecture, configuration of the application software and the structure of the hardware meet the requirements of guide YVL 2.7
- What is the reliability design goal of the programmable system and how will it be reached
- What are the response time requirements for the total integrated computer-based system including all system components like communication busses etc.

The application software and possible data base requirements are generally derived from the system level requirements. Formal or semi-formal methods are recommended. For example the usage of function chart or function block sheets to describe the functions of the application software is a semi-formal method. The same applies to a clear description of the data base related to the corresponding application program.

Software design requirements

The guide YVL 5.5 does not repeat the more detailed requirements which can be found in relevant standards. Only some general basic requirements are mentioned like each single program (module, function block) shall be designed in such a way that it behaves always correctly and logically. The program shall be able to perform all the functions which are defined in the functional specification and no unintended functions are allowed.

This principle requirement shall be applied to all software including system, application, pre-existing and other software used in any programmable system. The inspection activities of STUK are mainly directed to the safety of application software.

The software shall monitor itself in terms of executing order and eg. validate the input and output signals. If a failure or an erroneous input/output signal is observed,
a predefined function mentioned in the specification shall be performed. Eg. if the
programmable system detects that it can not perform a specified task, it shall set the
corresponding system, equipment or device from the plant safety point of view to
safe condition and give the necessary diagnostic messages and alarms.

All the programs shall be written and documented so that they can be inspected in
order to assess the fulfilment of the design requirements. The software documents
shall be clear and always up to date. The implementation of the design requirements
and the configuration management shall be appropriate and properly followable for
a third party. The software structure shall be such that the probably necessary
modifications and updating changes can be performed reliably not forgetting quality
and administrative procedures.

How to validate a programmable system

It is required that the validation of the software integration and the total system shall
be demonstrated. The software validation shall be performed according to a separate
validation plan, which includes eg.

- description of the software design and production process, design
  organization, quality assurance and testing procedures
- description of software functions
- report of validation of single programs (modules, function blocks)
- validation plan for the software integration.

The validation plan for the total system shall include eg.

- plan for static analysis
- plan for dynamic testing
- test equipment, tools etc.
- QA of test procedures
procedure to document and analyze the observed deviations and probable errors or failures
- the modification procedure of the application software and the extent of regression testing.

More detailed requirements are presented in the guide YVL 5.5 concerning the expected behaviour of the object system with different static and dynamic test input signals including the erroneous values of them. Also selection criteria of test cases and the number of them shall be justified. The plan shall include also the accuracy requirements for the output signals. The test acceptance criteria shall be proposed.

The validation plan of the total programmable system shall include a plan of necessary trial operation. The licensee shall assess the pre-operation need for a new programmable system and parallel operation need/possibility with the old system to be replaced with the new one.

The independent assessment of the programmable system

In safety class 2 the independent assessment of the programmable protection systems and their software is required based on a separate plan. The assessment plan shall include depending on the case

- the assessment of the software production process
- inspection of the software functions
- static analysis of the application software
- comparison of the source and object code in the case there are not enough good experiences
- dynamic testing of the programmable system.

The need for independent assessing of other safety classified systems will be decided by STUK case by case based on the proposal of the licensee.
The are also some typical requirements concerning the independent organization following the principles in corresponding standards and generally accepted practices.

Communication busses (automation busses)

There is no programmable system without reliable communication busses and/or local area networks and/or fieldbusses. From operator point of view it is required as real-time feedback from his/hers control activities during all plant operational situations. The earlier mentioned general design principles apply for communication busses. The automation busses shall have the required redundancy to perform their tasks reliably.

5. PREINSPECTION OF PROGRAMMABLE SYSTEMS

In general the preinspection activities of STUK concerning systems in safety class 2 and 3 remain as they have been until now. But in the case of a programmable system certain new reports shall be included into the pre-inspection material for STUK. In order to demonstrate that the earlier mentioned design requirements are followed, the following reports or plans are required:

- the requirement specification of the programmable system including software and hardware specifications
- the design principles of the application software
- validation plan
- system integration plan
- plan for the independent assessment
- data of operation experience of similar systems.

The system integration plan shall include e.g.
- system configuration and software architecture
- function charts and the design principles of the function block diagrams (application software)
- priorities, usage of interrupts and cycle time of the application software including information about the database structure
- the integration of all software and hardware to a total programmable system including self diagnostic functions and its influence on the system behaviour.

The safety report of the system shall be supplemented with the functional analysis of the self diagnostics in the case of a programmable system including the coverage of it. Correspondingly the reliability analysis of the system shall include the effects of the self diagnostics functions.

6. PREINSPECTION OF PROGRAMMABLE DEVICES

As a new requirement only such programmable devices shall be used at nuclear power plants of which one can show good enough operational experiences. The operational experiences shall be analyzed and the arguments for them shall be provided. Also a summary of failures and probable faults shall be presented. In the case of a safety classified programmable device, the preinspection material shall include

- report of the software and hardware configuration and the feasibility of the device to the intended task
- the requirements of the application software
- description of the application software development and testing
- self diagnostic features, functions and coverage
- application software documentation and the modification procedures.

7. OTHER ACTIVITIES PERFORMED BY STUK ACCORDING GUIDE YVL 5.5
STUK will perform a commissioning inspection of electrical and automation systems and equipment at nuclear facilities to all those systems and devices, which have been preinspected and accepted by STUK. The new guide YVL 5.5 gives detailed information among other requirements about the commissioning phase and the required test, independent assessment and quality control measures and reports. Also the test results of possible preoperation or parallel operation shall be shown during the commissioning inspection. The procedure for later life-cycle modifications of a commissioned safety classified programmable system is described briefly in guide YVL 5.5 and will be added to guide YVL 1.8 /16/ in more detail.

8. CONCLUSION

Fast technical development of the programmable control and protection systems is a major problem to most of the regulators. While the new systems provide improved monitoring and control possibilities, and even seem generally more reliable than the old ones, the verification of their correct performance in all operational situations is difficult. A complete testing of such systems is not possible in practise. Reliability which can be claimed with certain confidence depends on the number of test cases, and this number grows necessarily very high. Another major problem which is difficult to address is the potential common-mode failure probability.

The principles of the licensing of a safety classified programmable system or devices for nuclear power plant applications is presented in the paper and the concept is based on Draft 3 of guide YVL 5.5, 9.2.1996. The paper describes the design principles and requirements and finally the inspection activities of STUK. There is not yet much experience of the practical application of the new guide, but the experiences will gather rapidly during 1996. It is also necessary to develop so called assessment guidance for the inspectors daily work. If it were possible to develop computer aided tools to improve the inspection work and make it strict and systematic, which could take into consideration all relevant requirements and design guides, it may be worth of a research program.
REFERENCES


12. YVL 2.8, Probabilistic Safety Analyses (PSA) in the Licensing and Regulation of Nuclear Power Plants, STUK, 18 Nov. 1987, (to be revised).


15. YVL 1.8, Repairs, modifications and preventive maintenance at nuclear facilities, 2 October 1986, (to be revised).
LICENSING REQUIREMENTS FOR SAFETY CRITICAL SOFTWARE OF THE NPP TEMELIN I&C SYSTEMS

C. KARPETA
State Office for Nuclear Safety
Prague, Czech Republic

Abstract

A brief description of the NPP Temelin I&C replacement project is presented. The architecture of the innovated plant I&C system is described with a focus on the plant safety systems. Computer based parts of the I&C portions of those systems are the Primary Reactor Protection System, the Diverse Protection System and the Post Accident Monitoring System. Czech Republic legislative platform for the licensing process of nuclear facilities is then outlined and the specifics of the approach to licensing of the NPP Temelin software based safety systems adopted by the licensing authority, the State Office for Nuclear Safety, is summarized. Acceptance criteria set forth for safety critical software of each of the above mentioned systems as well as guidelines for carrying out an on-site audit of the software development process and for performing independent verification and validation of the final software product are described more in detail.

1. Introduction

The NPP Temelin I&C system replacement is one part of a unique project aimed at upgrading to western standards the two units of the Russian designed VVER 1000 plant which are under construction in the Czech Republic. The primary objective is the improvement of the plant safety. This objective is followed by a secondary one, i.e. the improvement of the plant performance and operational flexibility which specifically relates to the design of the plant I&C system.

Functional design of the plant I&C system must incorporate the same control and protection philosophies which are implemented in western PWRs such as defense-in-depth, diversity, independence, testability, etc. At the same time the design must meet or exceed the performance requirements of the original analog system as well as to meet a number of constraints imposed by the existing plant components, actuated devices, layout of the internal structures, rooms, etc.

The plant operator, i.e. Czech Power Company (CEZ) awarded the contract for delivery of the I&C replacement to Westinghouse Electric Corporation (WEC).

The WEC designed I&C replacement is an integrated system comprising both safety and safety-related systems as well as non-safety systems. Advanced digital computer technology is used for implementation of these systems.
The plant non-safety I&C systems include the Plant Control System (PCS), the Turbine Control System (TCS) and the Unit Information System (UIS). PCS and TCS provide process control functions for the secondary circuit as well as non-essential control functions in the plant primary circuit. The UIS processes data from plant protection, control and measurement systems, integrates it and makes it accessible over the plant redundant fiber optic highway to any of the UIS users.

The plant main safety-related system is the Reactor Control and Limitation System (RCLS). The control functions provided by this system are aimed at maintaining the key Nuclear Steam Supply System variables within predefined limits important to the plant operation. The limitation functions prevent challenges to the Reactor Trip System for a number of specified initiating events by
- blocking of control rod banks withdrawal
- sequential control rod banks insertion at nominal speed
- sequential control rod banks dropping.

The plant safety systems are briefly described in the next section.

2. NPP Temelin Computer Based Safety Systems

The plant computer based safety systems include the Primary Reactor Protection System, the Diverse Protection System and the Post Accident Monitoring System

The system level functions of the Primary Reactor Protection System (PRPS) are as follows:
- to provide an automatic reactor trip whenever the limits of safe operation are approached
- to provide automatic actuation of the plant Safety Engineered Features (ESF) to prevent or limit consequences of the plant design basis accident conditions
- to provide information to the Post Accident Monitoring System and Unit Information System so that these systems are able to alert the plant operators to any abnormal conditions
- to provide certain portion of input information to the Reactor Control and Limitation System.

PRPS is divided into three reactor trip divisions and three ESF actuation system divisions. The divisions are physically and electrically separated from each other. The voting scheme is 2 out of 3. Each of the three divisions is composed of the integrated protection cabinet (IPC) which is associated with the reactor trip switchgear, the ESF actuation cabinet (ESFAC), the integrated logic cabinet (ILC), the non-programmable logic (NPL) cabinet, the main control board (MCB) and emergency control board (ECB) multiplexer cabinets and the data highway gateway cabinet. Manual system level command signals from the MCB and ECB are interfaced directly to the IPCs and ESFACs over hardwired I/O lines. Manual control of each of the ESF components in each division is provided through control board multiplexer cabinets to the ILCs over the PRPS data highways.
The PRPS hardware platform is the WEC Eagle family using Intel 486 microprocessors and Multibus architecture. The programming language is PL/M-86.

The Diverse Protection System (DPS) has been provided to further reduce the probability of non-accomplishment of essential safety functions in the case of common mode failure occurrence within the PRPS. Hence, the mission of the DPS in such situation is to prevent severe core damage or reactor coolant system overpressurization for plant design basis events with an expected frequency more than once every 1000 reactor-years. DPS comprises the following subsystems:

- Diverse Reactor Trip (DRT)
- Diverse ESF Actuation (DESF)
- Main Control Room Diverse Monitoring System (MDMS)
- Emergency Control Room Diverse Monitoring System (EDMS).

The DRT and DESF provide automatic protection as the initial response to the DPS design basis events. They consist of three divisions physically and electrically separated from each other with the 2 out of 3 voting logic.

MDMS is a one-division subsystem of the Diverse Monitoring System (DMS) which together with the Diverse Manual Controls implemented as part of the NPL allows the operators in the main control room to manually control and monitor those plant variables that are needed to stabilize the plant following the occurrence of a DPS design basis event concurrent with a common mode failure in the PRPS. None of these short-term actions are needed in the first 30 minutes of a design basis event.

EDMS is a one-division subsystem of the DMS which together with the Fixed Wire Controls implemented also as part of the NPL has been designed for long-term control and cooldown performed from the emergency control room.

The DPS hardware platform is based on Motorola microprocessors and VME bus architecture. The software programming language used in DPS is C-SMART from Alsys which is a safe subset of ADA language.

The Post Accident Monitoring System (PAMS) is a set of instrumentation used by the operators for monitoring conditions in the primary coolant system, secondary heat removal system and containment, as well as for monitoring the performance of the ESFs and other systems. Its functions are: post-accident data acquisition, display and recording of plant variables important to accident mitigation.

PAMS is a modular system consisting of redundant data processing and display subsystems. Appropriate plant variables are monitored and displayed on display pages that are arranged in a hierarchical plant system order to facilitate operator responses to an emergency.

The hardware platform of PAMS is the WEC Eagle family. The software programming language is PL/M-86.
3. Licensing Process

The licensing process that is followed for obtaining the Czech Republic licensing authority, i.e. the State Office for Nuclear Safety (SONS) approval of the NPP Temelin I&C replacement system designed by WEC is basically patterned after the Act No. 28/1984 on State Supervision of Nuclear Safety of Nuclear Facilities, the Act No. 50/1976 on Territorial Planning and Civil Construction Order and the Regulation No. 85/1976 which gives guidance for enforcement of those provisions of the Act No. 50 which relate to construction of nuclear facilities.

Safety requirements applicable to the design of NPP I&C systems are set forth in the Regulation No. 2/1978 on Nuclear Safety Assurance during Design, Licensing and Construction of Nuclear Power Facilities. The basic requirements are formulated as follows:

- the plant I&C systems shall be capable of monitoring, recording and governing process variables and plant’s items important to safety during normal operation, anticipated operational occurrences and accident conditions;

- during accident conditions these systems shall
  * provide information on the actual state of a nuclear facility in order that protection actions can be taken,
  * provide information on the time course of an accident and record it
  * provide information on important process parameter deviations from their limit values;

- protection systems shall be capable of
  * initiating relevant systems in order that design values of process variables are not exceeded,
  * recognizing accident conditions and initiating relevant systems to mitigate the consequences of accidents,
  * overriding the control systems and the operational staff interventions,
  * being manually initiated by the operational staff;

- protection systems shall be designed for high functional reliability and shall meet the single failure criterion through application of redundancy and independence;

- sharing of equipment between protection systems and control systems shall be minimized;

- systems important to safety shall be designed, installed and tested to high quality standards in other that their functional reliability is ensured. This shall be achieved through implementation of a relevant QA program;

- systems important to safety shall be designed to remain functional during natural phenomena which could be reasonably considered or during external man-induced events in order to:
  * shut down the reactor and maintain it in subcritical state,
  * provide residual heat removal for sufficiently long period,
* maintain contingent radioactive releases below site specific limits.

Quality assurance issues are regulated by the Regulation No. 436/1990 on Quality Assurance of Classified Items with regard to Nuclear Safety of Nuclear Facilities. This regulation determines basic requirements relating to technical and organizational measures for implementation of QA of that equipment, machines, civil structures, means of automated control of processes including their hardware and software which are important to nuclear safety of nuclear facilities.

Regulations No.2 and No.436 do not address more in detail issues pertinent to computer based I&C portions of safety systems. Therefore, the following additional approach has been adopted by SONS in the licensing process of the software based safety systems designed by the supplier of the innovated NPP Temelin I&C system:

- they shall be licensable in the country of origin of their design
- they shall meet applicable requirements stated in internationally recognized codes, guides and standards
- the issue of the potential for a common cause failure in the computer based reactor protection system shall be coped with by providing a diverse system capable of taking the plant to safe shutdown conditions and maintaining it there with the assistance of the operational staff.

Practical implementation of this approach works as follows:

- a number of Topical Reports is being submitted to SONS by the licensee to provide information on the design of individual I&C systems as well as on integral development processes of the whole I&C replacement prior to the official submittal of the amendment to the Preliminary Safety Analysis Report (PSAR); these reports are prepared by WEC and they reflect the stage of a system design at the time of the report issuance
- WEC generated inputs to the I&C relevant sections of the PSAR amendment are also being made available to SONS for review and comments
- evaluation of these submittals is performed against Czech and US regulatory requirements as well as against applicable industry standards and international codes and guides following basically the pattern of the US NRC Standard Review Plan
- frequent meetings are being held between SONS and the licensee assisted by its subcontractors including WEC to address concerns, comments and requests for additional information raised by SONS.

4. Acceptance Criteria Set Forth for Software of the Plant Safety Systems

An ad-hoc regulatory framework has been established by SONS for software of the computer based I&C portions of the NPP Temelin safety systems. It is aimed at achieving the following two objectives:

- to ensure that high quality software be produced by the I&C replacement supplier
• to strengthen confidence in high quality of the final software product through independent verification and validation (IV&V) of its safety critical part.

A number of acceptance criteria and guidelines for performing the IV&V activities have been identified for each safety system. They are presented in the sequel.

4.1 PRPS

The following acceptance criteria have been set for the PRPS software:

• compliance of the software development activities, i.e. planning, requirements, design, implementation and integration activities as well as quality assurance (QA), verification and validation (V&V), configuration management (CM) and Safety Analysis (SA) activities at the supplier including factory acceptance tests as well as site acceptance tests and maintenance activities at the plant operator with IEC 880/1986, ASME NQA-2a/1990 part 2.7 and IEEE 7-4.3.2/1993

• carrying out independent audits which provide evidence that the requirements and recommendations of the above mentioned standards have been met and followed during the software development project

• performing independent verification and validation of the delivered software product consisting of:
  * verification of system requirements and their complete and correct translation into software requirements and specifications
  * static analysis of selected parts of the software
  * dynamic testing of the software as a whole;

• this IV&V shall be performed independently of the software developer and shall provide reasonable assurance that the final software product is free from faults that could impair accomplishment of the system safety functions.

The following guidelines concerning the scope and techniques of the IV&V have been developed:

(1) **System software**

Correctness of the system software functions should be verified by evaluation of the supplier information on:
• verification and validation of this software performed within the Temelin project or for other applications in safety systems
• operational experience gained from other applications.

(2) **Application software**

• program modules that do not have to be statically tested are those which meet the following two requirements:
* they are identical to the Sizewell B primary protection system modules which have been subjected to full scope static analysis
* they are used in the Temelin application in the same configuration of higher level functions and under the same constraints as those of Sizewell B

- Sizewell B program modules which have been modified for the Temelin application shall undergo full scope static analysis
- new Temelin specific program modules or pre-existing modules which have not been independently tested shall also be subject to full scope static analysis
- static analysis should be performed using a qualified tool that is different from the one used by the software developer; if a tool employment is not feasible manual static analysis techniques are also acceptable
- dynamic tests shall be performed on equipment which is to be installed at the plant site. Testing of one division is acceptable provided that information flows to and from the two other divisions are realistically simulated. Tests shall cover a number of accident scenarios which are representative for the following two groups of design basis events:
  * postulated low frequency design basis events with potentially serious consequences
  * postulated frequent design basis events with a potential for serious consequences if viewed with respect to this particular group of events.

The division line between frequent and infrequent events is the frequency of occurrence 1E-3 per year. The responsibility for selection of the test scenarios rests with the licensee but they will have to be approved by SONS.

(3) Data

Correctness of data shall be verified against plant specific data. Employment of a qualified tool for this verification is acceptable.

4.2 DPS

Acceptance criteria for the DPS software call for meeting the following requirements:

- conformance of the software development activities, i.e. planning, requirements, design, implementation and integration activities as well as QA, V&V, CM and SA activities at the supplier including factory acceptance tests as well as site acceptance tests and maintenance activities at the plant operator to IEC 880/1986, ASME NQA-2a/1990 part 2.7 and IEEE 7-4.3.2/1993
- special attention shall be paid in the development process to reduction of the potential for common mode failures similar to those postulated in the PRPS software
- carrying out independent audits which provide evidence that the requirements and recommendations of the above mentioned standards have been met and followed during the software development project
• performing independent verification and validation of the delivered software product consisting of:
  * verification of system requirements and their complete and correct translation into software requirements and specifications
  * static analysis of selected parts of the software
  * analysis of the potential for common mode failures similar to those of the PRPS
• this IV&V shall be performed independently of the software developer and shall provide reasonable assurance that the final software product is free from faults that could impair accomplishment of the system safety functions.

The following guidelines for the scope and techniques of the independent verification and validation of the DPS software have been given.

(1) System software

Correctness of the system software functions should be verified by evaluation of the supplier information on:
• verification and validation of this software performed within the Temelin project or for other applications in safety systems
• operational experience gained from other applications.

(2) Application software

• full scope static analysis analysis shall be performed for the application software
• static analysis should be carried out using a qualified tool different from the one used by the software developer; employment of the same tool as the one used for independent static analysis of the PRPS software is acceptable
• if a tool employment is not feasible manual static analysis techniques are also acceptable.

(3) Data

Correctness of data shall be verified against plant specific data. Employment of a qualified tool for this purpose is acceptable.

4.3 PAMS

Acceptance criteria for the PAMS software are as follows:

• compliance of the software development activities, i.e. planning, requirements, design, implementation and integration activities as well as QA, V&V, CM and SA activities at the supplier including factory acceptance tests as well as site acceptance tests and maintenance activities at the plant operator with IEC 880/1986, ASME NQA-2a/1990 part 2.7 and IEEE 7-4.3.2
• performance of independent audits which provide evidence that the requirements and recommendations of the above mentioned standards have been met and followed during the software development project
• independent verification of system requirements and their complete and correct translation into software requirements and specifications.

4.4 Guidelines for the Scope of an Independent Audit

An on-site assessment of the Temelin safety system software development process should be carried out by an audit team independent of the I&C replacement supplier. The scope of the audit should be focusing on the following topics:

(1) Assessment of process implementation

The QA, V&V, CM activities and software safety analysis should be examined. Particular attention should be paid to resolving the questions identified during earlier SONS and audit team review of the pertinent WEC documents.

(2) Assessment of design outputs

A number of individual safety system functions should be selected and the implementation of the associated functional requirements should be traced from top level system requirements to design documentation and testing. Particular attention in the functions selection should be given to those reactor trip functions which change with the number of reactor coolant loops in operation. The assessment should also include evaluation of design outputs for these functions and evaluation of the employed development processes, i.e. QA, V&V, CM activities and Abnormal Conditions and Events (ACEs) analysis.

This assessment should also include review of samples of the diagnostic software, the automatic surveillance testing provisions and provisions for confirming system return to service.

(3) Assessment of design output translation to executable code

The documentation of the process for qualifying tools should be inspected. Both commercial off-the-shelf software products and developer’s in-house software products should be covered.

5. Conclusions

Nuclear safety regulations which are in existence in the Czech Republic at the time of the NPP Temelin I&C replacement project do not address specifically the area of digital computer technology application in safety or safety-related systems in nuclear facilities. Therefore, the licensing authority developed an ad-hoc approach to the assessment of safety critical software of the three major computer based safety systems of the plant. This approach invokes both the existing international standardization as well as the experience and practices in the area of safety critical software development and assessment in countries where advanced computer technology has already been applied for accomplishment of safety functions either in
new NPP projects or in I&C upgrade projects in the operating NPPs. It is believed
that it will contribute to further enhancement of confidence in high quality of the
innovated I&C system and safety of the plant.

References

Temelin PBZ Chapter 7 Instrumentation and Control
WEC document TEM-I&C-LICEN-015 Rev.2a, April 1995
Current Status and Licensing Experience of Computer-Based Safety Systems in Korea

WON-YOUNG YUN

Korea Institute of Nuclear Safety
Taejon, Republic of Korea, 305-600

Abstract

The Korea has carried out a nuclear plant standardization program to innovate the nuclear plant safety and economy. One of the key elements in the standardization program is the adaptation of the computer-based safety system. And also, the instrumentation and control upgrade program to resolve the component obsolescence issue in conventional nuclear plant operations was proposed by Korea Electric Power Company (KEPCO) in early 1990s. Thus the Korea Institute of Nuclear Safety (KINS) as a Korea nuclear regulatory organization is currently confronted with the pressure to establish the regulatory position and to support the KEPCO’s safety improvement efforts. From these viewpoints, the current status and the licensing experiences of KINS related to the computer-based safety system are briefly discussed.
1. Background

The Korea nuclear industry has limited operational experience with digital computer safety system technology. However, like other countries, practical applications of computer technology to safety systems in nuclear power plant have increased rapidly in recent years. Among of them, the Korea Standard Nuclear Power Plant (KSNPP) project which is undertaken by Korea Electric Power Company (KEPCO) is one of the examples in those movements. A key element in the KSNPP project is a I&C upgrade program which is explained as adaptation of computer-based system.

The Korea Institute of Nuclear Safety (KINS), government established nuclear regulatory organization, has recognized that the computer-based system design and/or the digital upgrade can enhance the operational safety and the system reliability of nuclear power plants when properly implemented. This is based on the fact that the modern digital technology can offer the potential of greater system reliability and operational safety through the use of reliable digital component and adaptation of some design features such as automatic self-testing, diagnostics and automatic calibration capabilities, etc..

Therefore, to improve the reliability and safety of the future nuclear plant design, the KINS has carried out a research project, called development of regulatory guideline on the next generation nuclear power plant, to interact with the Korea nuclear industry recently. The final goal of this project is the preparation of safety review and inspection guidelines on computer-based safety system in Korean nuclear power plants. It is expected that well-defined Korean regulatory guidelines can be issued after the project termination of late 1990's.

In the mean time, the KINS established the regulatory position that conceptual design changes of the safety-related system in nuclear power plants should be justified so that the reliability and quality of the proposed system will not be reduced in comparison with the conventional system design. Therefore, there has been a lot of controversies due to lack of well-established procedures or methodology to assess the reliability of computerized control and protection systems in Korean nuclear power plants.
2. Current Status of Computer-Based System Applications

2.1 Digital Interposing Logic System in Yonggwang Plants

The Yonggwang nuclear plant units 3 and 4 (YGN 3&4) are 2-loop type pressurized water reactors provided by ABB Combustion Engineering with an electrical output of 1050 MWe per each unit. These plants were started to be constructed in the late 1989 and the commercial operation of YGN 4 plant was started in January, 1996. The NSSS design for YGN 3&4 plants is an application of ABB-CE's standard System 80 design. The instrumentation and control system design in YGN 3&4 plants, however, incorporates the evolutionary features of the system 80 design such as digital interposing logic system (ILS) in order to enhance the operability and maintainability of the plant.

From a point of the system configuration, the ILS is a integrated microprocessor-based control system which was developed by Forney International Inc. with the AFS-1000 system. The AFS-1000 ILS has master controllers, master I/O processors and control loop circuits to control the field devices such as motors, valves and breakers etc., which are shown in Fig. 1.

The software program for the master controller, the master I/O processor and control loop circuit consists of a firmware patch panel. Those firmware programs are classified as various executive programs and logic modules. The executive program is used for control functions such as power up, initialization, self-test, I/O control and inter-communication within the system. The logic module is used to provide programmable logic functions, internal timers, memory cells, and flags etc. The firmware program is written in Intel 8085 assembly language using top-down modular design techniques.

The main function of ILS is the plant control system operation but some part of the system operation is closely related to the safety function actuations. Thus the ILS should be designed and fabricated with the class 1E qualification program. However, the hardware and the software of the ILS was developed originally with the Forney's equipment qualification program which was internally used for the commercial product fabrications.
2.2 Programmable Digital Comparators in Wolsong Plants.

The Wolsong nuclear power plant units 1, 2, 3 and 4 are CANDU-PHWR supplied by Atomic Energy of Canada Limited (AECL). Each of them has a net electrical output of 600 MW(e) approximately. The commercial operation of Wolsong 1 plant was started in 1983 and the construction of Wolsong 2, 3 and 4 plants were started in 1991 and 1992 respectively.

In the CANDU-600 plant engineering, automatic reactor trip with programmable digital comparator (PDC) was firstly introduced to the Canadian Pt. Lepreau nuclear plant and Gently-2 nuclear plant in early 1980's. After then, this design was evolutionanyly implemented to the Wolsong 1 plant which has been under construction at that time. Therefore, it was a retrofit since the change was introduced in the latter stage of the plant construction. However, in the design of Wolsong 2, 3 and 4 plants, the more advanced type of the PDC system was introduced to reflect the new regulatory design requirements for the reactor shutdown system (SDS).

Generally, the CANDU-600 nuclear plant has two independent reactor shutdown systems so called as SDS 1 and SDS 2. Each of two reactor shutdown system has three trip channels. The channel trip logic for one of these channels, channel D, is illustrated in Fig. 2. Therefore, with three channels in each of two shutdown systems, there are 12 PDCs in total. As shown in Fig. 2, the digital outputs of PDCs drive relays in the channel trip logic as for other trip parameters, and internally-generated variable set points are displayed on control room panels.

The hardware of the PDC, which is composed entirely of commercial-grade components, was qualified in much the same way as analog equipment, plus additional testing appropriate to digital systems such as electromagnetic interference (EMI) test, an elevated temperature and humidity test etc. Seismic qualification was also performed by testing the PDC in its cabinet on a shaker table. In the Table 1, the hardware design specifications and other features related to the PDCs in Wolsong 1, 2, 3 and 4 plants are illustrated.
2.3 Digital Level Controller of Steam Generator in Kori Plants.

The Kori nuclear power plants units 1, 2, 3 and 4 are the pressurized water reactors supplied by Westinghouse company. The Kori units 1 and 2 are 2-loop type plants which have electrical power output of 600MWe and the Kori units 3 and 4 are 3-loop type plants which have electrical power output of 900MWe approximately. The commercial operations of Kori unit 1, 2, 3 and 4 were started in 1978, 1983 and 1985 respectively. Among them, the Kori 1 plant is the oldest nuclear power plant in Korea. The safety-related instrumentation and control systems in Kori nuclear plants are comprised with the analog electronic devices and the relay logics. Thus, the system reliability can be assured by adaptation of three or four redundant channels per each protection parameter.

However, unplanned reactor trips during low power start up operation of Kori unit 1 were the important safety issue in the late 1980's. Most of those trips were due to the unstability of analog level controllers in the Steam Generator Control System (SGCS). The analog level control diagram is shown in Fig. 3. As shown in Fig. 3, steam generator water level is controlled by two valves, the feedwater control bypass valve (FCBV) and the main feedwater control valve (MFCV). Here, the PI controllers are used for the compensation of non-linearity and time-delay arising in plant dynamics.

In the other aspect, the Korea Electric Power Company (KEPCO) had recognized the need for upgrading the I&C system in the Kori 1 nuclear plant to resolve the obsolescence issues. As a result, the KEPCO started a research project to improve the controllability of the SGCS in 1989 and the first research result was proposed to the KINS in 1992.

The KEPCO's resolution, which is shown in Fig. 4, can be summarized as the microprocessor-based controller implementation to the SGCS with parallel operation of the existing analog controllers. Therefore, it is expected that the various types of design modifications in the I&C system of the existing nuclear plants will be proposed to the KINS for the economic plant operations.
3. Safety Review Activities and Results

3.1 Safety Review Experiences on Digital ILS

In the safety review of the AFS-1000 ILS in YGN 3&4 plants, there has been a lot of concerns in the software system design. From the point of nuclear plant regulation, the digital ILS is the first application of the computer-based logic system in Korean PWR plants. Furthermore, the AFS-1000 system was originally developed for the non-nuclear plant applications in early 1980's. Therefore, evaluation on the software design process and the possibility of common mode failures due to a software programming error was a key issue in the licensing review process.

Major safety review activities and results can be summarized as follows:

Firstly, we reviewed the AFS-1000 software development process and software design team organization. Specifically, we verified that the overall software development effort is coordinated between different departments of a supplier so that the complete software package is internally confirmed with the original design specifications.

And we reviewed the manufacturer's quality assurance program and related documentations: We reviewed the manufacturer's software verification & validation program, the manufacturer / licensee interface program, and the software problem / error reports etc. In relation with these regards, multi-disciplinary design review team was dispatched to the manufacturer site.

And also we reviewed the detailed design documentations such as design specifications, source code listings, various technical manuals and test reports etc. To perform this work effectively, specially designated software engineers were participated in the licensing review period.

As a result, we found that all functional characteristics of the complete design were not documented in such a way that the licensee can maintain the system independent from the manufacturer. Therefore, we required to generate the supplemental documentations listed in the Table 2.

Secondly, we analyzed the system architecture from a point of communication data transmission. Data sharing in the digital system may bring out a critical safety issue related to the common mode failure (CMF) in the general case. concerning to this regard, our regulatory position is that
the data sharing in the communication network of the ILS should be minimized. However, in the case of YGN 3&4 ILS, transmission of interlock signal is controlled by the master logic controller. Therefore, we required that the interlock signals between loops should be implemented by the hardwired logic additionally.

Lastly, through the evaluation of overall system configuration, we found that the YGN 3&4 ILS had not reflected some design features detailed enough to defend against the CMF. Our regulatory position is that on acceptable defence-in-depth design concept for the PWR plant should be followed to SECY-93-087 published by the US NRC. Therefore, we required to perform more stringent EMI test and to install of a backup panel for safety critical functions.

3.2 Safety Review Experiences on PDCs

In the design review of the Wolsong 1 PDC system, the main concerns were concentrated to the design provisions for enhanced system reliability and the operational experiences of reference plants, Pt. Lepreau and Gently-2. We had reviewed lots of documentations and had several licensing review meetings with the designer groups.

The licensing review efforts performed by the KINS staff and the resulted findings can be summarized as follows:

Firstly, we reviewed the system architecture and the software program contained in the PDC. Generally, it is widely recognized that the most significant factor in the reliability of the computer-based system is the complexity of the system. Therefore, to attain the high reliable system, both the hardware system and software program should be kept to the minimum level of complexity required to carry out the safety function. In this regard, the PDCs have the simple software program with minimal use of subroutines and elimination of interactive loops. No operating system and no interrupt function were used in Wolsong 1 PDCs. Additionally, the software occupies about 3K words of PROM (Programmable Read Only Memory) and it uses a scratchpad of about 100 words of RAM (Random Access Memory) only.

Secondly, we review the operating experience of PDCs in Canada. From 1982 to 1990, the Pt. Lepreau plant and the Gently-2 plant have each 8 years of operating experience with PDCs. During this period, the Pt. Lepreau had experienced a total of 58 failures of PDC units and the Gently-2 had experienced a total of 70 failures. The root cause of those failures were the
major issues in the licensing review period. However, it was found that all of these failures were not safety significant failures and the failure rate was lower than the conventional systems. As a result, we concluded that the Wolsong 1 PDC system could be acceptable for issuing operational license.

Afterward, we have reviewed the another types of PDCs, which were implemented in Wolsong 2, 3 and 4 plants. Our safety review will be continued until September, 1996. However, it is expected that those systems can be acceptable since the system diversity and the operational reliability are enhanced compared with the Wolsong 1 case.

3.3 Safety Review Experiences on Digital Level Controllers

The safety review for design modification of SGCS in the Kori 1 plant was treated as the unreviewed safety question endorsed in U.S. 10 CFR 50.59 regulation. The utility guidelines on digital upgrade are reflected in the EPRI's documentations such as NSAC-105 and NSAC-125, etc. Thus the detailed design documentations and the experimental test results related to the design modification of the SGCS in Kori 1 plant were submitted for our licensing review.

The major safety review activities and the results can be summarized as follows:

We analyzed the system configuration and the interface system design firstly. Specifically, signal isolation between existing analog control system and the proposed digital control system was reviewed intensively. As a result, we verified the hardware system integrity.

After then, we reviewed the functionality of the digital control system. To do this, simulation study results were analyzed. The simulation system for the Kori 1 steam generator was developed by KEPCO. With these results, we verified that the system performance was improved by using of digital control system.

Additionally, we reviewed the quality of hardware component used for the digital control system. As a result, although the hardware of the Kori 1 digital level controllers was designed with the commercial grade components, it was verified that the high reliable component was selected. Furthermore, since the digital level controllers were used during short period of low power operations with the analog back up channel as shown in Fig. 3, we concluded that the overall system design could be acceptable from a regulatory point of views.
4. Concluding Remarks

In relation with the safety assessment of the computer-based system, most of the safety issues are arising from the complexity of software programming. To overcome those problems, the various kinds of research work to establish the regulatory guidelines have undertaken by foreign nuclear regulatory bodies and international organizations recently. However, because of inherent design features and economic concerns, it is not easy to establish the regulatory rules in practical cases. The situation is not different in Korea.

Therefore, in the mean time, the KINS established the regulatory position to the safety-related computer-based system in nuclear power plants that the licensee should provide the reliability informations, common mode failure provisions, and quality assurance programs to the government. Especially, for the case of the digital upgrade in the existing nuclear plant, the licensee should justify the design adequacy from points of the system reliability and safety.

Our safety review experiences on the computer-based system can be said that the more comprehensive reliability analysis and the more stringent Quality Assurance Program are needed.
Table 1.  
Wolsong PDC Design Specifications

<table>
<thead>
<tr>
<th>Wolsong 1 PDC</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SDS 1/2 PDC</strong></td>
</tr>
<tr>
<td>- Manufacturer: Data General Corporation</td>
</tr>
<tr>
<td>- Hardware Model: MP 100</td>
</tr>
<tr>
<td>- H/W Configuration:</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>- S/W Language: Assembly Language</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Wolsong 2, 3, 4 PDC</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SDS 1 PDC</strong></td>
</tr>
<tr>
<td>- Manufacturer: ABB Power Generation Ltd.</td>
</tr>
<tr>
<td>- Hardware Model: PROCONTROL P13</td>
</tr>
<tr>
<td>- H/W Configuration:</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>- S/W Language: P 10 FBD Language</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SDS 2 PDC</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Manufacturer: PEP Modular Computer of Kaufbeuren</td>
</tr>
<tr>
<td>- Hardware Model: PROCONTROL P13</td>
</tr>
<tr>
<td>- H/W Configuration:</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>- S/W Language: Modular 2 (Advanced Pascal)</td>
</tr>
<tr>
<td>Document No.</td>
</tr>
<tr>
<td>---------------</td>
</tr>
<tr>
<td>A-357880-01</td>
</tr>
<tr>
<td>A-357880-02</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>A-357880-03</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>A-357880-04</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>A-357880-05</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>A-357880-06</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>A-357880-07</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>A-357880-08</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>A-357880-09</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>A-357880-10</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>A-357880-11</td>
</tr>
<tr>
<td>A-357880-12</td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>
FIG1. AFS-1000 SYSTEM OVERALL CONFIGURATION

FIG2. CANDU-600 SDS TRIP CHAIN SIMPLIFIED LOGIC DIAGRAM
**Fig. 3. S/G Level ACS Block Diagram**

**Fig. 4. S/G Level DCS/ACS Operation Block Diagram**
Regulatory Aspect of Digital Safety Protection System in Japan

OECD/NEA International Workshop on Technical Support for Licensing Issues of Computer-Based Systems Important to Safety
Munich, Germany, March 1996

Zen'ichi Ogiso
Safety Examination Support Div., Institute of Nuclear Safety
Nuclear Power Engineering Corporation
17-1, 3-chome, Toranomon, Minato-ku, Tokyo 105, Japan

Isamu Yagyu
Agency of Natural Resources and Energy (ANRE)
The Ministry of International Trade and Industry (MITI)
Kasumigaseki, Chiyoda-ku, Tokyo 100, Japan

INTRODUCTION

The application of micro-processor-based digital controllers has been widely propagated among various industries along with the rapid progress of electronics in recent years. While in the nuclear power plant industry, the application of them has also been expanding gradually starting from non-safety related systems, taking advantage of their reliability and maintainability over the conventional analog devices. After the careful study of the feasibility of digital controllers to the safety protection system and based on the wide experiences obtained through the use of digital controllers to non-safety related systems, the Tokyo Electric Power Company proposed on May 1989 the adoption of digital controllers to the safety protection system in the Application for Permission of Establishment of Kashiwazaki-Kariwa units 6 and 7 (ABWR-1350Mwe each, K-6/7 in short).

This paper describes MITI-ANRE's regulatory activities leading to the adoption of the digital safety protection system. MITI-ANRE is regulatory body for commercial nuclear power plants.
1. Identification of the issues

The digital system consists of hardware and software. The failure modes of the hardware are approximately similar to those of the conventional system, and it may surely be said that the level of the risk accompanied with the adoption of digital devices can be maintained as equivalent to or less than that of analog devices, provided that the same design requirements are imposed to redundancy and channel separation.

As for the software installed in the computer system, all output signals of each independent channel are prone to be affected by any fault simultaneously owing to the common usage of the same software to each independent channel. This is the main issue from the regulatory point of view.

By introducing software to the computer-based system, the system is exposed to potential vulnerability. Therefore it is recognized that appropriate requirements on the computer-based system were necessary in addition to the conventional requirements in such area as redundancy, independence, separation and testability. While, the licensee recognized the necessity of establishing both design and confirmation method to enhance reliability of digital technology as applicable to the safety-related system. The licensees and manufacturers have developed the methods, and initiated the drafting of design guide with support of MITI. At the later stage of these activities, the demonstration test was also conducted to confirm the system integrity and reliability, partially spending the funds of MITI.

2. Activities

2.1 Establishment of design criteria

In Japan, the basic principles for licensing of light water nuclear power plants are defined in the "Safety Design Criteria for Light Water Nuclear Power Plant" authorized by the Nuclear Safety Commission. The basic requirements for the safety protection system, such as redundancy, independence, separation, are described in the Criteria. As for the industry side, the Design Guide for a Safety Protection System "JEAG-4604", established corresponding to the Criteria, covers conventional systems but not covers computer-based safety-related system. The need for industry to establish design guide applicable to computer-based safety-related systems was recognized. The task was carried out by the Electrotechnical Standard Survey Committee, which was established by industry, and in which officials of MITI were registered as regular participants.

The new design guide "JEAG-4609 ; Application Criteria for Programmable Digital Computer System in Safety-Related Systems for Nuclear Power Plants" for the computer-based safety protection system was established by incorporating additional requirements to ensure the reliability of software being installed in computer systems for safety-related systems. The contents are as follows,

1) Channel redundancy of safety protection system
2) Inter-channel independence of safety protection system
3) Separation of safety protection system from non-safety-grade I&C system
4) Testability of safety protection system
5) Tolerance against seismic force and/or environmental conditions
6) Fail-safe design
7) Response time
8) Emergency AC power supply in case of loss of power
9) V&V of software
10) Management of software modification

The requirements from 1 to 8 are similar to the conventional requirements for the analog system, and the remaining two are the newly added requirements solely for the safety critical software.

2.2 Demonstration test

The demonstration test was conducted by the Nuclear Power Engineering Corporation (NUPEC) under the sponsorship of MITI as one of the Safety Demonstration Test Programs for Japanese Nuclear Power Facilities. The test was carried out to confirm the adequacy of the V&V method stated in JEAG 4609 and demonstrated that the integrated system of software and hardware worked well without any troubles or faults. A steering committee composed of researchers and experts supervised the test. One of four independent channels was simulated with software-based devices to be used in the actual plant, while the others were simulated by a simulator arranged for the test purpose.

(1) Software

The software installed in the test system was the same one to be installed in the safety protection system of the K-6/7. Check and review of the software was carried out at each software production stage according to the V&V method stated in JEAG-4609. No trouble nor failure occurred through the series of the tests described below. This result confirmed the adequacy of the V&V method as well as the operational reliability of the produced software.

(2) Integrated system test

The test items of the software-computer integrated system consist of characteristics test, thermal aging test, noise test, vibration test and accident simulation test. The test conditions were determined based on the design requirements for the safety protection system and environmental conditions expected to be encountered during normal and abnormal plant operation. The noise tolerability tests, in particular, were carried out exhaustively since the computer system should work in a low voltage condition. Noise source in nuclear power plants was reviewed and simulated. The simulated noises, such as noises from the power supply system, induced noise, electro-static noise, radio-wave interference and lightning surge, were loaded one by one. The system worked well without any trouble or failure even with these noises exist. This result not only showed the integrity of the computer system, but also led to the establishment of the noise test method. The seismic test is very important in Japan. The computer system was mounted on a shaking table and a seismic load was applied. The computer system worked well
both during and after the seismic loading.

Through these demonstration tests, MITI could accumulate significant data for the safety review of the computer-based safety protection system.

3. Application

The first nuclear power plants that adopted the digital safety protection system in Japan were K-6/7. The safety review had been carried out by MITI for the system proposed by the licensee and granted the license on the condition that the regulatory judgment applied here should be limited to the proposed system only.

These two plants are now under construction. For Unit-6, a series of pre-use inspections are currently at the last stage and the commencement of commercial operation is scheduled in December 1996.

3.1 Safety review of digital protection system

The features of the proposed digital safety protection system are as follows.

The hardware of the digital safety protection system is designed and manufactured in the same manner as the conventional analog system and fulfills the safety requirements for the safety-related systems. Therefore the quality and the reliability of hardware was certified using similar manner as analog system.

As for the software, the language used in the system is symbolic language, POL (Problem Oriented Language). Review was carried out from the view point of compatibility of the systems written in POL and V&V. The characteristics of the software written in POL are as follows;

- program is written with comparatively less kinds of logic elements.
- program is executed toward one-way direction with cyclic execution, but without interruptive execution, thus minimizes the invasion of errors.

Since each logic element is written using symbolic language, the program is easily readable and clear when being reviewed by a third person.

The structure of software written in POL is simple and this enables both verification of the program and confirmation of the validity easy even after the system integration.

MITI admits the inherent characteristics of POL through the review of the documents submitted, considering the fact that the manufacturer had established the system on quality assurance in producing software by POL, based on the experience of software for the non-safety related system. It was convincing that the use of POL is more reliable than the mixed use of various kinds of different conventional software languages.

This means that the most important factor in securing the reliability of software for safety-related systems is that the licensee should establish the appropriate quality assurance measures and put them into practice when producing the software. Consequently it was recognized that the design system and quality assurance plan should be subjected to the review of MITI as a part of the Construction Plan.

As for the manual operation, the conventional design guide requires that the manual scram mechanism must be equipped separately from the automatic safety protection
system. The proposed safety protection system of K-6/7 had this function. So, this manual system was accepted without any additional discussion.

3.2 Inspection

MITI performs pre-use inspections for the system and equipment installed in nuclear power plants under the provisions of the relevant laws and confirms the basic characteristics and function, based on the licensee's self-imposed inspections*1. The same practice was applied for the inspection of the safety protection system. The primary responsibility lies with the licensee to check and review the system during its production according to the V&V procedure. MITI came to inspect essential function of the system after the licensee has completed and installed the system. MITI reviews the results of V&V being performed by the licensee at the final stage of the series of pre-use inspections.

note 1: The regulation and management of the safety of NPPs in Japan is carried out, not only by an authority on the safety review and the structural and functional inspections of the facilities, but also by the self-imposed system of utilities (licensees).

4. Conclusion and Future plan

MITI acknowledged the importance of establishing the QA system of producing and V&V of the software among the licensees and manufacturers for securing the reliability of the software. It should be stressed that the software should be constructed so that it can be easily readable by a third person.

The introduction of digital technology will be expected to contribute to promoting and upgrading the maintenance methods as well as to securing the high operational performance. MITI will continuously evaluate the operating status of the plants which have a computer-based safety protection system and hope for a sound propagation of digital technology in nuclear power plants as well.
RETROFITTING TO PROGRAMMABLE ELECTRONICS IN NUCLEAR POWER PLANTS

Focus on Requirements Aspects

Erik Johansson
Royal Institute of Technology
Dept. of Industrial Control Systems
S-100 44 Stockholm, SWEDEN
E-mail: erikj@incosys.kth.se

ABSTRACT

In most nuclear power plants in operation today, there is an increasing need to replace outdated instrumentation and control equipment with systems of programmable electronics. Despite many advantages that could follow a retrofitting to digital control systems, utilities are still reluctant to replace the old equipment. The reason for this reluctance is mainly related to difficulties in quantifying the reliability of software. Reliability in software depends mainly on the quality and procedures used in the development process of the product. It is well established that finding and correcting software errors and making the required changes is a lot harder during the operating phase than in the early development stage. Therefore, should confidence in the software reliability be established already during the elicitation of requirements. This paper present experiences from field studies made in the Swedish nuclear industry, with the main focus on difficulties in finding and formulating requirements for the requirements specification. The studies have been done in cooperation with the nuclear power industry and the Swedish Nuclear Power Inspectorate.

So far utilities have been making minor replacements of isolated parts of systems. But the improvements regarding I&C have generally consisted of minor items such as modifications of control logics or additions of measurements of new interesting plant variables. These replacements have been basically made on a component level due to deficiencies in the function or obsolescence or lack of spare parts. However, in Swedish NPPs, there are a number of computer based systems in use. There are even a few systems, which comprise a part of the plant protection system, that have been equipped with systems based on programmable electronics (thus the licensing process has required a hardwired backup system). The older NPPs face the need for a long term upgrade plan which observes the overall system structure in such a way that these systems become easy to manage and maintain.

A solution of replacing the old I&C with programmable electronics, has several possible benefits:

- reduced maintains, since digital systems enable self-diagnosits and replacement on-line.
- improved control capabilities, with adaptive tuning, and “drift-free” operation.
- increased reliability and cost efficiency, with the introduction of higher functionality and flexibility.
- higher degree of automation which can eliminate errors caused by stressed humans.
- easier modifications and extensions because software is better geared to this than hardware.

Despite many advantages that follow a retrofitting to digital control systems, nuclear utilities are still reluctant to replace the old I&C equipment. This reluctance is based on uncertainties about the risk of having a misfunctioning software, which could result in a lengthy outage or even costly reactor trips. Furthermore, the major "risk" at the moment is that software-based systems do not get approved by the regulatory authority without having a hardwired backup system.

The commissioning of traditional hardwired equipment for safety-critical systems is done partly by a quantification of its reliability, based on the assumption that failures are independent and appear randomly. However, today there is no possibility of making the same calculations for I&C systems with programmable electronics since software behaves differently than hardware, e.g., software does not age and
failures between different software programs are not completely independent.

As systems based on programmable electronics probably will take a central role in the replacement and modernization of existing I&C systems, the failure of software development efforts has come to the forefront. Investigations have reported that more than fifty percent of all bugs detected in software can be traced to errors made during the requirement stage [Davis 90]. Therefore, confidence in the software reliability has to be established already during the early development stage of requirements elicitation.

1.2 Purpose, Target Readers

The purpose of this paper is to present experiences from field studies carried out by the Department of Industrial Control Systems at the Royal Institute of Technology, Stockholm, Sweden. The field studies have been carried out by investigating several NPPs, I&C suppliers, and the Swedish Nuclear Power Inspectorate. Based on these field studies the paper presents some conclusions and suggestions for further work regarding requirements engineering. This paper should be of most interest to nuclear utilities and regulators but may also be of interest to suppliers and consultants in the area of safety-related processes.

1.3 Outline of the paper

In Section two some definitions from the field of requirement engineering are presented, and in Section three, experiences from field studies among Swedish NPPs are briefly presented with a focus on requirement aspects. The paper ends with conclusions and some suggestions for further work.

2. DEFINITIONS

According to IEEE "requirement" has the following definitions [IEEE 90]:

(1) a condition or capability needed by a user to solve a problem or achieve an objective;

(2) a condition or capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification, or other formally imposed documents;

(3) a documented representation of a condition or capability as in (1) or (2).

Requirements do not only state functions; they can state clearly nonfunctional requirements as well. Additionally, they can even be classified in terms of the goals (defined to guide the system developer in achieving the implementation of the agreed user requirements) and implementation constraints (e.g., use ADA) [Ashworth 89].

Requirement engineering is "the disciplined application of scientific principles and techniques for developing, communicating, and managing requirements" [STEP 91]. The goal of requirement engineering is the production of good requirements specification. IEEE defines a good requirement specification as being [IEEE 84]: unambiguous, complete, verifiable, consistent, modifiable, traceable, and usable during operations and maintenance.

The process of requirement engineering is an iterative activity that involves

• eliciting requirements from various sources,
• insuring that the needs are consistent and feasible,
• validating the derived requirements against the needs of interested parties.

The requirements should be expressed in a form which promotes communication and understanding between different groups of engineers, allows the developer/supplier to determine whether the expressed requirements are possible to implement, and lets quality assurance teams verify that an implementation meets the needs. Requirements should not be vague and untestable. An example of an untestable requirement is "user friendly." "Robust" is another example.

The problem area of requirement engineering includes:

• achieving requirement completeness without unnecessarily constraining system design,
• analysis and validation difficulty,
• changing requirements over time.

Many requirements errors are passed undetected to the later phases of the life cycle, and correcting these errors during or after implementation has been found to be extremely costly [Davis 90].

The requirements specification activity is the most significant part of the overall project in terms of its influence on the quality of the final product. Utilities and developers should therefore concentrate on reducing the requirement errors by improving their methods of requirement elicitation.

3. FIELD STUDIES

Experiences from the field studies point out some potential difficulties regarding the requirements elicitation and specification. In Sweden the general process of producing the requirements specification is mainly perpetuated by the Final Safety Analysis Report (FSAR), which describes the overall picture of the power plant's system requirements, and operational experiences and users' needs.

![Diagram of the procurement process of requirement specification at Swedish nuclear utilities.](image-url)
The requirements that the software must satisfy are identified from the overall system requirements and eventually documented in the requirements specification, see figure 1.

One of the main concerns when new technology is being introduced is that the elicitation activity must also meet the boundary conditions for the actual safety-related part of the target systems. Elicitation must still focus on the creation of requirements, not design activities, in order to adequately address the users' concerns and not just developers needs.

3.1 Over and underspecified requirements

The difficulties in the elicitation of requirements have shown to be mainly due to limited resources at the utilities. The system engineers are very adept in their knowledge about the process but sometimes lack understanding of the computers' capability and their limitations. On the other hand, external resources, such as consultants, are often skilled analysts but sometimes lack knowledge about the nuclear process and its constraints.

The field studies have shown that the basic distinction between the utility's role of describing "what" and the supplier's role of providing "how" is sometimes vague. Suppliers experience that initial requirements proposed by the nuclear utilities are overspecified in a sense that they are too complete (full coverage of user and system needs), burdened with needless design constraints (unnecessary design information). In general these requirements are based on the utilities' own technical estimations and experiences. In many cases these arguments can seem to be well-founded, but experience from other fields has shown that these kinds of requirements often create severe problems and high additional costs [Forsgren 95]. The overall system has a tendency to be more complex, since the suppliers' product family may not fit the initial requirements without special solutions.

However, there are also examples where lack of knowledge and clarity have arisen in underspecified requirements, concentrated on information relevant to the solution development but incomplete regarding user and system requirements. Underspecified requirements may lead to situations where a supplier cannot make a fixed price proposal due to the difficulties in planning for the required resources. This can lead to lack of time and resources in a project which can threaten the software quality.

Utilities express their concern about receiving proposals from suppliers which are hard to understand and complicated to verify that all of the requirements are fulfilled.

3.2 Communication problems

Difficulties with misunderstandings can lead to requirements which are ambiguous, incomplete, and even incorrect because they do not address the true essential needs.

When retrofitting old I&C systems, there is a need to fully understand and clarify their fundamental design criteria and old documentation. Developers and staff with this knowledge have changed employment or are about to retire. Therefore the utilities and the suppliers of the old NPPs in Sweden have started projects for documenting the fundamental design criteria. These projects are: BOKA at Barsebäck, REDA at Ringhals, and RAK at Forsmark.

There are examples from the studies where the language used to express the requirements is a cause of misunderstanding. It may be too formal or too informal to meet the demand of different groups; user and analyst speak different languages.

In the field studies, requirements analysts expressed their concern about difficulties in how to interpret and integrate information gathered from different people with a variety of backgrounds and experiences. For example, common knowledge to one group can be completely unknown to another.

Another difficulty is the large amount of information evoked during the analysis of requirements. This information is not always structured in a way that coincides the scopes of the interested parties. In this process, it is easy to omit "obvious" information. Furthermore, it is possible that conflicting views from different users are not foreseen because the analyst has often poor knowledge of the problem domain.

3.3 Change in requirements

As more experiences are gained about installed programmable electronics standard documents are reviewed and updated, for example the standard document IEEE 603 from 1981 was updated in 1991. In terms of time requirements change, some signal classifications were changed, for example, after the accident at Three Mile Island. Another example is when the users' needs mature because of increased knowledge, and the user requirements therefore change. Environmental changes to the system are another cause of change.

If changes are not accommodated, the original requirements set will become incomplete, inconsistent with the new situation, and potentially unusable because they capture information that has since become obsolete. Requirements do evolve over time and this has to be taken into account when structuring the specification.

4. CONCLUSIONS AND SUGGESTIONS FOR FURTHER WORK

The reason for overspecified requirements may be due to psychological factors such as the engineer's desire to "build things." The results is that high level design and requirement specification are carried out almost simultaneously, i.e., the "How's" cannot be separated from the "What's." It is important, therefore, to have a structured approach in the development of requirements specifications.

The information gathering task has been identified as a difficult task. There is a gap between the system suppliers, developers, and requirements analysts (often engaged external resources with limited knowledge about the application domain and its constraints), and the system engineer who knows the controlled process well but does not have knowledge about the design methods for the development of systems with crucial software components. There is a need for a well-defined conform terminology together with
notations to describe the requirements, for example, by use of formal methods. This would give a "natural" hierarchal structure which could avert misunderstandings and non-verifiable requirements. Increased knowledge and the use of well-structured formal methods could be a useful tool in clarifying many of the mentioned difficulties. Today formal methods are not being used to any great extent in the Swedish NPPs. There is a risk, however, that created formal requirements will be ambiguous for the users. These formal requirements may not be verifiable by the users because they cannot adequately understand the languages. The lack of internal resources and the gap in knowledge are the main reasons for the difficulties of getting formal methods accepted in the nuclear industry.

Many requirements difficulties regarding old NPPs consist of poor knowledge about fundamental design criteria and poor understanding of the computer's capability and the limitations of the procurement process. So far, the utilities have compensated this lack of knowledge by finding the people who were engaged in the project during the first development of the plant. These people have the knowledge in their heads, but can one be sure they remember? The ongoing projects REDA, BOKA, and RAK will hopefully compensate this lack of knowledge. Another important advantage of these projects is that the younger generation of engineers is being introduced to the nuclear process and the system constraints.

In recent years, many of the difficulties regarding computer-based I&C systems have been identified by both the utilities and the regulators. This has resulted in both guidelines and modifications in the utilities' instructions, such that they now take into consideration the software issues of programmable electronics. It would be a strength for the nuclear industry to "synchronize" international standards and experiences and to present clear recommendations on available tools and methods including their benefits and drawbacks. These should be presented with a conform terminology.

To increase the confidence in this "new" I&C technology, it is imperative to understand the overall picture of requirements specification and the development process by strictly using well-defined standards and guidelines. The regulatory authority should concentrate its resources on auditing and reviewing the knowledge about important principles and standards. In this way confidence of programmable electronics can increase.

Further work should be carried out on producing clear, conform guidelines for requirements elicitation and on easing the use of formal methods in planning and specifying the fundamental I&C requirements.

ACKNOWLEDGMENTS

The author would like to acknowledge the industry and the licensing authority that have shared their vast experiences and provided relevant documents. A special thanks to Professor Torsten Cegrell and Doctor Tomas Rahkonen, at the Department of Industrial Control Systems, and Ingmar Tirén from the Nuclear Power Technology Center at Royal Institute of Technology, Sweden.

BIOGRAPHY

Erik Johansson was born in 1967 in Västerås, Sweden. He received his M.Sc. degree in electrical engineering in 1993 from Royal Institute of Technology, Sweden. Mr. Johansson is an IEEE student member and since 1994 has been working towards his Ph.D. degree at the Department of Industrial Control Systems, Royal Institute of Technology, Sweden.

REFERENCES


SESSION 2 - MORNING

HIGH INTEGRITY SOFTWARE - WHAT SHOULD WE DO? WHAT CAN WE DO? WHY DON'T WE DO IT? - Prof. David L. PARNAS, McMaster University

REVIEWABILITY GUIDELINES FOR COMPUTER-BASED SAFETY SYSTEMS - M. Claude Esmenjaud, Schneider Electric

DESIGN REQUIREMENT AND DEVELOPMENT OF SOFTWARE FOR THE DIGITAL SAFETY PROTECTION SYSTEM IN KASHIWAZAKI-KARIWA - UNITS 6 & 7 - Mr. Takaki Mishima, Tokyo Electric Power Company

SAFETY CASES - PRODUCING A REVIEWABLE SYSTEM - Mr. Norman Wainwright, Nuclear Installations Inspectorate (NII)

DESIGN FOR LICENSIBILITY - TELEPERM XS FROM SIEMENS - Dr. Arnold Graf and Dr. Heinz-Wilhelm Bock, Siemens/KWU

APPLICATION OF GUIDELINES FOR REVIEW OF SOFTWARE IN A PROGRAMMABLE REACTOR PROTECTION SYSTEM - Mr. Gustav Dahl, OECD Halden Reactor Project
High Integrity Software

What should we do? What can we do?

Why don’t we do it?

David Lorge Parnas
Telecommunications Research Institute of Ontario
Communications Research Laboratory
Department of Electrical and Computer Engineering
McMaster University, Hamilton, Ontario Canada L8S 4K1

ABSTRACT

How can an engineer fulfill professional responsibilities when building critical systems in which software plays a major role?
Introduction

Safety-critical software highly controversial.

Some: software technology is not mature

Others: no inherent barriers advantages greater than risks

Both sides are correct!

In critical applications:

(1) the software and system design must be carefully inspected.

(2) inspection impossible without precise documentation.

(3) software designers have too much freedom.

(4) discipline must be imposed.

(5) reviewers should not be forced to guess designer’s intent.

(6) safe modification requires precise documentation.
Why is Software A Special Concern?

(1) It never works the first time it is really used

(2) It has no natural internal boundaries

(3) It is sensitive to minor errors - there is no meaning to "almost right". (chaotic behaviour)

(4) It is difficult to test because interpolation is not valid

(5) There are "sleeper bugs".

These are all manifestations of complexity.

They are "inherent" properties, not signs of immaturity in the field.
Why do we use software in spite of these properties?

(1) We can implement things that we could not implement with older technologies.

(2) Digital systems allow relatively easy increases in accuracy.

(3) We get much better information displays.

(4) Software doesn't wear out or deteriorate with age.

(5) Easier to change, especially if there are many installations.
What Should We do?

The lessons that we learned in Engineering School are clear:

- Don’t try to solve a problem until you can state that problem precisely.
- All work must be reviewable.
- In systems, the interfaces between components must be precisely defined.
- Use mathematics, as well as experimental techniques, to evaluate the trustworthiness of products.
- Reviews must be thorough and systematic.
- Testing must be “sufficient”.
Why can’t we do this for Software?

- Requirements documents are vague, wordy, incomplete, and inaccurate.
- Other people’s code is very difficult to understand.
- Module interface documents are vague, wordy, and often reveal implementation details.
- Most programmers don’t know how to use mathematics to analyse software.
- Books and courses on reviews discuss the organisation of the review teams, but now how to do the work.
- We don’t know how much testing is enough!

The bottom line: We need better documentation and systematic, mathematically based, inspection procedures.
Documentation Principles

An observation by Ludwig Wittgenstein:

- "Was sich überhaupt sagen läßt, läßt sich klar sagen; und wovon man nicht reden kann, darüber muß man schweigen"
- "Anything that can be said at all, can be said precisely. Anything that you cannot say precisely, you should not say at all."

What do we need in software documentation.

1. Precision: Mechanical Interpretation, no ambiguity.
2. Accuracy: not "almost right".
3. Consistency: no contradictions, hence no duplication.
4. Completeness: All visible behaviour covered, even when there is a choice.
5. Verifiability: Organised for easy checking, a place for everything.
6. Changeability, design for retrieval, no searching.
Formal Methods: What and Why?

"Formal Methods" is an unnecessary phrase. We are talking about the use of mathematics in engineering. That is nothing new!

1. Inadequacy of Natural Language: Not designed for this.
2. Engineering Traditions.
3. Programs as Mathematical Objects
   - programming languages are mathematical languages
   - they can be described mathematically
   - Computer scientists often fail to distinguish the descriptions from the objects. Engineers never do!
4. Semi-formal Notation
   - N. G. de Bruijn distinguishes "the vernacular of mathematics" from logic.
   - Some things don't need to be said: \( x + 2 \), not \( \lambda(x), x+2 \)
   - Software engineering mathematics should be engineering mathematics.
5. Formally Defined Notation
   - Syntax must be prescribed
   - Semantics must follow syntax and be complete
   - Assume standard mathematical knowledge.
6. Formally Structured Notation, remove unnecessary freedom.

Communications Research Laboratory
Software Engineering Research Group
"connecting theory with practice"

CSA.slides

March 3, 1996
Why apply mathematics to documentation and inspection?

If we can't use formal methods for documentation, we can't use them for anything.

If we can't use them in an inspection process, how can we begin to think about proof of correctness or checking of properties?
How can we document system requirements?

Identify monitored variables \((m_1, m_2, \ldots, m_n)\)

Identify controlled variables \((c_1, c_2, \ldots, c_p)\)

For each scalar variable, \(x\), denote the time-function describing its value by \(x^t\)

The value of \(x\) at time \(t\) is denoted \(x^t(t)\)

The vector of time-functions \((v_1^t, v_2^t, ..., v_n^t)\) will be denoted by \(y^t\)
How can we document system requirements?

Describe the following relations:

**RELATION NAT**

- domain(NAT) is a set of vectors of time-functions containing only the instances of \( m^t \) allowed by the environmental constraints,

- range(NAT) is a set of vectors of time-functions containing only the instances of \( c^t \) allowed by the environmental constraints,

- \((m^t, c^t) \in \text{NAT} \) if and only if the environmental constraints allow the controlled quantities to take on the values described by \( c^t \) when the values of the monitored quantities are described by \( m^t \).

**RELATION REQ**

- domain(REQ) is a set of vectors of time-functions containing the instances of \( m^t \) allowed by environmental constraints,

- range(REQ) is a set of vectors of time-functions containing only those instances of \( c^t \) considered permissible,

- \((m^t, c^t) \in \text{REQ} \) if and only if the computer system may permit the controlled quantities to take on the values described by \( c^t \) when the monitored variables are described by \( m^t \).
Documenting System Design

System Design determines communication with the computers.

RELATION IN

Let"t" denote the vector (i_1, i_2, ..., i_n)
- one element for each of the input registers
- domain(IN) is a set of vectors of time-functions containing the possible instances of r,
- range(IN) is a set of vectors of time-functions containing the possible instances of i,
- (r, i) ∈ IN if and only if i describes possible values of the inputs when r describes the values of the monitored quantities.

It should be the case that,

\[ \text{domain}(\text{NAT}) \subseteq \text{domain}(\text{IN}) \]

RELATION OUT

Let"s" denote the vector (s_1, s_2, ..., s_n)
- one element for each of the output registers
- domain(OUT) is a set of vectors of time-functions containing the possible instances of s,
- range(OUT) is a set of vectors of time-functions containing the possible instances of s,
- (s, s) ∈ OUT if and only if s describes possible values of the controlled quantities when s describes the values of the output registers.
Documentation of “Fail-Soft” Design Properties

Partial failure of the equipment can be described by writing weaker versions of IN and OUT.

Weaker versions of REQ can be described to specify the behaviour that is allowed under undesirable circumstances as described in the weaker versions of IN and OUT.
Documenting Software Behaviour

RELATION SOF:

The software can be described by a relation, which we call SOF.

- \text{domain}(SOF) is a set of vectors of time-functions containing the possible instances of \(i^t\).
- \text{range}(SOF) is a set of vectors of time-functions containing the possible instances of \(o^t\).
- \((i^t,o^t) \in SOF\) if and only if the software could produce values described by \(o^t\) when the inputs are described by \(i^t\).

SOF will be a function if the software is deterministic.
How can we document software requirements?

Software requirements = system design + system requirements

1. REQ($m^t$, $c^t$)
2. IN($m^t$, $i^t$)
3. OUT($o^t$, $c^t$) and
4. NAT($m^t$, $c^t$)

The actual software can be described by

5. SOF($i^t$, $o^t$)

For the software to be acceptable, SOF must satisfy:

6. $\forall m^t \forall i^t \forall o^t \forall c^t \left[\text{IN}(m^t, i^t) \land \text{SOF}(i^t, o^t) \land \text{OUT}(o^t, c^t) \land \text{NAT}(m^t, c^t) \rightarrow \text{REQ}(m^t, c^t)\right]$

Using functional notation:

6a. $\forall m^t \left[m^t \in \text{domain(NAT)} \rightarrow (\text{REQ}(m^t) = \text{OUT(SOF(IN(m^t))))})\right]$
Why these definitions are important.
They end the bickering about whether or not something should be in the document.

One page of definitions, specifies what should be in each document.

It makes it possible to do simple checks on documents, mechanical checks or clerical checks.

(1) If you haven’t identified the variable, you can’t mention it
(2) This forces you to find all the variables.
(3) It enforces simple semantic rules, such as “black box”
(4) It provides precision similar to that used in other areas of engineering.
(5) It leads to requirements documents with two properties:
   • Pilots can read them
   • Programmers can program from them
   • Programmers do not have to read the user’s mind.

Note that we can talk about real-time requirements without special (temporal) logic.
# Reviewers:

| Systems Requirements | Nuclear Plant Safety Specialists  
| Computer System Design Team  
| Licensors |
|------------------------|-----------------------------|
| Computer System Design | Authors of System Requirements  
| Computer System Specialists  
| Software Design Team  
| Licensors |
| Software Requirements | System Requirements Team  
| Software Design Team  
| System Design Team  
| Licensors |
| Software Behaviour | System Requirements Authors  
| Software Design Specialists  
| System Design  
| Licensors |
| Module Structure | System Requirements Authors  
| Software Design Specialists  
| Programming Experts  
| System Design  
| Licensors |
| Module Interface Design | Software Design Specialists  
| Programming Experts  
| Licensors |
| Module Internal Design | Programming Experts  
| Programming Language Specialists  
| Licensors |
| Program Effects | Algorithm Experts  
| Language Specialists  
| Licensors |

Communications Research Laboratory  
Software Engineering Research Group  
"connecting theory with practice"

CSA.slides  
March 3, 1996  
157
Why We Need Tabular Expressions

Computer System Functions/Relations have certain distinctive characteristics.

- Arbitrary set of discontinuities
- Values are tuples, elements of different types,
- - no "typical" element

We have designed tabular forms of expressions to be suited for such relations.

Previously used on ad hoc basis, a formal definition was badly overdue.

Conventional mathematical notation is not adequate for this application.

We need a multi-dimensional notation, tables.

We already know how to read the expressions that appear in the tables.
A Case Study: Darlington Shutdown Systems

Three control systems:
   One normal control system
   Two independent shutdown systems

Safety analysis assumes control system will fail.

Only the shutdown systems are considered safety-critical.

Shutdown systems were analogue and relay systems.

At Darlington they are software controlled.

Each Software System has a simple task

Their designs are "diverse"

The systems are more complex than their predecessors

AECB could not be confident of their trustworthiness.
Why do we need formal methods?

"Shut off the pumps if the water level is above 100 meters for 4 seconds".

What does this simple sentence mean?
Three Reasonable Interpretations:

1) "Shut off the pumps if the mean water level over the past 4 seconds was above 100 meters".

\[
\left( \int_{T-4}^{T} WL(t) \, dt \right) + 4 > 100
\]

(2) "Shut off the pumps if the median water level over the past 4 seconds was above 100 meters".

\[
\max_{[t-4,t]} (WL(t)) + \min_{[t-4,t]} (WL(t)) + 2 > 100
\]

(3) "Shut off the pumps if the "rms" water level over the past 4 seconds was above 100 meters".

\[
\sqrt{\left( \int_{T-4}^{T} WL^2(t) \, dt \right) + 4} > 100
\]
Fourth Possible Interpretation:

(4) "Shut off pumps if the minimum water level over the past 4 seconds was above 100 meters".

\[ \text{MIN}_{[T-4,T]} [WL(t)] > 100 \]

The most literal interpretation!

A disaster waiting to happen!
The lesson in this anecdote

This is but one of many stories we could tell.

With Complex systems there are thousands of such ambiguities waiting to confuse us.

Ambiguities are dangerous because each reviewer may think he knows what is meant.

All agree that there is only one good meaning— they just disagree on which one.

Formal Methods are the only way known to eliminate these ambiguities.

But, the formal descriptions must be easily read.
What is the Role of Formal Methods?

1. Formal Methods provide professional documentation.

2. Formal Documents provide the basis for black-box testing.

3. Formal Methods provide the structure for systematic inspection.

4. Formal Methods provide a check on white-box testing.

5. Formal Methods are the basis for support tools with semantics.
## Tabular Description of Sample Code

### Table 1

<table>
<thead>
<tr>
<th>( \text{OKTT} = \text{FALSE} )</th>
<th>( \text{OKTT} = \text{TRUE} ) AND NOT InOsenTrip!</th>
<th>( \text{OKTT} = \text{TRUE} ) AND InOsenTrip!</th>
</tr>
</thead>
<tbody>
<tr>
<td>B('PTB',</td>
<td></td>
<td>DOW1</td>
</tr>
<tr>
<td>B('#CND#',</td>
<td></td>
<td>DOW2</td>
</tr>
<tr>
<td>['/EXI']</td>
<td>['/EXI' .OR. 'MASK']</td>
<td>['/EXI' .OR. 'MASK']</td>
</tr>
<tr>
<td>'[H1]'</td>
<td>'[H1]'</td>
<td>'/HTL(5)'/ '+[HYS]'</td>
</tr>
<tr>
<td>'[H2]'</td>
<td>'[H2]'</td>
<td>'/HTL(5)/'</td>
</tr>
<tr>
<td>'[LO1]'</td>
<td>'[LO1]'</td>
<td>'/LT(5)/'</td>
</tr>
<tr>
<td>'[LO2]'</td>
<td>'[LO2]'</td>
<td>'/LT(5)/'</td>
</tr>
<tr>
<td>'</td>
<td></td>
<td>MC</td>
</tr>
<tr>
<td>'</td>
<td></td>
<td>PC</td>
</tr>
<tr>
<td>B(i,</td>
<td></td>
<td>STBV</td>
</tr>
<tr>
<td>j = ['STB'] + i-1,</td>
<td>i in {1,...5}</td>
<td>i in {1,...5}</td>
</tr>
<tr>
<td>B(i,</td>
<td></td>
<td>STBV</td>
</tr>
<tr>
<td>NOT (j in ['STB'] + i-1,</td>
<td>i in {1,...5})</td>
<td>i in {1,...5})</td>
</tr>
<tr>
<td>B('STB' + i-1,</td>
<td></td>
<td>STW</td>
</tr>
<tr>
<td>B(i,</td>
<td></td>
<td>STW</td>
</tr>
<tr>
<td>NOT (j in ['STB'] + i-1,</td>
<td>i in {1,...5})</td>
<td>i in {1,...5})</td>
</tr>
<tr>
<td>B('STB'</td>
<td></td>
<td>STW</td>
</tr>
<tr>
<td>B('TIBL',</td>
<td></td>
<td>TW</td>
</tr>
<tr>
<td>['/HIF(1...5)']</td>
<td>['/HIF(1...5)']</td>
<td>['/HIF(1...5)']</td>
</tr>
<tr>
<td>'[;]'</td>
<td>'[;]'</td>
<td>6</td>
</tr>
<tr>
<td>'[LOF(1...5)']</td>
<td>'[LOF(1...5)']</td>
<td>'[LOF(1...5)']</td>
</tr>
</tbody>
</table>

---

Communications Research Laboratory  
Software Engineering Research Group  
"connecting theory with practice"  

CSA_slides  
March 3, 1996
### Table 2

<table>
<thead>
<tr>
<th>MHiHys(i)</th>
<th>iHIF(i)</th>
<th>iLoF(i)</th>
<th>iINorm(i)</th>
<th>iINHiHys(i)</th>
<th>iINLoHys(i)</th>
<th>iBWLohys(i)</th>
</tr>
</thead>
<tbody>
<tr>
<td>.FALSE</td>
<td>.TRUE</td>
<td>.TRUE</td>
<td>.TRUE</td>
<td>.TRUE</td>
<td>.TRUE</td>
<td>.TRUE</td>
</tr>
<tr>
<td>.TRUE</td>
<td>.TRUE</td>
<td>.TRUE</td>
<td>.TRUE</td>
<td>.TRUE</td>
<td>.FALSE</td>
<td></td>
</tr>
</tbody>
</table>

### Table 3

<table>
<thead>
<tr>
<th>NOT iSensTrip(i)</th>
<th>iSensTrip(i)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| B[i,STB[i]]     | B[i,(iSTW[i].OR.'#TMASK(iSTW[i].AND.UM)]) | B[i,(iSTW[i].AND.'#FMASK(iSTW[i].AND.UM)]) |

### Table 4

**Modes:**

```
*A* = [ ("MC") 1"DEL" OR ("MC" < 0) OR ("PC" + 1) 1"PCL" OR ("PC" + 1 < 0) ]
```

<table>
<thead>
<tr>
<th><em>A</em></th>
<th>NOT <em>A</em></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PC[i]</th>
<th>PCL</th>
<th>PC[i] + 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>MC[i]</td>
<td>DEL</td>
<td>MC[i]</td>
</tr>
</tbody>
</table>

| B[PTB[i],DOW[i]] | B[PTB[i],DOW[i].AND.'#FMASK(PTB[i])] | B[PTB[i],DOW[i]] |
|                 |                                   |                 |
| B[CN#,DOW[i]]  | B[CN#,DOW[i].AND.'#FMASK(CN#)] | B[CN#,DOW[i]] |
|                 |                                   |                 |
| B[CND#,DOW[i]] | B[CND#,DOW[i].AND.'#FMASK(CN#)] | B[CND#,DOW[i]] |
The Inspection Process at Darlington

Four teams:

(1) Application Experts
(2) Programming Experts
(3) Verifiers
(4) Auditors

Roles of The teams:

(1) Produces Requirements Tables
(2) Produce Program Function Tables
(3) Show (1) = (2)
(4) Audit the “proofs”
Who was on these teams?

They were not mathematicians.

Many were engineers with a non-computer background.

Some were graduates of the University of Toronto program in system engineering.

We had our token Ph.D computer scientist who managed to keep up.

Some had only written machine code and could not write boolean expressions at the start.

All required extensive training and supervision at first.

Both AECL and Ontario Hydro now advertise their capabilities proudly.
What was the work like?

We did not work like mathematicians, alone in a room.

There were open presentations of tables and the corresponding code, design reviews, walk-throughs, ....

It took many levels of review because we had no tools to check even the simplest things.

It was dull hard work, not exciting in its detail.

It did not seem unusual to the engineers involved in these critical systems.
Advances after Darlington

(Work at McMaster University, University of Quebec a’ Hull, Warsaw University)

(1) We can explain our methods far better.
(2) We have defined the meaning of a broad variety of table formats.
(3) We have defined the meaning of the logic used in our tables.
(4) We have the kernel of a Table Tool Set.
(5) We have simple printing tools.
(6) We have a test oracle generator.
(7) We have module interface simulators.
(8) We have a statistical black-box module tester.
(9) We have a prototype Display Management System.
(10) We can teach you to inspect your code. We found problems in every industrial program inspected in our course!
MEASURES OF SOFTWARE QUALITY

We must assume the existence of a specification

- The ability to tell “right” from “wrong”

**Correctness:**
Does the software always meet the specification?

**Reliability:**
The probability of correct behaviour

**Trustworthiness:**
Low probability that catastrophic flaws remain after all verification.

Each of these methods is different, each requires a different method.

**Formal methods should not be expected to stand alone**
WHEN SHOULD WE USE EACH OF THESE QUALITY MEASURES?

**Correctness:**

Rarely need it!

Nice to reach for, hard to get.

To a perfectionist, all things are equally important.

Not our real concern, we accept imperfections.

*Use formal methods and rigorous proof.*
WHEN SHOULD WE USE EACH OF THESE QUALITY MEASURES?

**Reliability:**

when all errors are equally important,
when there are no unacceptable failures,
when operating conditions are predictable,
when we can talk about the expected cost,
when your concern is inconvenience,
when we want to compare risks.

Use Testing, Statistical and Planned.
WHAT ARE THE LIMITS OF SOFTWARE TESTING?

Testing can show the presence of bugs but never their absence.

(E.W. Dijkstra)

It is impractical to use testing to demonstrate trustworthiness.

One can use testing to assess reliability.

Two sides of a coin:
I would not trust an untested program!

At Darlington we found serious errors in programs that had been tested for years!
WHEN SHOULD WE USE EACH OF THESE QUALITY MEASURES?

**Trustworthiness:**
when you can identify unacceptable failures,
when trust is vital to meeting the requirements,
when there may be antagonistic "users",
We often accept the systems that are unreliable.
We do not use systems that we dare not trust.
Testing does not work for trustworthiness.
Use formal documentation and systematic inspections.
WHAT ARE THE LIMITS OF SOFTWARE TESTING?

1. It is not usually practical to prove correctness by testing.

2. Testing cannot predict availability.

3. Reliability predictions based on old versions are not valid.

4. Testers make the same assumptions as the programmers.

5. Planned testing is a source of anecdotes, not data (H.D. Mills).


   Formal Methods complement testing.
Even in "Black Box" testing, what's inside does make a difference!

The number of tests needed to identify a finite state machine depends on an upperbound for the number of states.

The length of a test-trajectory will depend on the memory characteristics of the system.
WHAT DOES IT MEAN TO TALK ABOUT SOFTWARE RELIABILITY?

Software is not a random process.

It is the input data that introduce randomness.

"Software Reliability" is a measure of the input distribution through a boolean filter.

Software cannot be assessed as a set of components.

Software + Hardware must be assessed as a single component.

Formal Methods contribute directly to trustworthiness and correctness but less directly to reliability because of the importance of the input distribution.
MEANINGLESS MEASURES USED FOR SOFTWARE RELIABILITY

1) The number of errors per line.

2) Time derivatives of the number of errors per line.

It is the failure rate that matters.

Formal methods do help the first two!
How much testing is needed to assess reliability?

(1) Assume that we have the right input distribution (difficult). We will use tests selected randomly from this distribution.

(2) Let \(1/h\) be the required reliability.

(3) What is the probability of passing \(N\) properly selected tests if each test would fail with probability \(1/h\)?

\[ M = (1 - 1/h)^N \]

(4) \(M\) is the probability that a marginal product would pass a test sequence of length \(N\).

(5) Some examples for \(h = 1000\)

- \(N = 500, M = 0.606\)
- \(N = 1000, M = 0.36700\)
- \(N = 5000, M = 0.00672\)

(6) Some examples for \(h = 1000000\)

- \(N = 1000000, M = 0.36788\)
- \(N = 5000000, M = 0.00674\)

Critical Systems, systems with high reliability requirements, are much harder to test.
Real-time systems are harder to test than batch (memory free) systems

In real-time systems, a test is a trajectory, not an input state.

The trajectory must be long enough that sleeper bugs are revealed.

There must be an upper limit to the memory of the system.

Systems must be structured with testing in mind.

Most of the memory must be periodically reinitialised.

Testing must be repeated for each mode of the remaining memory.

Defining the probability distribution of trajectories is the hardest part.

Formal methods are more important in real-time applications
Conclusions

(1) We can specify software/system requirements precisely and accurately.
(2) We can control complexity by Decomposition, Precise Interface Documentation, Tabular Notations.
(3) Safety-Critical systems require both inspection and testing.
(4) Formal methods have been and can be useful in real-life situations.
(5) They are useful for documentation and inspection even if you are not ready for proof.
(6) Engineering training is essential.
(7) The cost is high but could be reduced by sound tools, tools that "understand" functional semantics and logical notation.

Let's worry about first-things first.

If we cannot use these methods for documentation and inspection, they won't be good for the more advanced applications.

You wouldn't hire an EE who refused to learn calculus would you?

Why do you hire software engineers who do not know classical logic?
The Critical-Software TRIPOD

1) Precise, Organised, Mathematical Documentation and Extensive Systematic Review.

2) Extensive Testing
   • Systematic Testing - quick discovery of gross errors
   • Random Testing - discovery of shared oversights and reliability assessment

3) Qualified People and Approved Processes.

The Three Legs are complementary.

The Third Leg is the shortest.

It's the shortest leg that we should worry about.

We need engineers who know computing!

We need computer experts who understand engineering!

Software Engineering is an unconsummated marriage.
REVIEWABILITY GUIDELINES FOR COMPUTER-BASED SAFETY SYSTEMS

Claude ESMENJAUD
SCHNEIDER ELECTRIC - SES
F-38050 GRENOBLE CEDEX 9
Tel : +(33).76.60.58.60 Fax : +(33).76.60.57.87

ABSTRACT

Computer-based systems important to safety shall meet stringent functional and non functional requirements. When complex functions are performed, commissioning tests based on a black-box approach are usually not considered to be sufficient to gain confidence in the future reliable operation of the system.

One possible way to get additional input on the system is to have it reviewed independently.

This paper gives some guidance, from a manufacturer point of view, on what should be reviewable and what does not need to be, in a computer-based system, important to safety.

Software aspects are particularly addressed, including design method, implementation and tools. Features that may help the independent review process are suggested.
1. INTRODUCTION

The Safety Electronics and Systems (SES) department of SCHNEIDER designs and manufactures I&C systems and equipment for nuclear power plants. The first designs for computer-based safety systems started in 1978 with the French P4 1300 MW plant series, including the 1E classified P4 protection system (P4 SPIN). Nowadays, 20 nuclear power plants are running these systems and have accumulated about 150 reactors x years of successful operation on site.

Nevertheless we know that, for safety systems like the SPIN, successful operation is not a sufficient means for demonstrating safety integrity: the system shall be demonstrated to be safe before starting operation. Furthermore, experience gained before operation through commissioning tests is far from being enough to bring the requested confidence in the system.

In order to increase confidence, it is necessary to design for safety and be able to demonstrate it through the product characteristics and processes used to develop it.

Reviewability is probably one important characteristic the system shall gain during the development process until delivery.

This paper shows why reviewability has been, since the very beginning a basic need in SCHNEIDER - SES for the development of safety systems and why it is still important. It gives our understanding of the term "reviewability" and propose a few guidelines, coming from the recent N4 protection system (N4 SPIN) licensing, on some desirable features that a safety system and his development process should have according to reviewability.

2. REVIEWABILITY - DEFINITIONS AND CHARACTERISTICS

2.1. Reviews through definitions

A review is defined by IEEE Std 610.12-1990 as "A process or meeting during which a work product, or set of works products, is presented to project personnel, managers, users, customers, or other interested parties for comment or approval".

It has also the more specific meaning of searching defects in a product as in the "design review" definition of ANSI/ASQC A3-1978 : "the formal review of an existing or proposed design for purpose of detection and remedy of design deficiencies that could affect fitness-for-use and environmental aspects of the product, process or service, and/or identification of potential improvements of performance, safety and economic aspects" or more recently in the definition of "peer reviews" used in the CMU/SEI-93-TR25 : "the purpose of peer reviews is to remove defects from software
work products early and efficiently. An important corollary effect is to develop a better understanding of the software work products and of defects that might be prevented".

The IAEA "State of the art report on software important to safety in nuclear power plants - draft 2 1993" makes the following comment: "reviews can be divided into several sub-techniques such as walkthroughs, inspections and audits. The common feature is that human interactions play a dominant role. In contrast to more formal techniques such as static analysis or testing, it is human, intuitive understanding that is important in reviews".

2.2. Reviews through SES experience

Reviews are used by SES as part of the development process either internally or with participation of external people.

The main interest of the review process is that it is usable for any kind of document or product at any stage of the development, including plans and requirements. Most of the time, it is the only practical means of gaining acceptance of a document during internal development process.

The important point is that review findings are shared and agreed by several persons other than the author and give the opportunity to correct or improve products or processes. These findings are documented and may be reviewed themselves if necessary.

Reviews aimed to find defects have been systematically performed at SES since the beginning of software development for safety systems and have proved to be very effective in achieving quite fault-free software.

2.3. Definition for reviewability

There seems to be currently no standardised definitions for reviewability. It does not appear in ISO 9126 among the quality characteristics or subcharacteristics for software products: ISO 9126 defines six quality characteristics (functionality, reliability, usability, efficiency, maintainability, portability) and 21 associated subcharacteristics.

"Reviewability" in this paper will mean the ability of a product or document to be reviewed according to the above meanings of "review".

We will discuss "reviewability" mainly from the safety assurance point of view.
3. GENERAL NEED FOR REVIEWABILITY

3.1. Customer needs

The customer is here the people in charge of getting the I&C systems according to plant requirements. They usually established the I&C requirements.

The customer need reviewability of the I&C requirement document. It is essential for a good common understanding with the safety I&C manufacturer.

Then in order to be sure that requirements are correctly taken into account in the design by the manufacturer, the customer often requires to be involved in preliminary design reviews and in the validation process.

This enforces the need of good reviewability for the associated documents.

3.2. User needs

The user is mainly the plant operator or part of the maintenance personnel.

The user may need reviewability of users documentation and system specification.

3.3. Licenser needs

The licenser may need reviewability in order to be convinced that the system to licence is safe.

It appears that acceptance of the system by the customer according to the contractual requirements may be not sufficient to get licensed:

- Validation and commissioning tests, either systematic or random, are not recognised as a valid demonstration of safety requirements, according to the large number of possible states and paths that actual systems may take.

- Assurance gained through the use of tool-based V&V analysis, formal specification and design methods, automatic code generation, use of existing components is partly lost because of the novelty and the increasing complexity of the languages, methods and tools needed to get it.

The licenser tends to require additional safety evidences like additional tests and analysis. He also asks for additional reviews performed by independent personnel in order to get an in depth understanding of the product according to safety concerns.
and to verify the correct application of the plans and standards used for the development.

3.4. Safety I&C manufacturer needs

Reviewability is in fact a basic need for the manufacturer to monitor system and software development.

It consists in establishing formal documents at the beginning and all along the design phases to make the development process visible and kept under control.

The first documents established are development and quality plans, system and software specification documents, verification and validation (V&V) plans. Then come design, code and tests documents, ...

Review of documents allows for continuous tracking of development progress. It gives the opportunity to find defects early and to anticipate possible problems.

Reviews aimed to find defects are used as a basic technology during V&V in order to build quality in the product. Tests and analysis are rather used as a means to check that the expected quality has been achieved.

The need for diverse reviews of the requirements and of the design has been emphasised during the development process by running an independent V&V team. This is strongly recommended by the IEC 880 standard.
4. REVIEWABILITY FOR SAFETY ASSURANCE

In this paper, we will not discuss the benefits of external reviews of safety systems in order to demonstrate safety, compared to other possible means.

We have recognised, from our experience, that reviewability is essential for the development process of safety software in order to meet contracts deadlines, requirements and costs. Furthermore, we use reviews as important means to improve one essential aspect of software safety which is absence of defects.

We will give some reasons why reviewability should be considered after the development process, give some desired properties in order to ease the review process and then give a few guidance on what to review.

4.1. Possible need for reviewability for safety assurance

For the safety I&C manufacturer, the development process starts by signing a contract, generally with the nuclear safety system supplier.

The contract includes basically a set of requirements, a delivery date and an agreed cost. If cost and delivery date are items easy to track, requirements need further investigations.

From a contract point of view, the requirements should have some well known good properties, stated for example in IEEE Std 830-1984. The IEEE standard says that a good software requirements specification (SRS) is unambiguous, complete, verifiable, consistent, modifiable, traceable, usable during operation and maintenance phase.

"Verifiable" is explained by IEEE as : "a requirement is verifiable if and only if there exists some finite cost-effective process with which a person or machine can check that the software product meets the requirement". It means that it should be possible to answer by yes or no to the question "is the requirement met ?" without need of expert judgement (different persons should find the same result).

But, when taking a closer look to requirements, it appears that safety requirements, are generally not "IEEE" verifiable.

Let us consider a simple functional requirement like "Emergency trip of the reactor when Lpress<470 mm & Pcore out <117 bar". Typically, you show that the software is able to give the right order under specified inputs by designing a set of test values for Lpress and Pcore out.

It is usually enough to demonstrate good operation for industrial process control and may be accepted as sufficient for commissioning of the safety system.
It is generally considered as not enough to give assurance of continuous reliable operation as it does not demonstrate that the software will never either generate spurious order or fail to give the right order during operation.

In fact, for industrial process control, this ultimate assurance is gained a posteriori through the first months of operational use.

For safety system, operational use, like testing, is only able to give assurance of presence of fault when a failure occurs. It cannot assure of absence of fault and it would not be wise to start operation with a non fully trusted system.

Reviews performed on the delivered system, aimed to find possible remaining defects and to have external people understand the system behaviour, is one possible way to increase assurance that the software and more generally the design may be trusted.

4.2. Reviewability at delivery time

We consider the situation at product delivery time, from the point of view of getting assurance of future reliable operation.

At that point, the product exists with its documentation and is ready for plant integration and operation. It has undergone manufacturer tests and validation, generally with involvement of the customer. It is supposed to meet the requirements according to the agreed verification clauses stated in the customer-manufacturer contract.

It shall be pointed out that aspects to consider at that time are different of those to consider when establishing the requirements and selecting a contractor from the customer point of view.

Basically, the following aspects should be considered:

- the product itself as delivered with its user and maintenance documentation,
- development by the manufacturer including:
  - design documents,
  - verification and validation (V&V) results,
  - methods, languages and tools used for design and V&V,
  - others (quality assurance (QA), training, manufacturing, subcontracting, support...).,
- requirements (safety, plant, I&C requirements).
For licensing purposes, additional verifications processes, involving others organisations that the manufacturer's may be required.

Typically:

- additional testing

The external organisation needs the requirements, the product with specification and users documents and checks the behaviour of the product according to the requirements by designing and running additional tests (either systematic or statistic).

- reverse engineering (back from the code to the requirements)

The external organisation needs the requirements and the binary code to establish, using static analysis, that the product will perform consistently with the requirements.

- product review

The external organisation gets the requirements and all design documents and source code to redo a manual or possibly tool assisted general review of the design.

The main goal is to understand the design and code and to find possible discrepancies that could impair the future operation and maintenance of the product, from a safety point of view. It is also an opportunity to improve assurance that the product is free from design faults.

- other

it is also possible to add verifications on other aspects, including relevance of V&V plans, methods languages and tools, compliance with standards and with state of the art practices, verification of test results, ...

The main inputs for additional verifications are the product design documents, the code and the requirements.

These additional verifications usually focus on the product because it is the product that will ultimately run during operation. Nevertheless, others factors should be considered as the operating environment, the operator and the requirements for their possible interaction with the product.
5. GUIDELINES FOR REVIEWABILITY OF THE PRODUCT

The product here is the I&C safety system at delivery time.

In this chapter, the system will be viewed as a set of hardware-based and software-based processing units interconnected through data links. It delivers outputs according to inputs and internal states. A typical example is a digital protection system.

In order to review the system design, it is necessary that:

- functional requirements are stated and understood from an I&C point of view,
- the goal of the review is understood, i.e. to understand how requirements are implemented in the system, how the system will actually handle them in operation, and find possible defects,
- design documents are available from the manufacturer.

5.1. Guidelines for reviewability of system & software design

The following guidelines addresses the design documentation as it is a main input for the review. (Another important input not developed in this paper is the requirements).

(1) traceability of functional requirements in the design

The following may help to better understand how functional requirements are implemented in the design:

◊ Separation of application functions and system functions. The requirements could be mainly implemented in application specific functions using services offered by system functions.

◊ Use of application oriented formalisms closed to those used for the requirements. For instance, data flow diagrams or boolean and numerical expressions to translate logical requirements.

◊ Mapping of requirement data on system input/output or internal data.

(2) system behaviour description

The following may help to better understand the behaviour of the system that is to understand how the system actually performs the required functions.

◊ Explanation of system behaviour, using for example functional diagrams or states/transitions diagrams at the appropriate levels of system decomposition.
◊ Give description of the protocols or mechanisms used for building the system with characteristics and possible limitations. For example, sensitivity to traffic load for networks or to computation load for processing units.

◊ Explanation of the underlying execution models used to support formal design descriptions.

(3) unit and link description

The following may help to better understand the system decomposition in parts and the contribution of each part to the performance of the requirements.

◊ Function of each unit or link, according to requirements (either direct implementation of application function or system function). For this purpose, several levels of decomposition of units may be provided.

◊ Unit or link behaviour and performances (processing power, data flow rate, response time,...).

◊ Characteristics of each unit or link (hardware-base or software-based, new or existing, customised use or standard use, part of a building-bloc system or part of a dedicated architecture.

◊ For application function units, either hardware-based or software-based, designation of the hardware and/or software parts actually implementing the application function.

(4) basic components

The following may help to better understand which components are directly under manufacturer control and which are from outside.

◊ Explanation of basic components used to build the system.

they may range:

• For units: from standard application units ready for use at system level to standard signal converters or automatism function boards, either hardware-based or software-based (in that case, software is said to be embedded or hidden).

• For links: from standardised multipurpose networks through dedicated networks either software-based or hardware-based to electrical wires for carrying one logic or analogue signal.

• For hardware: from standard boards and racks through complex integrated circuits with microcode or gate arrays to discrete components.
For software: from configurable application software through sets of libraries, operating system, real time kernel executive to the microprocessor instruction set.

◊ Functions and characteristics of basic components (unit or link, hardware-based or software-based, new or existing, proprietary or commercially available, ...).

(5) implementation

the following may help to better understand the relationships between diagrams representing units, links and components and actual pieces of hardware and software (only the software aspect is considered here).

◊ For each type of software design description: explanation of actual implementation of the software design description: in line or threaded code, data configuration, use of basic components, ... . Several levels of description may be provided depending on the various possible forms of intermediate code used for code generation.

◊ For each unit: explanation of the software configuration of the unit with distinction when appropriate between "system software" (or operating system) which provides interfaces with the hardware and standard services to application software and the application software itself.

The frontier between application software and system software may be different according to the software architecture.

Typical situations are:

• no clear frontier (example of some embedded software),

• set of software modules directly associated to design descriptions (like SAGA diagrams, functional blocs, state charts, ..) running on a tailored system software,

• set of tasks written with low level, high level, or object oriented languages running on a real time executive kernel or operating system,

• set of configuration data customising an application oriented existing software or software-based unit.

5.2. Reviewability priorities for a safety system

This paragraph gives a short outline on what is reviewable on the N4 SPIN system. It defines the lower level of hardware and software decomposition which is available from the SES department.
5.2.1. The N4 SPIN system

The N4 SPIN system is composed of a set of hardware-based and software-based processing units interconnected through NERVIA data networks. It delivers emergency actuation orders according to rod position, process and nuclear instrumentation information.

It is composed of various processing units (Acquisition Units, Functional Units, Voting units, ...) arranged in a four channels redundant architecture with some functional diversity at functional units level.

All application specific design documents are established by SES from the customer's requirements. They are available for licensing purposes.

The hardware is composed of:

- A standardised safety dedicated set of boards including input/output boards, Motorola 68k based CPU boards, NERVIA interfaces, ... . This set is called GRENAT.

- Typical basic hardware components are MP 68000 microprocessors, MP 68881 mathematical coprocessors, Intel 8031 and 8344 microcontrollers, PLA's and standard electronic components.

All hardware design and manufacture is directly controlled by SES. Associated design documents are available for licensing purposes.

The software is composed of:

- System software embedded in I/O boards and NERVIA interfaces.

- Standardised system software components for the MP680x0 microprocessors.

- Specific application software, designed with the SAGA data flow formalism, traceable to a software module making use of a set of libraries components and running cyclically under control of the system software. The execution model is very simple: no interrupts, unending loop doing hardware supervision tests, acquisition from inputs board and NERVIA, processing of application, output towards output boards and NERVIA.

All the software used is directly written or controlled by SES. All design documentation, source code (SAGA diagrams or C code for libraries function and parts of system software, Assembly code for embedded software and all associated binary code) is available for licensing purposes.

5.2.2. Comments on the SPIN N4 reviewability

The execution model for processing units and NERVIA networks has been kept very simple. It allows for deterministic calculation of worst response times of the system and eases system behaviour explanation under functional and dysfunctional situations.
The N4 SPIN uses software-based integrated circuits like the MP 68881 mathematical coprocessor. Review of the internal software of this component has not been considered. Black box tests of the limited set of routines used by the application software have been considered by the French licensing authorities as an acceptable means of validation of this component, taking into account an implicit feedback of experience.

The SES department has now moved to the MP68040 microprocessor associated with the MP68360 controller for communication management. It may use others sophisticated hardware components, provided that they are widely used.

At the moment, the SES department does not consider using existing commercially available software layers in safety system design such as real time kernel executive or standard software communication layer.

5.2.3 Proposition for reviewability priorities

1. First, anything developed specifically to meet the functional requirements as application part of the system (system architecture if application specific, implementation of functional requirements, data configurations).

2. Then the ability to analyse and predict the system behaviour and response time according to given situations. Learning the behaviour of the system from experiments is not recommended. Experiments should be necessary only to confirm expectations.

3. Then, anything new in the system: hardware and software enhancements, new system functions, new libraries components, new generation tools, ....

4. Then, the way reused parts are used in the system (same or different from former uses).

5. Then, if the system is built from highly configurable components, the parts of the system components actually used by the application for possible never tested before situations.

Parts of the system which are continuously exercised like for example the instruction set of the microprocessor, communication protocols between units, acquisition units exercised typically every 10ms through the same algorithms may not need in depth further investigations.

Components or units already in use in other safety systems and which have been subjected to licensing may need far less licensing effort when reused.
6. REVIEWABILITY OF THE DEVELOPMENT PROCESS, TOOLS AND V&V RESULTS

6.1. Development process

We call development process, the way the safety I&C manufacturer organises phases, activities, man-power, reviews, QA, training, to develop and manufacture the product.

This process is generally required to be reviewable; it means that evidences of plans, and of the performance of activities like reviews, quality assurance tasks, training shall be available.

These evidences are not basically an assurance of the safety of the final product.

They should not be needed for external safety reviews (outside contractual context).

They can be used to try to make correlation between practices and results.

6.2. Tools

We call tools, software used for development, V&V, configuration management, manufacturing, documentation, but not used to operate the product.

From a safety point of view the priorities may be:

(1) development and manufacturing tools because they may include faults in the product,

(2) V&V tools because they may fail to detect faults they are supposed to detect,

(3) other tools like software configuration management (SCM) tools for their possible undesired consequences on the product.

Guidelines

• Development and manufacturing tools used for safety development should produce reviewable outputs: transformations made by the tool should be such that it is possible and practical to check if the output is a sound translation of the inputs (it does not mean that it has to be done systematically on every piece of output).

• V&V tools - principle of verification or validation achieved should be understood it should be possible to demonstrate that they are able to find samples of fault they are supposed to find.
As a general principle, even if the risk to include faults is far less when doing a task by a tool than doing it by hand, no stage of development should rely only on tools. The apparently ease and security given by fully automated development processes does not exempt from understanding the actual transformations achieved by a tool and from being able to check if results are corrects.

6.3. V&V results

V&V results include all verification, formal proof, simulation, test results,... done by the manufacturer during development and for commissioning.

V&V results needs to be reviewable as they are the demonstration that the product meet the requirements established by the customer.

It should be possible to understand and verify computations, justifications and proofs, and to replay tests if necessary.

A basic need of reviewability of V&V results may be to find possible complementary verifications needed according to what has been done by the manufacturer.
7. CONCLUSION

For the safety I&C manufacturer, reviewability of his product during the development is a basic need to be able to meet the contract requirements. It helps him make the appropriate choices and find defects as early as possible by using diverse views on the design coming from various internal and external teams (design, V&V, QA, customer representatives, ...).

Review of the product is a means used to increase assurance of fulfilment of the safety requirements during the licensing process.

For the P4 SPIN and the N4 SPIN systems, SCHNEIDER - SES has been able to provide all the application and system software as design documents, source code and binary code, as well as hardware documents and equipment for external licensing purposes.

This level of reviewability is available to the licensing authorities in charge of our customer projects abroad, provided a contractual agreement with confidentiality clauses is accepted.

What is important to review remains the licensor choice. How easy it is to review and what may be improved in order to get a better assurance of safety behaviour is an open workshop involving our customers FRAMATOME and EDF, the French licensing authorities DSIN and the SCHNEIDER - SES department.
The design requirement and development of software for the digital safety protection system in Kashiwazaki-Kariwa units 6 and 7

Takaki Mishima
Tomoaki Shirakawa
The Tokyo Electric Power Co., Inc

ABSTRACT

Kashiwazaki-Kariwa units 6 and 7 (K-6/7) are 1356MWe Advanced Boiling Water Reactor (ABWR), which are the first application of the digital safety protection system in Japan.

Because digital control and network systems have already been applied to the non-safety Instrumentation and Control (I&C) systems in BWRs in Japan, Tokyo Electric Power Company (TEPCO) has expanded digital technology to the safety protection system through the reliability estimation based on the design and operating experiences.

Upon the application of digital technology to the safety systems, it is important to achieve the highly reliability of the total system in its design and manufacturing process.

The software design requirements for the safety protection system in K-6/7 were based on some previous R&D results and a guideline for the application of digital controllers (JEAG4609)

The characteristics of software are such that the programing process is made simple and easily to be understood by using the symbolic language (POL), the software is made simple by applying cyclic execution and non-interruptive execution, and the software size is relatively small to confirm the validity after system integration easily.

The software for the safety protection system was designed, manufactured, and installed according with the so-called V&V procedures with sufficient reliability.
INTRODUCTION

Kashiwazaki-Kariwa units 6 and 7 (K-6/7) are 1356MWe Advanced Boiling Water Reactor (ABWR).

Tokyo Electric Power Company (TEPCO) improved the sophistication of operation, the reliability and maintainability of Instrumentation and Control (I&C) systems by adopting integrated digital control technology in K-6/7 and achieved to expand digital technology to the safety protection system through the reliability estimation based on the design and operating experiences.

This paper reports the TEPCO's way of consideration concerning with the adoption of the digital safety protection system for K-6/7.

DEVELOPMENT HISTORY OF DIGITAL CONTROL SYSTEMS

The I&C systems in the nuclear power plants can be characterized by a huge amount and high space density of information, high reliability requirements and many safety features required by regulation.

In the early 1980s a significant number of troubles in the BWR in Japan were caused by failures of I&C equipment as shown in Fig-2.

The mid-1980s saw a rapid progress of the microprocessor technology in the I&C systems, which brought an incentive to improve the I&C systems using this technology. The first application was for making the main control systems more reliable by triplicating their Central Processing Unit (CPU) and this application was so successful that triplicated main control systems have been retrofitted to many plants up to this time. As a result of such an effort, the number of troubles caused by failures of I&C equipments have been decreasing year by year.

The purposes of application of the digital control and network systems can be summarized as below:

1. higher reliability with reduced number of components, self-diagnosis, and driftless feature,
2. better maintainability with unitwise repair or replacement, standardized components, on-line maintenance, and self-diagnosis,
3. reduced amount of hardware with less cabling by using multiplexing transmission and fewer control panels,
4. easy modification with software, etc.

With these features in mind, we have applied digital technology, step by step, to single loop controls (early 80’s and later), radio-active waste treatment plant systems (early 80’s and later), dedicated main controls as FDWC, RFC, EHC (mid-80’s and later), and non-safety systems (early 90’s and later). Finally, we adopted the digital safety protection systems for K-6/7 through the
reliability estimation based on such design and operating experiences.
Fig-1 shows the application of digital controllers in BWRs.

DESIGN CONSIDERATIONS OF DIGITAL CONTROL SYSTEMS

In order to successfully apply digital control and network systems to the nuclear power plants, consideration must be directed to not only their performance, but also their failure modes and effects. This is particularly true for safety-related systems. The following shows some of the considerations for K-6/7 design.

(1) Scope of digital control
   Even in the fully computerized system like K-6/7, hardwired controls are still used for some parts of the system as follows.
   ① A function is closed in a local area,
   ② Very little signal transmission to the main control room,
   ③ TSI, Turbine trip signals, and Trip sequence logic which require short response time,
   ④ SLC and RSS which are respectively used for backing up the RPS and the ECCS operation from the main control room,
   ⑤ Another backup system which are provided aiming at the defense in depth consideration and similar to the U.S. ABWR design.
     - Display system
       - RPV water level
       - RPV pressure
       - D/W pressure, etc.
     - Control system
       - Manual scram
       - Manual MSIV control
       - Manual HPCF(C) initiation, etc.
     - ATWS(Anticipated Transient Without Scram)
       - RPT
       - ARI

(2) CPU sharing
   In the K-6/7 design, CPUs were shared in the same way as the conventional analog systems and divided into groups considering annual inspections and maintenance as well as risks against CPU failures.

(3) Redundancy
   In K-6/7 design almost all of the control systems were duplicated.
Exceptionally, dedicated main control systems were triplicated and some parts of the system are single which do not directly affect plant outputs such as radiation monitoring system and TIP system.

(4) Assignment of signals and functions

Assignment of signals and functions to each multiplexing unit and Input/Output board was carefully made through the design review in the case of single or double failure modes.

K-6/7 I&C SYSTEM CONFIGURATION

As an example of fully computerized I&C systems, a schematic drawing of the digital control system for K-6/7 is shown Fig-3 and a detail is shown in Fig-4. The system consists of plant computer system (process computer), distributed microprocessor-based digital controllers for reactor systems, and those for turbine systems. The process computers serve as a CRT driver, trend recorder, overall automation manager, core performance calculator, etc. Most of the plant controls and logic are performed in each of the distributed control systems.

SYSTEM CONFIGURATION OF SSLC

The digital control system for safety protection system is called Safety System Logic and Control (SSLC). Configurations for the Reactor Protection System (RPS) and the Emergency Core Cooling System (ECCS) are somewhat different because RPS is of "fail safe" design while ECCS is of "fail as is" design. SSLC configurations are shown in Fig-5 and Fig-6. 2 out of 4 logic is used for RPS and ECCS. SSLC for RPS consists of four divisions of Digital Trip Module (DTM) and Trip Logic Unit (TLU). DTM s issue trip signals and send them to TLUs. TLUs generate trigger signals by actuating circuits. SSLC for ECCS consists of four divisions of DTMs and three divisions of Safety Logic Units (SLU). Each division of SLU is duplicated. SLUs generate start signals by 2 out of 4 logic and send them (via multiplexing units) to actuating circuits.

SYSTEM DESIGN REQUIREMENTS FOR SSLC

In adopting 2 out of 4 logic for SSLC, although it was not a requirement or target by safety regulation, we confirmed through the calculation of reliability based on FTA that the reliability of this logic was the same as or better than that of conventional logic \([(1 \text{ out of } 2) \times 2\] ). Further, we
comparison of the digital system with the conventional analog system. Table-1 shows a risk comparison of the digital system with the conventional system.

Considering these factors, we gave below system design requirements to the SSLC.

1. higher reliability compared with conventional system,
2. high speed response time which was referred to the Establishment Permit,
3. locate a fault and enable the on-line maintenance,
4. electric and physical separation between safety divisions,
5. minimize half-scrams and scrams and also minimize inadvertent ECCS operation,
6. ensure clarity and traceability of design and manufacture of software.

RESEARCH & DEVELOPMENT

Digital technology is widely used in the nuclear power plants and many experiences have been obtained on the design, manufacturing, testing, operation and maintenance. However, because K-6/7 were the first application of digital safety protection system in Japan, the following R&D were previously performed.

1. Software

A guideline for the application of digital controllers to the safety protection system was established after the R&D program conducted by the Japan Electric Association, including the simulating test using the digital controllers and investigation of European (IEC-880) and American (ANSI/IEEE-7.4.3.2) standard. In order to standardize Verification & Validation (V&V) method, this guideline was published as "Application Criteria for Programmable Digital Control System in Safety-related System of Nuclear Power Plants" (JEAG-4609) in which basically the same method was used as in American standard (ANSI/IEEE-7.4.3.2).

V&V methods are specified in this guideline to achieve the reliability and validity of software and to be easily understood by third person.

Fig-7 shows the correspondence between the design and manufacturing flow and V&V flow as defined in JEAG4609.

2. Hardware

The validation tests of the computer system, loading the prototype software for RPS were performed by NUPEC (Nuclear Power Engineering Corporation), including implementation of the V&V. The tests were event-simulation such as the noise immunity test, thermal aging test, vibration test, heat-up test, etc. This series of tests confirmed that the computer system used
for the RPS possessed the required ability. Fig-8 shows the proving test flow.

SOFTWARE DESIGN REQUIREMENTS FOR SSLC

The software used for digital safety protection system was mainly classified into operating system software which was originally installed in controller and application software.

Operating system software was given no special requirement because it would be completed to debug as it has been successfully done in non-safety systems and other industry system.

On the other hand, the below requirements listed below were given to the application software to be easily understood by a third person.

① symbolic language (POL),
② simple logic,
③ relative small size,
④ module by each function,
⑤ cyclic execution,
⑥ no interruption,

VERIFICATION & VALIDATION OF SSLC

(1) V&V along with JEAG

The V&V work for SSLC was performed in accordance with the Construction Plan and at each stage from design phase to pre-operation phase.

In case of hardware, the V&V was performed in accordance with conventional QA/QC procedure (JEAG4601).

In the case of software, the V&V work was performed along with JEAG 4609. We had confirmed through reports submitted from manufacture at each stage that V&V had been carried out in an appropriate manner and had properly controlled those documents as QA/QC records.

(2) V&V along with TEPCO’s QA activities

We adopted two types of SSLC consequently due to different manufacturer between K-6 and 7 although the design was based on same design specification.

In view that this was the first application for safety protection system, we also had crosschecked between K-6 and 7’s V&V documents and semi-dynamic simulation test data performed voluntarily by the manufacturer. Finally, we confirmed that both systems also met the design requirement.
① Crosscheck between K-6 and 7’s V&V documents

We mainly compared and coordinated Interlock Block Diagram (IBD) between K-6 and 7.

Fig-9 shows the software design and implementation flow.

The original concept of software which has almost one to one correspondence and correlation with IBD are inputted in the CAD system using keyboard, etc. The software is automatically produced as output of the CAD system and contained in the floppy disk. The software contained in the floppy disk is loaded by using loading function of the maintenance tool connected to the digital controller.

This flow also shows that the software has correspondence and correlation with IBD and there is less possibility of human-error during software manufacture.

Furthermore, the maintenance tool was developed by Hitachi and Toshiba individually and has been used in many plants with satisfactory results.

Therefore, we are confident that to compare and coordinate between K-6 and 7’s IBD identified with to confirm the propriety of software directly.

② Semi-Dynamic Simulation Test

After the manufacturer confirmed that the integrated hardware and software met the system design requirement, the Semi-Dynamic Simulation Test was conducted as the voluntary quality activity. The manufacturer varied the process conditions in conformity with the safety analysis conditions referred to the establishment permit by using the simulator and analyzed time record of output-signal during this test.

We also compared the results of this test between K-6 and 7. Finally, we reconfirmed through the series of V&V and crosschecking of V&V documents that the software for SSLC was sound.

We confident that these approaches are effective to prove the soundness of software to a third person

CONCLUSION

Digital control systems have been applied step by step to single loop controls, dedicated controls, radio-active waste treatment plants, and non-safety systems since 1980’s. In this stage, many confirming experiences have been obtained on the design, manufacturing, testing, operation and maintenance.

Furthermore, the validation tests of the digital safety protection system, loading the prototype software for the RPS were performed by NUPEC (Nuclear Power Engineering Corporation), including implementation
of the V&V. This series of tests confirmed that the digital safety protection system used for the RPS possessed the required ability.

Therefore, we can say that the digital safety protection system for K-6/7 (SSLC) was completed by integration of existing technologies and extensive experience. We are confident that the digital safety protection system for K-6/7 can satisfactorily stand comparison with foreign systems and standards because of the V&V performed in accordance with JEAG4609 and the hard-wired back-up system for the digital safety protection system.

References


2. Takao Tochigi, Yusuke Kajikawa, Chikara Takayama, "DEVELOPMENT OF INTEGRATED DIGITAL INSTRUMENTATION AND CONTROL SYSTEMS", In proceedings of the 1992 OECD/IAEA international symposium, Tokyo, 18-22 May, 1992

3. Hiromu Kikuchi, "Digital control technologies applied to TEPCO nuclear power plants", In proceedings of the 1995 INE international conference on C&I in nuclear installations, UK, 19-21 April, 1995
### TABLE-1 Risk Comparison of the Digital System with the Conventional System

<table>
<thead>
<tr>
<th>ITEM</th>
<th>CONVENTIONAL SYSTEM (ANALOG SYSTEM)</th>
<th>DIGITAL SYSTEM</th>
<th>EVALUATION (*1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Hardware</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(1) Random Failure of Components</td>
<td>• Loss of function in one channel</td>
<td>• Same as left</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Large number of components</td>
<td>• Less number</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Some causes of drift</td>
<td>• Few causes</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Some causes of aging</td>
<td>• Few causes</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Can be detected by surveillance test (Usually once a month)</td>
<td>• Same as left (*2)</td>
<td></td>
</tr>
<tr>
<td>(2) Common mode Failures</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>a) External event</td>
<td>• System function shall be lost if fire occurs in Central Control Room but reactor will be shutdown automatically and subsequent core cooling operation is performed from remote Shutdown System</td>
<td>• Same as left</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Damages remains in one channel in case of fire in Local Room</td>
<td>• Same as left</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• seismic class : As</td>
<td>• Same as left</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• not relevance</td>
<td>• Not possible because of isolated computer system</td>
<td></td>
</tr>
<tr>
<td>b) Internal event (*5)</td>
<td>• Verified on the postulated environmental conditions</td>
<td>• Same as left (*3)</td>
<td></td>
</tr>
</tbody>
</table>

| 2. Software |  |  |  |
| (1) Errors in basic design phase | • Loss of system function, but errors can be detected in design review or pre-operational test | • Same as left |  |
| (2) Errors in detailed design phase | • Loss of system function (same drawing is used for each channel), but errors can be detected in design review or pre-operational test | • Same as left |  |
| (3) Errors in manufacturing phase |  |  |  |
| a) errors in wiring | • Loss of function in one channel | • Same as left |  |
|  | • Comparatively large number of wiring | • Less number |  |
|  | • No software | • Loss of system function |  |
| b) programming errors |  |  |  |

(*1) ○: Digital is better or Comparable, △: Conventional is better.

(*2) Self-diagnosis capability of Digital system makes failure detection and repair early

(*3) Errors detection is enhanced by V&V in Digital system

(*4) Programming errors are not likely to occur because of the high level language and simple logic processing and small size of program. Besides, errors can be eliminated in design review and/or pre-operational test through V&V.

(*5) Rapid change on physical characteristics by environmental change.
Fig. 1  APPLICATION OF DIGITAL SYSTEMS  
IN JAPANESE BWR PLANTS

<table>
<thead>
<tr>
<th>Application</th>
<th>Year</th>
<th>'80</th>
<th>'85</th>
<th>'90</th>
<th>'95</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1) Dedicated System Control</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>In Operation: 26 units</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Under Construction: 2 units</td>
</tr>
<tr>
<td>• Local Loop Control</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Major Control Systems</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>In Operation: 21 units</td>
</tr>
<tr>
<td>(Feedwater Control,</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Under Construction: 2 units</td>
</tr>
<tr>
<td>Turbine Control, etc.)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(2) Integrated Systems</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>In Operation: 11 units</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Under Construction: 2 units</td>
</tr>
<tr>
<td>• Radio-active Waste</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>In Operation: 4 units</td>
</tr>
<tr>
<td>Processing (RW) System</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Under Construction: 2 units</td>
</tr>
<tr>
<td>• Non-safety System</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Under Construction: 2 units</td>
</tr>
<tr>
<td>• Safety-related System</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Fig. 2  Troubles Occurrence of Japanese Nuclear Power Plant (except RW Plant)
Digital Control Systems for the ABWR

Fig. 3
Division-1

Division-2

Division-3

Division-4

(Detectors)

Note: RHU: Remote Multiplexing Unit
DTH: Digital Trip Module
SLU: Safety Logic Unit
RSS: Remote Shutdown System
ECCS: Emergency Core Cooling System

Fig.6 SSLC Configuration for ECCS
System Requirements for Digital Safety Protection

- System Design Requirement Specifications
  - Hardware/Software Design Requirement Specifications
    - Hardware Design and Manufacture
      - Software Design
        - Software Manufacture
          - Hardware/Software Integration
            - Validation
              - Final System

- Basic Plan for Verification and Validation
  - Verification 1
    - Verification 2
      - Verification 3
        - Verification 4
          - Verification 5

Note 1: Design and Manufacturing
Note 2: Verification and Validation

Fig. 7 Verification and Validation Flow
Fig.8 PROVING TEST FLOW FOR THE SOFTWARE LOGIC PROTECTION SYSTEM

INITIAL CHARACTERISTICS TEST
- SAFETY SYSTEM FUNCTION TEST
- DIAGNOSING FUNCTION TEST
- ANTI-NOISE TEST
  - POWER NOISE TEST
  - INDUCED NOISE TEST
  - STATIC ELECTRICITY NOISE TEST
  - TRANSCEIVER EMISSION TEST
- AMBIENT TEMPERATURE TEST

THERMAL AGING TEST
- SAFETY SYSTEM FUNCTION TEST
- DIAGNOSING FUNCTION TEST
- ANTI-NOISE TEST
  - POWER NOISE TEST
  - INDUCED NOISE TEST
  - STATIC ELECTRICITY NOISE TEST
  - TRANSCEIVER EMISSION TEST
- AMBIENT TEMPERATURE TEST

SEISMIC AGING TEST
- SAFETY SYSTEM FUNCTION TEST
- DIAGNOSING FUNCTION TEST
- ANTI-NOISE TEST
  - POWER NOISE TEST
  - INDUCED NOISE TEST
  - STATIC ELECTRICITY NOISE TEST
  - TRANSCEIVER EMISSION TEST
- AMBIENT TEMPERATURE TEST

ACCIDENT, SIMULATED TEST
- AUTOMATIC OPERATION TEST

FINAL CHARACTERISTIC TEST
- SAFETY SYSTEM FUNCTION TEST
- DIAGNOSING FUNCTION TEST
- ANTI-NOISE TEST
  - POWER NOISE TEST
  - INDUCED NOISE TEST
  - STATIC ELECTRICITY NOISE TEST
  - TRANSCEIVER EMISSION TEST
- AMBIENT TEMPERATURE TEST

EVALUATION
Software Design and Implementation Flow

Block Diagram
- Sw Condition
- Reset Signal
- Valve
- Open Condition
- Test Signal
- Operation Signal

Software design
- Logic Symbol connection planning with a Symbolic Language
- According to the Block Diagram

Software Manufacture
- Connecting Logic Symbols Using a Display and a Keyboard etc.

Logic Diagram
- Sw Condition
- Reset Signal
- Valve
- Open Condition
- Test Signal
- Operation Signal

Confirmation

CAD System

e.g. FD
OECD/NEA International Workshop

on

Technical Support for Licensing of Computer-Based Systems Important to Safety

SAFETY CASES - PRODUCING A REVIEWABLE SYSTEM

N Wainwright
Principal Inspector
Health And Safety Executive
Nuclear Safety Division
Bootle, Merseyside, U K

INTRODUCTION

The UK's nuclear industry is regulated through the Nuclear Installations Act which empowers, through delegation from the Health and Safety Executive, the Nuclear Installations Inspectorate to grant nuclear site licences and set licence conditions which must be observed by the licensee. Of these license conditions, particular ones require the licensee to produce an adequate safety case for the installation. In addition, in some of the UK's other industries, eg. oil and railways, the content of a safety case is now regulated through legislation (refs 1,2); and for other UK industries risk assessment is required to a degree appropriate for the hazard.

The UK Health and Safety Executive's definition of a safety case\(^1\) is (ref 3):

\[
\text{a systematic and where possible quantified demonstration that an installation or system meets specific safety criteria,}
\]

and since for the purposes of this paper I propose to define a reviewable system as:

\[
\text{a system that is understandable by its reviewer, and where the evidence and reasoning offered by the licensee enables a conclusion to be drawn on whether or not the system is fit for purpose,}
\]

I, therefore, contend that a safety case and the documentation of a reviewable system are one and the same thing.

Hence I feel that the requirement for the production of a safety case encourages the generation of reviewable systems. And it should mean that the likelihood of a reviewer being found sitting in front of a computer-based system-important-to-safety, frantically scanning a set of drawings and a software listing, trying to decide whether or not the system is fit for purpose will be ruled out forever.

\(^1\) In this paper the term "safety case" is used in relation to a specific system - this is only part of the "site safety case" which would address wider issues.
THE SAFETY CASES OF SYSTEMS IMPORTANT TO SAFETY

Firstly, I shall discuss the general requirements of safety cases\(^2\) for the various systems important to safety in nuclear plant (here I am referring to safety systems and safety-related systems as defined in the IAEA regulatory guides (ref 4)). This will enable me to determine the degree of reviewability required to demonstrate the fitness for purpose of these systems. As with all systems important to safety, the evidence and reasoning required in the safety case of a plant incorporating a computer-based system depend on the significance of errors or failures in that system to the overall plant risk; the greater the significance, the more robust (in terms of the amount of evidence and rigour of the reasoning) must be the safety case demonstration.

The system having the highest level of importance in a nuclear facility is the safety system since it is designed to detect potentially dangerous fault sequences\(^3\), and implement appropriate safety actions, ie. to terminate a fault sequence or mitigate its consequence. As a result, the safety cases of safety systems need to withstand a most critical examination to establish their fitness for purpose. This level of examination, of course, places a greater requirement for reviewability on the system. The review will be seeking evidence of the appropriate application of the sound principles of redundancy, segregation, diversity and fail-safe design as required by Principles P78 to P81 of the UK Nuclear Installations Inspectorate's Safety Assessment Principles (SAPs) (ref 5). The redundancy requirements need to satisfy the single failure criterion; reduce the potential for spurious actuations; and meet the maintenance needs. For high reliability systems, common cause failures limit the reliability that can be claimed for redundant systems, therefore, the inclusion of a diverse system may have to be considered.

Because control and surveillance systems are not the principle means of ensuring nuclear safety, their safety cases require lesser demonstrations. A safety demonstration commensurate with their contribution to the overall risk needs to be developed. NII's SAPs typically require that the design, construction and inspection of control and surveillance systems should be reviewable against the appropriate national or international standards (P208). Additionally, functionality, reliability, accuracy, stability and responsiveness should be demonstrably adequate for their intended duty (P209). A documented analysis of the foreseeable ways in which internal control system failures (including multiple failures) could place demands on the safety systems is required as part of the safety demonstration (P207). The safety case should then show by analysis that the safety system is designed to protect against these failures. Furthermore, control system failures should not be excessively frequent (P207); and, control systems' stability and response times should be such that no demands, under normal conditions, are placed on the safety systems in the presence of normal plant disturbances (P206). Both these latter two properties should be fully justified in the safety case.

With regard to operator and maintenance aids such as computer-based advisory systems and the support systems for systems-important-to-safety, again there needs to be a determination of their contribution to plant risk plus a commensurate level of safety demonstration. This will need to be undertaken on a case-by-case basis noting the points made above. However, procedures should be in place on the plant such that the advice given by these systems is checked before action is taken

---

\(^2\) Assuming from now on that the documentation needs of a safety case represents the needs of a reviewable system.

\(^3\) Here a fault sequence is defined as a combination of a events starting from postulated initiating event and including any additional failures that may occur.
since incorrect operator action may defeat correct protective action, or may result in the placing of unusual or more frequent demands on the safety system.

So, in considering software-based systems, the need for reviewability increases as the degree of criticality for safety increases. It may be acceptable to use a proprietary "black box" system for less-critical uses (although its pedigree needs to be established), but the reviewer will need to go much more deeply into the more critical systems.

SAFETY CASES AND REVIEWABILITY

Having established the degree of reviewability required by systems important to safety, I shall restrict the remaining discussion in this paper to computer-based, safety systems since these represent the most safety significant aspects of systems important to safety. Furthermore, since the software of these systems is currently regarded as their most problematical aspect, I shall restrict the discussion solely to that area.

It is currently accepted that the techniques for quantitatively determining the reliability of a software-based safety system, even in terms of its probability of failure on demand, are not sufficiently mature: therefore, I do not consider that a safety case based on quantification alone would be acceptable at the present time. And even if a sound reliability quantification methodology were available, it is unlikely that this on its own would be accepted by regulatory authorities as a robust demonstration of the kind required for the safety systems of nuclear installations: there would still need to be, in my opinion, a demonstration of the application of the appropriate software engineering techniques. Thus given that quantification is not currently an option for the determination of the fitness for purpose of a safety system, then the safety case has first to establish the rationale for the approach adopted.

Since the production of safety cases in the UK nuclear industry is the responsibility of the licensees then the rationale will be determined by them. NII's approach, however, is to invoke the Special Case Procedure of the SAPs (P70 and P71) when reviewing the safety cases of software-based, safety systems. NII's assessors expect to see evidence of all the elements of the "special case procedure". As already mentioned the first part of this assessment procedure calls for the demonstration of a high quality production process (P179)\(^4\) by:

- the application of accepted international standards to the production process;
- an adequate QA programme and plan;
- independent verification and validation;

and an independent assessment of the finally validated software using:

- a suitable static analysis tool;
- a machine assisted comparison of the source code with the machine resident code;
- visual inspection of the code and supporting documentation;
- a programme of dynamic testing using input trajectories selected from the operational input space.

\(^4\) I should perhaps make clear at this stage that the paper assumes that a system requirements specification is available as an input document to the development process as has been stipulated in the Workshop Final Announcement.
In addition, there needs to be an equivalent demonstration of the validity of any fixed data used to customise the system.

Employing the techniques outlined in the Special Case Procedure is not claimed to produce software of known quantified reliability. The approach does, however, in the opinion of the NII, represent an acceptable method for producing software suitable for the safety systems of nuclear reactors, and for demonstrating the adequacy of that software ie. it can be considered as producing a reviewable system.

The rationale for NII's approach is predicated on the premise that software that is produced to the highest standards currently available, and whose final installed version and testing have withstood a searching independent assessment, will be suitable for safety system duty. This view that an excellence of production argument can be made in support of a safety system's fitness for purpose is widely recognised. What is perhaps unique to the UK is the requirement for independent assessment as part of the demonstration of fitness for purpose. NII currently believes that an adequate safety case requires strong evidence of the inclusion of techniques for error avoidance, error detection and error tolerance; and a searching examination of the product, and its testing, that confirms its quality by finding no errors that would prevent correct safety behaviour.

NII's approach to the assessment of the software of safety systems can thus be viewed as one perspective on the documented evidence and reasoning required for a reviewable system.

REVIEWABILITY REQUIREMENTS

In a paper such as this it is not possible to provide a comprehensive definition of reviewability requirements for all development aspects, however, some guidance will be given. Basing this guidance on the elements of the special case procedure, but without imposing the special case procedure's two-legged argument of excellence of production and independent assessment, I have set out below my views on the reviewability requirements.

The Production Process

As mentioned above the special case procedure for safety systems is interpreted as requiring, as one element, the application of accepted international standards. The application of such standards, obviously, imposes a discipline on the design and development process that aids reviewability. Also, since standards represent accepted good practice, the reviewer is given greater confidence in the final system. For example, IEC 880, which is currently deemed by NII to be the most appropriate for the conventional software aspects, requires the adoption of good coding practices and the prohibition of undesirable techniques.

A documented demonstration of compliance with any quoted standards, including a case-by-case justification for any non-conformance, aids the review process since the reviewer is presented with the arguments which may answer many of his/her concerns. To ensure reviewability of this document there should be, for each clause in the compliance statement, sufficient evidence to support either the claim of compliance or the justification for the non-compliance.

---

5 Note, however, that if a high reliability figure would seem to be required by the needs of the Probabilistic Safety Analysis, then it might be necessary to back up the software-based system by a diverse type of system, at least for the more likely faults.
The system design phase for the purposes of this paper assumes, as has already been stated, that a fully correct and unambiguous requirements specification is available. Of course, this is often not the case in reality, and parallel development and iteration take place. However, as far as the final design is concerned, this must be reviewable: its description should be clear and not encumbered by the iterations of the design process. And, of course, the specifications should be properly updated.

As a contribution to the reviewability of a system, consideration needs to be given during system design to the balance between hardware and software solutions to meet requirements: a simple hardware solution is preferred to a complex software solution. Also, the hardware and software architectures (for example, safety kernels, distributed architectures and modular configurable software) required to deliver these solutions can, if appropriately designed, contribute greatly to reviewability. Simple, understandable designs with no complex interactions between concurrent systems are preferred from the point of view of reviewability, unfortunately this may not always be achievable. What is essential, however, is a logical mapping from the requirements specification to the system design since it aids the review process by demonstrating that all the functionality described in the specification has been covered by the system's design.

Structured design techniques also play their part in the production of a reviewable system in that they aid understanding. The use of application specific graphical languages linked to code generators is accepted as reducing the number of errors introduced, and certainly reduces the number of production steps needing to be reviewed. However, the application of such tools merely transfers the reviewability problem to the validation of the tool since a full understanding of the mapping process and confidence in its veracity is required.

The application of formal methods to the proving of correct mapping from one level of design to the next would seem greatly to aid reviewability. However, this is an area still under development and will warrant particular attention in the future when these methods are of industrial strength.

The safety case, following changes to the software, needs to be sustained throughout the safety system's life cycle. Therefore, there should obviously be an adequate set of documentation in support of the change process. This documentation should include full records of all changes with an appropriate safety justification plus a comprehensive record of the impact analysis undertaken and the subsequent testing performed.

There is an obvious concern with regard to the possible incorporation of unauthorised code (such as viruses, logic bombs, time bombs and trap doors) into software-based, safety systems. The arguments regarding the security provisions need to be included in the safety case for the system to be reviewable. In addition, the evidence of the achievement of these provisions should be supplied.

Finally, the level of documentation required to be produced and retained in support of the safety case for a safety system is obviously inextricably linked to reviewability. This should be at least equivalent to the requirements of IEC 880 plus any additions defined in this paper.

**Quality Assurance**

It is generally accepted that the application of an adequate Quality Assurance (QA) programme and plan improves the end product of any process; therefore, it can be assumed that this will be the case with software. QA promotes a structured and disciplined approach to the production process that
helps ensure that there is sufficient auditable evidence available to enable a proper review of the process to be undertaken. Hence, a demonstration of compliance (by all organisations involved in the production process) with a QA standard equivalent to ISO 9000-3 (ref 6) is required for any reviewable system. Also, there should be well documented procedures for all aspects of the software life-cycle and reliance should not be placed solely on the quality of the staff involved. It does, however, have to be recognised that QA does not compensate for poor design, therefore, careful consideration needs to be given to other aspects of the review.

Independent Verification & Validation (V&V)

Verification of all phases of the production process and validation of the final safety system against its requirements' specification are important contributors to the final dependability of the safety system. A further contributor is the use of separate and independent teams supported by suitable static and dynamic testing tools. To aid reviewability these teams should produce suitable V&V plans which justify the level of test coverage to be undertaken. And these plans should also demonstrate in an understandable and auditable manner the traceability of the test cases to the requirements. The recording of test coverage (both in terms of code and specification coverage) should also be auditable. Summary reporting of this information should be included in the safety case with appropriate referencing to the original material. Furthermore, there should be a clear process for controlling the reconciliation of anomalous findings and recording this.

Commissioning is an activity which, due to its necessary involvement with the plant and the other engineering disciplines, is best regarded as separate from the verification and validation process. Nevertheless, it is an important phase in the development of any operational computer system since it represents confirmation or otherwise of the correctness of the computer system designers' model of the plant to which the computer system is connected. The documentation needs for reviewability, however, are similar to those mentioned above for the V&V process.

Static Analysis

The application of a comprehensive static analysis tool (whether by the V&V team or the Independent Assessor) which demonstrates that the source code meets its program specification has, because of the nature of the analysis, the benefit of exposing aspects of the system documentation that are not of a standard suitable for review. Clear, understandable and correct documentation is required because, in addition to a control flow analyser, a data use analyser, an information flow analyser and a semantic analyser\(^6\), there is usually a compliance analyser which extends the semantic analysis facility by providing automatic comparison of the analysed function with the program specification. Without good documentation the full set of analysers cannot be applied; hence, the application of a static analyser provides a good demonstration of the reviewability of the available documentation in terms of its understandability. The use of such a tool should be described in the safety case, and comprehensive records of its application, and the resolution of any of its findings, kept and referenced by the safety case.

\(^6\) Semantic analysis produces the exact mathematical relationship between inputs and outputs for each semantically possible path through the software.
Validation of the Compiler, Linker and Loader

A reviewable safety system needs evidence that the source code has been correctly translated into a form suitable for loading into the computer systems. The use of validated tools assists this demonstration: and, if this demonstration is to be reviewable, a full set of documentation describing the validation process should be available for inspection. A summary should, of course, be included in the safety case together with appropriate references. The NII, however, believes that the use of a reverse engineering process to compare the source code with the machine resident code is the most convincing demonstration that a correct translation has been achieved. It also has the benefit of demonstrating the absence of unauthorised code (such as viruses) that might perhaps be introduced during the linking process. This particular demonstration is not an easy one. NII's experience to date suggests that it is best achieved by a proven automatic tool which performs a reverse engineering exercise on the machine resident code and compares this with the source code. Again the activity needs to be fully reported in the safety case with the attendant recording of arisings and their satisfactory resolution.

Visual Inspection

The special-case-procedure's requirement for visual inspection of the code and supporting documentation by the Independent Assessment team places a need on the system supplier to provide the necessary software requirements specifications, software design documents and appropriately commented software listings. In order to ensure reviewability one might seek to ensure that the content of these documents was at least compliant with an appropriate national or international standard. Any findings and their resolution should be reported in the safety case.

Dynamic Testing

Whilst it is recognised that dynamic testing is part of validation, the special case procedure identifies it separately since part of its purpose is to create confidence that the reliability requirement has been met. In addition, the subjecting of the safety system to simulated input trajectories selected from the operational input space is a necessary extension of the validation exercise since it is not reasonably practicable to perform such tests on the plant. NII has accepted as part of a programme of dynamic testing the use of just one complete protection system train linked to a suitable test harness. In this case, the other three trains were simulated by the test harness. The correct response of the system is checked and recorded for each test. Here the credibility and reviewability of the activity is aided if both the checking (by an oracle) and recording are by proven automatic means. For a satisfactory review there needs to be a fully documented description of the test harness and the oracle which enable any failed runs to be analysed so as to determine the cause. Additionally, the analysis of failed runs should be fully documented in such a way that the arguments justifying the cause can be followed.

Customising Data

A feature of some system software, not covered by existing standards, is its facility to be customised to each application by means of data tables. The confirmation of the correctness of this configuration data is obviously a critical aspect in the verification and validation of the system. There, therefore, needs to be an equivalent demonstration of the validity of any fixed data used to customise the system. This should, of course, be fully reported in the safety case.
CONCLUSION

Safety cases, now required by law in certain UK industries as they have been for some time in the nuclear industry, require appropriate documented evidence and reasoning to be deemed adequate. This evidence and reasoning can be seen as the constituency of a reviewable system since the arguments support the fitness for purpose of a system. Given that a safety case based approach achieves reviewability, then the degree of rigour, and thus reviewability, required depends on the significance of errors or failures to the overall plant risk.

Since safety systems have the highest safety significance, and their software is the most problematical aspect, the paper has concentrated on that aspect. NII's assessment approach for safety systems software, known as the "special case procedure", has the benefit of indicating the elements of the development process which most require to be reviewable. These elements include, the application of appropriate national and international standards, a QA programme and plan, independent V&V, and dynamic testing. The application of a static analysis tool, reverse engineering and visual inspection also promotes reviewability since these techniques are dependent on good documentation. Understandability is the key to reviewability: and this is aided by the design's simplicity, and the use of a structured approach with design features and tests which are traceable to the requirements' specification. This traceability and the degree of test coverage achieved should be clearly documented to show that the design and testing are adequate. Other aspects that have been identified as requiring special attention are any customising data, change control and system security.

The paper has given an indication of the approach to the development of a reviewable system that needs to be adopted: and I feel that provided these broad principles are followed it is likely that a system will be demonstrably fit for purpose.

ACKNOWLEDGEMENTS

I should like to thank HM Chief Inspector of Nuclear Installations for permission to publish this paper. The views expressed are those of the author and not necessarily those of the NII.

REFERENCES

DESIGN FOR LICENSIBILITY - TELEPERM XS FROM SIEMENS

Heinz-Wilhelm Bock

Arnold Graf

SIEMENS KWU
Abteilung NLL
Frauenauracherstraße 85
91056 Erlangen Germany
E-Mail: graf001a@erls17.kwu.siemens.de

ABSTRACT

To rule out hazards to the public due to the operation of nuclear facilities, modern nuclear power plants feature a number of staggered safety barriers which complement each other in reliably preventing the uncontrolled release of radioactivity even in the event of an accident in the plant. The safety instrumentation and control equipment, as the last active safety barrier, triggers reactor trip and the active safety systems. For this reason, this equipment is the most important factor in any safety evaluation. This is reflected in the particularly stringent design requirements imposed on such systems.

TELEPERM XS is designed to meet these requirements. The underlying safety concept is derived from an analysis of potential failure causes and mechanisms and includes aspects of failure avoidance, early failure detection and failure tolerance.

A special engineering process was developed for use in TELEPERM XS applications, with this process incorporating a modern approach to maximize fault avoidance. A wide range of self-checking mechanisms coupled with essentially automatic in-service inspections allows test and maintenance effort to be reduced to a minimum, and this coupled with extremely high system availability. To assure the safety goal even in case of design faults the staggered safety barriers of the fluid system functions are kept independent within the I&C system. This is done by mechanisms to avoid failure spreading and to reduce the affected area of a failure to the faulty component.

INTRODUCTION

The safety philosophy of modern nuclear power plants is based both on passive barriers and on active measures which activate the final control elements of the safety systems. These active measures are in the first instance of a preventive nature, as they pre-empt the development of inadmissible operating conditions and thus counteract the incipient causes that could lead to malfunctions and accidents. If an accident should occur even despite these comprehensive prophylactic measures, further active measures are implemented to contain and control the consequences of the accident. Initiation and performance of these active measures are the responsibility of the instrumentation and control systems, which thus have a cardinal role to play in the safety philosophy followed in nuclear facilities.

The high safety standard in nuclear engineering calls for multiple staggered safety barriers within the instrumentation and control equipment as well. These barriers are classified according to their importance to safety and must be implemented with the aid of mutually independent equipment. The nuclear engineering codes and standards distinguish between operational equipment, safety-related equipment, and safety equipment. In normal operation, a nuclear power plant is automatically controlled by the operational closed-loop and open-loop controls. In these control loops, the essential process conditions such as pressures, temperatures and liquid levels are measured, compared with pre-defined reference values, and, if necessary, corrected by means of suitable countermeasures. If the operational equipment should fail, or if the countermeasures initiated do not have the desired effect, more powerful countermeasures are initiated by the safety-related instrumentation and control equipment.

The safety instrumentation and control equipment, as the last active safety barrier, triggers reactor trip and the active safety systems. For this reason, this equipment is the most important factor in any safety evaluation. This is reflected in the particularly stringent quality requirements imposed on such systems.

The requirements applicable are specified in the guidelines of the Germany Reactor Safety Commission (RSK) and in German nuclear safety standards (KTA). Safety I&C systems as implemented in Germany are essentially based
on a defence in depth concept (fig. 1) with the following properties:

- all safety goals shall be assured by a number of independent safety barriers
- integrity of each safety barrier shall be assured by adequate measures for fault avoidance, failure detection and for the accommodation of faults.
- for the last safety barrier implementation in N+2 configuration is required, with common-cause failures considered.

Although the nuclear codes and standards were essentially formulated for hardwired I&C systems, the underlying safety concept can nevertheless also be applied to digital systems without any great difficulty. The revised edition of the RSK guidelines for pressurized water reactors (draft edition of September, 1995) in fact confirms that these proven design criteria are also to be used for digital I&C system applications.

qualification by operating experience is supplemented by type testing of the components by independent experts to verify compliance with pertinent German and international codes and standards.

The system software was developed from scratch and qualified to meet the requirements of the relevant codes and standards, and is based on proven algorithms. On design of software architecture, major emphasis was placed on simplicity and on independence between the fluid system process and the behavior of the I&C system. In spite of these preventive measures, German licensing practice still requires that a common-cause failure has to be postulated. The analyses performed in this context and the system features that result for TELEPERM XS are summarized below.

Analysis of common-cause failures is made all the more difficult by the fact that the term „common-cause failure“ (by contrast with random failure) denotes a very general failure category with no possibility of more precise definition of their type or the affected area. Evaluation is, however, only possible if failure mechanisms are known, so that the type and affected area for the failure can then be estimated.

To nevertheless permit objective analysis, consideration in this context will be restricted to such failure mechanisms as cause simultaneous failure of several items of I&C equipment. This type of event is referred to as a simultaneous failure. On the basis of experience to date, failures that occur at uncorrelated times or with large time intervals between them will be detected in good time to allow appropriate corrective action to be taken. This is not valid for simultaneous failures.

A prerequisite for such simultaneous failures is, by definition, a common „trigging event“, with the affected area for this type of failure restricted to such items of I&C equipment as are subject to this event (fig. 2). The term „trigging event“ in this context does not mean that the event itself results in admissible operating conditions, but only in specific conditions which will result in failure if they coincide with a design lack or a hidden failure (e.g. caused by ageing). Trigging events can have various different properties, with a distinction to be made in this context between events that result in changed

---

**Fig. 1** Defence in depth concept

This paper will describe how these requirements are met by TELEPERM XS. Interest here will centre on the measures for the avoidance of design failures (by use of a focused design process) and the measures implemented to cope with common cause failures.

**ACCOMMODATION OF COMMON-CAUSE FAILURES**

Development of the TELEPERM XS system is based on analyses of potential failure causes and mechanism. The hardware used consists exclusively of proven modules for which robustness has already been demonstrated by their use in a wide variety of industrial applications. This
environmental conditions and events that result in a changed application profile.

Changed environmental conditions are typically caused by events such as fire, flooding, explosions, etc. A very critical case is if an accident within the plant simultaneously constitutes the triggering event for a failure of the I&C system, as in this case simultaneous failure of the I&C system will coincide with the onset of the accident which the I&C system is designed to control. The safety I&C is, however, installed outside the containment, meaning that only the transducers or even the sensors are directly exposed to the accident conditions. By contrast, the environmental conditions of the central I&C equipment are not affected by accidents. For other events such as fires or flooding, physical separation into trains ensures that the maximum affected area for this type of failure will be restricted to a single train. Appropriate measures are implemented to confine the effects of such failures to within small, defined sections within a building area or within a train (e.g. to a single electronic equipment cabinet). External events such as explosions, lightning and storms are accommodated not only by separation as above, but also by additional design measures implemented for all equipment of the safety I&C (e.g. seismic protection or use of optic-fiber links between cabinets).

A CCF is a breakdown of more than one component for the same cause in such a way that it is no longer able to perform its design requirements

Prerequisites: Lack in the design or production of the component and any initiating reason

- possibilities of occurrence
  - uncorrelated
  - simultaneous
  - if the common breakdown is caused by aging
  - if the common breakdown is caused by a triggering event

- spontaneously detectable if all relevant design requirements are continuously stressed
- hidden if the component only is required on demand

Fig. 2 Common cause failures

In the context of I&C systems the application profile is given by one or more complete sets of permitted input data. These data are governed by the measurement signals which mirror the fluid system process. This means that it is not possible to rule out a priori that changes in the fluid system process (such as occur under accident conditions) could lead to failure of the safety I&C. As software is not affected by environmental conditions and is not subject to ageing, failure of the software can only be triggered by a changed application profile, whereas the hardware of the central I&C is essentially unaffected by changes in measurement signals.

The fact that a failure of the software can only result from a specific application profile is of major consequence for the affected areas to be postulated for such a failure. From this condition one can conclude that simultaneous failure of independent items of equipment as a result of a software failure only need be postulated for equipment that is subject to the same application profile, meaning that the items concerned are used to implement the same fluid system functions. It likewise follows in reverse that in the case of a multiple barrier safety system where the separate safety barriers are implemented by discrete fluid system functions (functional diversity) and on independent items of equipment, a simultaneous failure can only affect a single (multi-channel) safety barrier.

Nevertheless for computer-based safety I&C systems, standard procedure is to use a single processor unit to implement several fluid system functions. According to the above assessment a fault within one fluid system function may cause failures of all functions implemented within the same processor unit. Reliability can significantly be increased if it is possible to assure that the affected area of such a fault can be restricted to the faulty function. This goal requires two important features of an I&C system:

a) The program flow within the I&C system has to be completely independent from the application profile of any application function.

b) Different fluid system functions have to be implemented in independent modules without any information exchange.

It follows from requirement a) that the application profile of any fluid system function cannot be a triggering event for a fault within the system software. It follows additionally that any fault within a safety function does not affect the system software.

From requirement b) follows that any fault within one safety function does not affect any other function. Both properties together assure that the response behaviour of any fluid system function is governed only by its own application profile.

The design principles on which TELEPERM XS is based ensures that the above independence is present, with this
verified by appropriate tests. Thereby these fault tolerance properties are much easier to be demonstrated than the freedom of design errors. That means reliability targets can be met by simpler architectures and safety demonstrations require less effort.

AVOIDANCE OF DESIGN FAILURES

It is nothing new to say that the best way to improve a chain is to strengthen its weakest link. In software development, the weakest link is the phase transition from the fluid system requirement specification to the software specification. This is one important result of some experiments on software reliability which were performed in the mid-eighties. The main problem in this context appears to be that, by contrast with the other phase interfaces, the first transition involves a language barrier. Software specifications are typically drawn up by computer scientists or instrumentation and control engineers, who may use means of expression that are unfamiliar to the process engineers who are responsible for the fluid system requirement specification. The consequence is that no effective verification measures can be installed, with the result that incomprehensible formulations, ambiguities, or simple misinterpretations of the system requirement specification remain undiscovered. This applies in particular when strictly formal methods are used to formulate the software requirement specification. Such methods typically require a sound command of mathematical logic and are used because they are very effective in reducing errors and costs in design and code generation.

This leads to the absurd situation in which the use of stricter, more consistent measures for the software specification only brings the software faults forward from the later generation phases to the first phase transition, but without bringing about any genuine reduction.

Software specification

The most important feature to be demanded of a suitable software specification language is that the software specification must be intelligible to all parties involved in the relevant phase transition. To our knowledge, there is currently no one procedure that enjoys universal recognition. This indicates that it is no trivial matter to fulfill this seemingly simple requirement in practice. The main problem appears to lie in the requisite universality of such a procedure, which must be suitable for formulating system requirements for the most diverse applications of the most diverse complexity.

In the case of instrumentation and control applications - and in particular in the field of safety instrumentation and control - the system requirements are typically very simple and are defined by the process engineers. Advantage can be taken of this "limitation" to define a "specification language" which is suitable on the one hand for formulating with adequate precision the requirements to be made of the software and which on the other will be understood both by process engineers and by instrumentation and control engineers.

---

![Diagram](image)

Fig. 3 Formal software specification

Formal methods of formulating the requirements to be made of the software are understood to be mathematically-based methods that suitably describe the specific properties of the software. Such methods provide a highly systematic framework for the specification, development and verification of software. A method is said to be "formal" if it is mathematically based. This forms the basis for a precise definition of terms such as "consistent", "complete" or "correct".

It is standard practice to describe automation tasks by means of algebraic, differential or integral equations. While this form of representation is of great assistance in the analytical treatment of simple, linear automation tasks, it can lead to lack of clarity when applied to complex automation tasks. For this reason, it is expedient for the specification of real-life automation tasks to transform this mathematical representation into a graphic representation by assigning each operator a graphic symbol and defining rules for linking up the operators (fig. 3).
Just such a transformation yields the familiar function diagram which in the past has proved to be a useful aid to the specification of instrumentation and control tasks. The symbols used in the function diagrams are known as function blocks. They stand for the elementary semantic units and thus constitute the vocabulary of a formal language. From the point of view of implementation, the function blocks represent the elementary functionality which can be used to solve fluid system tasks. The use of function diagrams in the specification of instrumentation and control functions is thus a formal method, and the function blocks together with the rules for their logic combination constitute a formal specification language.

Experience has shown that the specification for the entire reactor protection system for a nuclear power plant can be drawn up in a readily comprehensible manner with the aid of only about fifty different function blocks and very simple logic combination rules. Such specifications are equally intelligible to process engineers as to instrumentation and control engineers and they afford all the advantages of formal methods.

The use of computer-based tools in this process facilitates not only the efficient generation of the software specification by providing, for example, suitable graphic editors, it also permits detailed analyses to be performed from the most varied of aspects. For instance, checks can be performed not only to ensure that all syntactic rules have been observed but also to verify further essential features such as the unambiguity and completeness of the specification. The quality of the software specification that can be achieved in this way goes far beyond the quality attainable with non-formal methods. The following procedural model can therefore be established for the generation of the software specification: the system requirements specification is drawn up - as usual - for the most part by mechanical engineers, process engineers and physicists in the form of text, diagrams, equations and tables and is submitted to the instrumentation and control engineers in this form. This task definition is read and understood by the instrumentation and control specialists - who may include control engineers, communications engineers, physicists, computer scientists and mathematicians. System discussion meetings are held to clarify any questions of interpretation. The instrumentation and control engineers for their part then process all the available data and information in a pre-defined, always identical manner and use the editor of a computer-based specification tool to produce the instrumentation and control solution to the defined task, i.e. the software specification(fig. 4).

The software specification is then subjected to a detailed automatic analysis to verify its consistency. Finally, the consistent software specification is again verified in detail by the party who had defined the original task, i.e. the originator of the system requirement specification, and is released for further use.

The formal software specification thus generated and verified constitutes the reference basis for all subsequent phases in the generation of the software. As in all formal measures, statements as to the quality of the software, e.g. its correctness or freedom from errors, can be referred only to this software specification and not, for example, to the system requirement specification. For precisely this reason, it is of decisive importance that the formal software specification must be not only unambiguous and consistent but also and in particular intelligible to all parties involved in this phase of development.

![Diagram](image)

**Fig. 4** The engineering process

**Code generation**

Code generation is very much facilitated by the fact that a strict formal procedure was used for generating the software specification. It follows that all relevant properties of the software have been positively defined by the specification, so that the code can be generated from the specification in accordance with a given set of rules. What is more, the procedure uses a very simple language based on less than fifty function blocks of very simple
functionality. This in turn means that the code can be generated in a very simple manner and has a simple and orderly structure.

If code generation is based on the simple functions implemented in the function blocks, this process can be reduced to generation of the corresponding block calls and to linking up the blocks by means of data structures. This means that the code generated consists of a sequence of block calls statically connected via data structures. It is apparent that this task can ideally be performed by a generator representing a compiler of the most simple kind. This yields a maximum of reproducibility, permitting quality characteristics for the code generated to be derived from the properties of the generator. In this way, the rules implemented in the generator ensure that, for example, branches or dynamic resource management can be largely or completely dispensed with.

But automatic code generation also affords other advantages. For instance, it eliminates the need for comprehensive design and implementation documentation. In conventional software development models, design and implementation documentation are required to record the results of the development phases following on from the software specification. Since, in the procedure used here, the code is generated in accordance with a given set of rules, these records can be replaced by documentation which specifies the universally applicable rules. This not only significantly reduces the effort required for generation and review of the documentation but also makes it easier to keep them up to date at a high quality level. Changes to the software have to be documented only in the software specification, while the rules governing code generation remain unaffected.

The basic functions specified with the aid of the function blocks can be made available in software form as non-memory library functions. The simplicity of the functionality makes it possible to use familiar procedures to verify correct implementation, which can also be formally confirmed by means of a type test. This measure is directly comparable with the use of qualified hardware modules for hardwired instrumentation and control systems, which commonly feature a similar functionality.

Verification and validation

The immediate consequence of automatic code generation in accordance with a given set of rules is that the code thus generated can also be automatically verified by implementing the "inverse rules" in a verification tool. This verification demonstrates that the code generated actually fulfills the software specification. Since both the task definition and the functioning of this verification tool are completely different from those of code generation, the two tools can be considered independent of each other. It thus follows that the demonstrated quality characteristics of the code generated by the generator tool can be significantly enhanced by the independent verification. Automatic verification replaces the code inspection required for conventional software development measures, but demands less effort and yields much better quality.

As a consequence of automatic code generation, functional software tests take on a new significance. While in the case of conventional software development measures the tests served to identify errors from all phases of development, in our case their purpose is only to reveal errors in the software specification. As far as the subsequent code generation is concerned, these tests serve only to verify the quality of the software.

It is also worth mentioning that automatic code generation also simplifies testing quite considerably. For instance, the generator used to generate the code can also be easily used to generate a test environment providing all the aids required for efficient testing. These aids make it possible on the one hand to subject instrumentation and control functions to any desired signal transients so as to provoke selective responses, while on the other hand they grant access to all internal signals and storage areas so as to permit simple evaluation of the functional response.

If a realistic feedback from the process is required for evaluation of the functional response, the code generated can also be linked with a process simulator. In the course of such functional tests, the process simulator simulates the thermohydraulic power plant process and supplies the necessary measurement information to the instrumentation and control function. The instrumentation and control function - that is to say the test object - responds to this measurement information by issuing commands to the final control elements, which in turn influence the thermohydraulic power plant process.

Fabrication of the hardware at the manufacturing plant proceeds in parallel with the functional test, so that integration of the system in the test facility can commence immediately thereafter. The final system validation performed in the test facility allows no credit for the software generation procedure. For the purpose of system validation, the instrumentation and control is treated as a black box whose response must be checked out as comprehensively as possible against the system requirement specification.
Application of Guidelines for Review of Software in a Programmable Reactor Protection System.

by

Gustav Dahl - Institutt for Energiteknikk/OECD Halden Project, Norway

Hans Eriksson - SKI (The Swedish Nuclear Power Inspectorate)

ABSTRACT

The paper describes the experience gained on the application of a set of “Guidelines for Reviewing Software in Safety Related Systems” during the review of a programmable safety critical system before installation in a nuclear power plant. The Guidelines were produced for the Swedish Nuclear Power Inspectorate (SKI) by Institutt for Energiteknikk in Norway. The actual project (the REPAC project) was an exchange of a current, outdated system (the signal conditioning and bistable part of the protection system) at the Ringhals NPP with a microprocessor based system.

The Guidelines are divided into chapters and sections according to the different activities made during the development of the system: General safety aspects, specification analysis, quality assurance, system development, V&V, use of pre-existing software, configuration management, system integrity, installation and maintenance, training and operation, and documentation. The text is divided into general explanatory information and a set of short recommendations to SKI. The paper goes through the recommendations, and make comments and evaluations on how each of these are handled in the REPAC project.

1. INTRODUCTION.

The introduction in Sweden of programmable equipment in safety systems of nuclear power plants have been done successively. The exchange of a neutron trip channel in the Barsebäck plant gave the first experiences. A diversification with an analog channel was made in order to get additional safety as experiences with programmable equipment in safety systems were few in Sweden.

The next step in the development, the REPAC project (REactor Protection And Control system exchange), was an exchange of the analog signal conditioning in the Ringhals PWR safety system with a microprocessor based system. This exchange is dealt with in this paper. It is very important to evaluate the experience from this project, as the one-to-one exchange in Ringhals is more simple to handle from a safety standpoint than future exchange projects, which will include larger parts of the safety systems and involve more complicated software.

To have a base for inspection activities the Swedish nuclear power inspectorate (SKI) signed a contract with Institutt for Energiteknikk at Halden/Norway to produce a set of “Guidelines for Reviewing Software in Safety Related Systems”. The aim with the Guidelines is to support the planning and carrying out of the inspection programmes. The inspection programs for different applications can vary depending on e.g. the safety significance, the chosen manufacturers, the design of the software etc.

The plant is designed according to a set of deterministic design rules which is documented in a safety report for the whole plant (FSAR). In a special safety report for the REPAC project it was shown that the REPAC exchange is fulfilling the FSAR requirements.

The intention of the paper is to evaluate the experience gained on the practical applicability of the Guidelines and on how the various recommendations of the Guidelines is fulfilled in a real project. The experiences related in this paper shall thus be used as a basis for further development in the inspection programmes to give preparedness for coming modernisations where the complexity is increased.

The Guidelines contain general explanatory information and a set of short recommendations on the main aspects important to the development and assessment of a safety related system: general safety aspects, specification analysis, quality assurance, system development, V&V, use of pre-existing software, configuration management, system integrity, installation and maintenance, training and operation, and documentation. These topics constitute the main chapters in the guidelines, and are also the main sections of this paper. The structure of the paper is such that the general text and the recommendations of the Guidelines are written in normal letters, whereas descriptions of what was particularly made in the REPAC project are written in italics. However, first an outline is given of the main principles on which the guidelines is based.

2. PRINCIPLES FOR THE GUIDELINES.

The objective of the approval of a new system to be installed in a plant is to demonstrate that this system (later referred to as the target system, or just the system when it cannot be misunderstood) will not jeopardise safety. Safety means that no accident, i.e. an unintended event with severe consequences, will occur during plant operation. The objective of the Guidelines is to assess software, but software in itself does not cause any accident. Safety is a global property and becomes meaningful only if the target system is considered together with the plant and the environment.
In order to establish safety requirements it is therefore necessary to identify not only the influence the target system has on the plant, but also the possible impact the plant has on its environment. This is used to identify potential faults in the software of the target system which may cause safety failures. The objective of the software safety requirements should then be to minimise or eliminate these. Having identified the target system safety requirements, one should relate them to the remaining target system requirements specifications, in order to identify potential conflicts, inconsistencies and omissions. This step demonstrates that the functional specification of the system is safe and classifies the functions according to their criticality to the safety goals.

A computer based system will probably to a large degree consist of software components, and the complexity of the software is often a main cause of problems. Limiting and mastering this complexity is one of the primary goals of a good software design process. A separation of the system functions into safety-critical and non-safety-critical classes should also be preserved in the software design. Extra effort can then be devoted to validate the safety critical software modules.

Safety dominates over availability in the sense that if the two goals conflict then availability should be sacrificed in order to maintain safety. However, both safety and availability are closely correlated to the quality and reliability of the software, and the major part of the Guidelines is devoted to this aspect. The first principle in this respect is fault avoidance through good software engineering and quality assurance throughout the complete lifecycle of the software. The second principle is fault detection through a thorough validation and verification activity. A third principle, which should also be considered is fault tolerance, i.e. the target system should be designed so that a failure will not jeopardise safety. This could be made so that the failure has no effect, e.g. through redundancy, or by bringing the system into a safe state, or into a state of reduced risk, in case of a failure.

3. SAFETY ANALYSIS.

An identification and analysis of all aspects with potential safety consequences should be made before one starts to develop, as well as to assess, a safety related system. This includes aspects which are of general nature with respect to safety, as well as aspects of the particular system which shall be implemented.

3.1 Overall safety aspects.

Necessary detailed information of the plant and its behaviour must be available, as well as the results from any safety analysis of the plant. Questions which may be raised are

- Which hazards or accidents may occur?

As REPAC is a one-to-one exchange of old analogue equipment in an existing plant, all information about the plant was thoroughly documented in advance. In this connection also a safety analysis had been made, and the result of this was documented in a final safety report (FSAR). It was shown in the safety report from the REPAC project that REPAC fulfills the requirements in FSAR.

- Is it possible to bring the plant into a safe state, or to a state of reduced risk, in case of a hazard situation?

A shut down will bring the plant in a safe state if a hazardous failure in the protection system occurs.
• Are there some superior plant protection system, which acts as a last defence if the target system should fail?

*The initiating events lead to reactor trip or other protective functions by a set of activated protection channels. A substantial part of the initiating events which activates the REPAC system, in particular the most critical ones, is also activating protective functions in a protection system based on neutron flux measurements, using analog technique. An ‘umbrella’ is thus given by signals from the analogue equipment as back-up for the REPAC system.***

• Are there any general safety principles which is applicable to the system?

*The original classification system for I&C systems at Ringhals contained two classes: ‘safeguard’ and ‘non-safeguard’. In 1986 a new classification scheme based on standards from USA was introduced. It contain three classes: 1E for safety critical systems, 2E for systems for normal plant operation, and 3E for auxiliary systems. REPAC consists of a protection system, classified as 1E, and a control system, partly classified as 1E and partly as 2E. In this paper we are only considering the protection system.***

3.2 Safety analysis of the specification

Even if a system is not made, it should be possible to make a safety analysis of it in advance, solely based on the specifications of the system. But also when the system is completed, e.g. in audits with the licensing authorities, the specification is a good basis for safety analysis.

A clear boundary between the target system on one side, and the plant and environment on the other must be defined. In this respect one must also establish the interfaces through which the target system interacts with the plant.

Three aspects of the specification are of particular importance: the definition of the system, the functional specification, and the specification of particular safety defences.

*The REPAC system is clearly defined as a one-to-one replica of a part of the existing protection system, and the interface between the new system and the rest of the plant should be exactly the same as for the existing one.***

3.2.1 The functional specification.

The functional specification is a document stating what the target system is intended to do, and what it actually does if it contains no faults.

*As the REPAC system is a replica of an existing system, the functional description of the existing system can be considered as the functional specification of REPAC. This is given by a set of functional diagrams and a document describing precautions, limitations and setpoints.***

However, even if the system does not contain any faults, it is not necessarily correct nor safe. The system may be incorrect if the functional specification is faulty. It may be based on a wrong process model, it may contain internal inconsistencies or ambiguities, or it may contain direct (clerical) faults. Even if the specification is correct, in the sense that it correctly expresses the intention of the specifier, it needs not be safe. It may contain inadvertent side effects which may jeopardise safety. A thorough analysis of the specification is thus necessary for the safety assessment.

*The safety analysis of the existing protection system specification has been made. It is documented and accepted by the safety responsible at Ringhals as well as by SKI. As the documents on this system can also be considered a specification of REPAC, this safety analysis also fulfills the requirement of safety analysis of the specification.***
3.2.2 Specification of safety defences.

In order to enhance safety and protect against undesirable consequences of failures in the target system, particular safety defences could be specified in addition to the functional specification. If one can otherwise show that the system have an acceptably high reliability, such defences are not strictly necessary. The inclusion of such defences will, however, enhance the confidence to the safety of the system.

A commonly used way to gain safety is through diversity, i.e. to obtain the same functional goal through different means. The highest degree of diversity is obtained through functional diversity. This can be obtained if the same functional goal can be reached with completely different functions.

Functional diversity was essential to reach the safety goals of the REPAC protection system. Such a system is designed to protect against the consequences of a set of initiating events, like uncontrolled boron dilution, loss of feedwater etc. The REPAC protection system is divided into a number of protection channels which each performs a protective function, e.g. a reactor trip, based on a measured or computed process variable. An analysis of the REPAC specification was made to identify the protection channel(s) which activates the appropriate protection function for each of the initiating event. This showed that each initiating event according to FSAR is handled by at least two protection channels as specified in the deterministic safety criteria for the whole plant. If each protection channel is performed by a distinct processor in the final system, one has obtained a complete functional diversity.

To protect against hardware failures the REPAC system is highly redundant, and emphasis has been made to optimise the protection against hardware failures in the distribution of the protection channels over the processors in the redundant protection sets. Protection channels within the same set which are used to protect against the same initiating event are always handled in different processors.

Another type of safety defence is to design special checks into the target system. These can detect failures or abnormal behaviour which may threaten safety. When such failures occur, special actions should be performed, either automatically or by an operator. The REPAC system contains various types of self-checks with alarm indicators.

3.2.3 FMECA and Fault Tree analysis.

The Guidelines recommend the use of FMECA (Failure Mode, Effect and Criticality Analysis) and Fault Tree Analysis based on the specification. These methods are well known methods from reliability and safety analysis. They have mainly been applied to hardware systems, but could also be applied to software based system. The functional modules, as identified by the functional specifications, are the components of the system to be analysed. The objective of this analysis is to identify potential failures in the system, and the consequences these may have on the plant. In this way one is able to identify the potential failures which are most safety critical, and which therefore require the most attention.

FMECA is a bottom up analysis, starting with the components, and gradually increase the scope to analyse total functions. The functional diagrams of the REPAC specification is made up by a large number of components of different types: bistables, filters etc. There are, however, a fairly small set of types, and a FMECA for each type should therefore be practicable. This method is a systematic way to analyse that the consequences of all potential failures has been considered.
The general design philosophy of the plant is based on a set of deterministic safety rules, although fault tree analysis has been performed in connection with the basic safety analysis of the plant. This analysis was, however, not directly applicable to an FMECA and fault tree analysis of the software modules. Furthermore, this type of analysis applied to software is not yet fully developed, and is a target for further research.

4. QUALITY ASSURANCE.

The quality assurance policy of a company should at least contain the following:

- an overall quality assurance policy document, based on well-known standards
- qualification routines for software based safety related systems
- a detailed quality assurance document for managing software projects.

All activity at Ringhals follows a general quality assurance system. It was originally based on the US 10CFR50, and revised in 1991 (to conform with the standard of ISO 9000/) and in 1994.

The basis for the quality assurance of the software in particular is the Ringhals document “Programmable control equipment at Ringhals”. This is based on IEC 880/ and /ANS93/, as well as other standards

In environments where the quality of the software product is essential from a safety point of view, only well-reputed vendors must be chosen for the system development. It should be mandatory that the vendors are able to supply information on previous delivery projects within the same kind of environment.

The manufacturer of pre-existing software (see section 7) must also be qualified, to show that it can be accepted as vendor of safety related software. It would also be favourable if the manufacturer could substantiate that it generally makes products of high quality and reliability. To ensure the highest level of quality assurance, an external QA control should be performed by an independent team, i.e. persons not directly involved in, and with no direct responsibility for, the system development.

The vendor, Foxboro, which is also manufacturer of pre-existing software used in the project (SPEC200 Micro), is a well reputed company, which has delivered many systems of different kind to NPPs. A quality evaluation of the vendor was made by the power company Vattenfall/Ringhals. A evaluation of the vendor has also previously been made by USNRC and NUPIC in a similar project in USA.

5. SYSTEM DEVELOPMENT.

The system development of a software project starts with the functional specification, and results in a final program implemented on a host computer.

5.1 Development model and project plan.

In order to realise delivery of software products, it is needed to go through certain development activities. A system development model is a mean to control the development, to measure progress, and to include reviews, audits and tests.

Although the phrase ‘development model’ is not explicitly mentioned in any REPAC document, it could clearly be seen from several documents that such a model was implicitly followed.
A well organised project is a must to reach the quality and safety level which is required according to what has been described in the system's safety analysis. The existence of thorough project plans, reporting schemes and the production of the right documentation, are key items for ensuring a successful project outline and to end up with a quality product. This is documented in the "Quality Activity Plan" in the REPAC Project Manual.

5.2 Requirement specification.

The requirement specification (functional specification, safety specification etc.) is the basis for the further development. A good specification document should be correct, unambiguous, complete, consistent, verifiable, modifiable and traceable.

As mentioned above, the REPAC system is a replica of an existing system, so the functional documentation of the existing system can be considered as the fundamental functional specification of REPAC. All information about the existing system has been thoroughly documented, and reviewed and accepted by SKI. A new review in connection with the REPAC project was therefore not considered necessary.

5.3 Software Design.

The Guidelines recommend that the software should be designed using well-known and widely accepted design techniques.

The functional diagrams and other specification documents were remade by the vendor into a form suitable for further programming. The design document was written in a form comprehensible for the Ringhals staff, who reviewed them in several iterations, to clear up misconceptions and correct faults. The design documents were finally accepted as correct by Ringhals.

The design reviews followed the recommendations in section E 3.1 of ANSI93 and section 6.2.2 in IEC880.

5.4 Programming related recommendations.

5.4.1 Languages

Within one single project, one should restrict the number of programming languages to a minimum. Where possible, use of high level languages are recommendable, since these are easier verifiable by human inspections. In extremely safety critical applications, it could be useful to use more low level languages, but these should then be verified by automatic code verification tools.

REPAC is coded in a language specific for SPEC200 Micro. It is certainly not high level, but it is not like assembly code either. It contains a library of standard function modules often used for control purposes, but is not suited for general programming. These modules are described in detail in a user's manual. The limitation of the language protects against some typical programming errors (e.g. in loop handling and interrupt handling). There are no available automatic code verification tools for SPEC200 Micro programs.

5.4.2 Coding guidelines

Relatively strict guidelines should be closely followed when producing code. One should have in mind that the code should be easy to read for an outsider, which means that the code must be made fairly simple. One should avoid tricky and fancy programming, code compression etc.

Some particular programming was necessary to squeeze the REPAC programs into a limited space in the processors. Fancy programming is impossible in this programming system.
Encouragement should be given to provide
- easily understandable code
- use of strict naming conventions for variables
- modularity (small modules appreciated)
- initiation of all variables
- program comments sufficient that outsiders can easily understand the code.

These recommendations, with exception for modularity, are difficult to follow in the SPEC 200 Micro-language. The code is broken up in a way which makes it difficult to understand, naming convention is impossible, initiation of variables are implicitly set to 0. In-line code commenting is impossible.

The Guidelines recommend that design, coding and implementation of the system software are made according to a generally accepted standard or guideline (e.g. /IEC-880/), or to a local standard which is in agreement with a general standard. It should also be clearly stated how the standard is used, e.g. if only parts of the standard are relevant, or if some requirements in the standard are relaxed upon.

General coding standards, like e.g. IEC 880 is not completely applicable for this type of language. On the other side, the coding language of SPEC200 Micro has so few options that the coding standard is more or less given by the structure of the language.

5.5 Reviews, audits and inspections.

It is of greatest importance to the vendor, the power company and SKI that the delivered product is of required quality, consistent with the constraints of cost, delivery time and functional and safety requirements. The Guidelines therefore states that inspections and walkthroughs are highly recommendable, and should be mandatory whenever possible.

In addition to the ordinary review within the Foxboro and Ringhals REPAC project staff, code inspections were performed by an independent (of the REPAC development project) expert at Ringhals. The method used was to make separate programs based on the same functional specifications (a kind of diverse programming), and then to compare this with the actual code made by Foxboro.

5.6 Formal software development methods.

The Guidelines states that the use of formal software development methods should be encouraged, in particular for safety critical systems. These are methods which provide a mathematically based framework within which specification, development and verification of software systems can be done in a systematic and precise way. The use of formal specification and design makes it possible to discover many errors which might otherwise very easily be overlooked.

At the present state of the art, however, the use of formal methods should not be mandatory. However, more experience with the use of such methods, and further development of associated tools, could mean that formal methods will be mandatory in the future for highly safety critical systems. Formal development was not used in the REPAC project.

6. VERIFICATION AND VALIDATION.

The concepts verification and validation (V&V) are not exactly the same, but they cover the same general purpose, viz. to show that the target system performs its intended tasks correctly. V&V is an activity which should be made, not only on the final product, but in every step of the development. As V&V activities are often expensive, and too rigorous V&V requirements on all safety relevant systems would prevent any fruitful progress in this area. The strictest requirements to V&V should therefore be concentrated to prevent the failures with the most serious consequences.
Some of the standard modules was frequently used in REPAC, in particular the limit checking modules. These were considered particularly important for the safety, and therefore analysed carefully. These modules were, however, very simple, and it was possible to perform a complete V&V of these, including an exhaustive testing. Some other standard modules are more complicated to verify, but it was shown that through the use of functional diversity (see section 3.2.2), failures in these would have no direct safety consequences.


A detailed plan of the V&V activities should be established at an early stage in the development. This plan should describe all the planned V&V activities which should be followed. This should result in a V&V document which forms one basis for the safety assessment and the final acceptance by SKI.

A general V&V plan for programmable equipment exists at Ringhals, but there were no specific REPAC V&V plan from the beginning of the REPAC project. However, such a plan has evolved during the project, so the V&V activity were made according to a documented plan.

6.2 V&V methods

No method, even not formal proofs, can guarantee correctness with 100% confidence. V&V can actually only show the presence of faults, it cannot prove the complete absence of faults. However, the more one search for faults, the higher is the probability to find all residual faults. There are several different types of methods which are complementary to each other, and the use of a combination of these methods will increase the confidence in the target system. These can be divided into some main classes, in particular static analysis and testing.

6.2.1 Static analysis.

Static analysis is defined as the process of evaluating a computer program without executing it. The main objective of the static analysis is to check that the final program conforms with the specification or design documents, but it is also used to reveal defects in the program. These defects may be direct faults, but they may also be violation of coding standards. The Guidelines states that static analysis should be made for safety related programs, and is mandatory for particularly safety critical parts.

The static analysis performed with the REPAC system was the design and code reviews mentioned under sections 5.3 and 5.5.

6.2.2 Testing.

To test a program is to execute it with selected test data to demonstrate that it performs its task correctly. Ideally the test data should be selected so that all potentially residual faults should be revealed. The Guidelines states that testing is mandatory for any program. The amount of testing depends upon the criticality of the program.

The test of the REPAC system was made in both a Factory Acceptance Test (FAT) and a Site Acceptance Test (SAT). The most extensive was FAT. All tests were made with the complete system, i.e. that the test was made with simulated analogue process inputs, as they occur in the plant, and similar for the output. This has the advantage that the testing situation is realistic, and that all links from input to output is tested. On the other side, however, this makes it more difficult to emphasis the testing of certain more complex modules.
6.2.3 Analysis of real time aspects

An aspect of safety critical systems which must be particularly focused is the real time aspect. One must verify that the system can fulfill its tasks within specified time limits. Both static analysis and statistical testing can be used in this respect. A thorough analysis of real time aspects must be performed for a safety critical system.

The real time aspect of REPAC is quite simple. All programs start at regular time intervals, as it has been shown that these intervals are by clear margin sufficient for the computations. The computation time is not dependent on input data. The programs reads process input data at the start moment, and uses these data throughout one computation cycle.

7. USING PRE-EXISTING SOFTWARE PRODUCTS.

Pre-existing software products (PESP) are software modules which are made as standard components for use in a broad application area, and not for a particular safety relevant purpose. Such software is to an increasing degree being used in digital control and supervision systems, also in safety critical applications. Even if PESPs in principle shall fulfill the same requirements as tailor made software, there are some particular characteristics of PESP which necessitates special guidelines for the approval process.

The standard modules, as well as the configurator of SPEC200 Micro can be considered PESPs, and the requirements to PESPs therefore yields for these.

A main advantage with PESPs, concerning their dependability, is that they are often extensively used in a multitude of applications. A long history of fault free application is an indication of high dependability. this is often referred to as 'Proven Design'.

A problem with PESPs, on the other side, is that they are often delivered without appropriate information on the process followed during its development or on the final product itself. It may thus be difficult to assess whether the system has been developed according to the standards required for safety critical software. It may also be difficult to obtain sufficient product information, as e.g. code listings, to perform an assessment and approval. Access to this information may be denied due to company policy, or the relevant information may actually be non-existing. It may also be difficult to gather sufficient field experience data from the producer to be able to compute any reliability figure.

Another problem has till now been that there has been a lack of internationally accepted standards supporting the use of PESPs in safety critical applications. However, a supplement of IEC-880 has been proposed by IEC SC 45A/WG-A3 (/IEC-sup/), and it contain a set of guidelines for PESPs.

7.1. Requirements to PESPs.

The selection of PESPs for use in safety related environments should be made very carefully. Requirements should be put on the manufacturer of the PESP, on the development of the product, as well as on the product itself. The manufacturer should be judged by previous experiences on products and services by the same. An official approval (e.g. ISO approval) would be a prerogative.

As stated above the manufacturer, Foxboro, is a well reputed company, which has delivered many systems of different kind to NPPs. Foxboro and all subcontractors to REPAC have a third party ISO certification.

The PESP should have been developed and maintained according to an accepted standard for software engineering and quality assurance. Software used in safety critical applications should be developed according to a standard for safety critical software (e.g. IEC-880).
SPEC200 Micro is a fairly old system, which were developed before the presently available standards for safety related software were established. The development procedure was, however, documented, and a retrospective comparison of this procedure with several standards (IEEE 603, IEEE 7-4.3.2, IEEE730 and IEC 880) was documented. Most of the requirements proved to be fulfilled. Identified deviations were accepted by Ringhals.

The functioning of each PESP module must be clearly specified, in accordance with the requirements to specification.

A documentation of the SPEC200 Micro system, with a specification of all available modules, is available.

7.2. Evaluation of PESP's.

In order to be accounted for in the assessment of a PESP, the product profile, i.e. distribution, usage and user experience, must also be documented.

Several documents referred to wide user distribution and experience, and also referred to other more detailed documents. The modules used in REPAC have also been used in 3200 applications in more than 300 plants. Although most of these are not nuclear power plants, the program modules are identical. The experiences from these are therefore relevant for REPAC. The last release of SPEC200 Micro was in 1991. Since 1986 no critical faults have been discovered, and no changes have been made, in any of the SPEC200 Micro modules which was implemented in the REPAC system.

A statistical analysis of the system based on test data was referred in the documentation. However, these documents did not contain enough information to give a quantitative measure of the reliability of the PESP used.

One shall promote a product evaluation according to the recommendations given in the /IECsup/. The power company shall demonstrate to SKI how this is made, and document any deviations from these recommendations.

In agreement with the recommendations given in /IECsup/, the correct functioning of all PESP's used in the REPAC system was tested with the same rigour as required for a safety critical the application program.

It must also be demonstrated that the PESP software implemented on site is actually identical with the software it claims to be.

SPEC200 is a quite stable system, i.e. that there has been no new version releases for many years. There is therefore no reason to believe that the implemented programs are not the correct ones. But this must of course also be checked in the installation process and during SAT.

8. CONFIGURATION MANAGEMENT.

A configuration management plan should document the method to be used for identifying software product items, controlling and implementing changes, and recording and reporting change implementation status. The plan should be applied to the entire software life cycle and is especially important in environments where software could impact on safety measures.
The Guidelines recommend that a proper configuration management plan is made and that all procedures in this plan are followed. Procedures for change control and revision control must be part of this plan. No short-cuts should be allowed in relation to this plan.

Ringhals has developed a preliminary version of a configuration management direction. This is not specific for REPAC, but shall form the basis for configuration management of all programmable equipment of safety class 1E and 2E implemented at the Ringhals plants. Even if the direction is preliminary, it is so complete that one can conclude that it is in accordance with the intention of the recommendations of the Guidelines.

There are several key items in a CM plan, like organisation, identification of items, change control procedures, and revision control.

A particularity of the REPAC system is that change of calibration values and set points requires reprogramming and reloading of the REPAC software. This is a potential cause of common mode failures, and requires very strict change procedures.

9. SYSTEM INTEGRITY.

Precautions should be taken to ensure that the target system is not inadvertently changed during its operation. It must be demonstrated how the program and essential data are protected from being overwritten during operation.

The structure of the REPAC programs, with minimal storing of data, and with all data storing made local, makes it highly improbable that inadvertent overwriting shall occur.

Random hardware errors and external events (e.g. electrical chocks) may destroy the content of memory cells. The software components in a safety related system shall therefore always contain protections against negative consequences of hardware errors.

The system contains various types of self-checks with alarm indicators.

If the safety related target system have the possibility to be influenced by a human during operation, there is a potential risk that a maloperation may jeopardise safety. Certain precautions should therefore be taken to prevent the consequences of such maloperations, unintended or intended. The latter includes bypassing of safety functions as well as unauthorised use of the system.

There are no possibility for an operator to interact with the system during operation. The actual system is locked into cupboards in a room with strict access control.

All errors which occur in a safety related system during operation should be logged and documented.

All errors in safety related systems at Ringhals are reported to SKI and discussed in special meetings between SKI and Ringhals.

10. INSTALLATION, MAINTENANCE, SYSTEM TRAINING AND OPERATION.

To preserve the high level of safety of a software system also in the installation and later maintenance phases of the system life cycle, proper plans and procedures are mandatory.
Procedures are made for the installation of the REPAC system, including a SAT. However, the software is already implemented in the system delivered by the vendor, so there were no special procedures at Ringhals concerning the software implementation.

Any maintenance activity must be in agreement with the safety technical specifications.

There was a discussion whether these specification needed to be changed due to the introduction of digital equipment. As REPAC is a one-to-one replacement of existing equipment, the same safety technical specifications as before should in general be valid. There are, however, some differences with the new equipment which is relevant. One difference is that several functions are located on the same card, so that a failure in one card may influence all these functions, and the concept of operability should be re-evaluated. Another aspect is the difference in time responses between a digital and an analogue system.

A particular aspect of relevance is when a failure occurs where one cannot identify the cause. If this failure is caused by a software fault, it is a potential common cause failure. The assumption of independence between redundant channels is therefore not necessarily valid.

The software system being brought to approval will after the final implementation at the site be handled either by technical personnel, operated by operators or probably a combination of the two. From a safety point of view, the quality of the software system is of course extremely important, but also how the users of the software system is trained to use it can have an impact on the overall safety.

As part of the project, it should therefore be developed and arranged a training programme for the personnel which will use the system, to ensure that they are able to handle the system in the best way. Proper system operation and user manuals must of course be available as an integral part of the software system.

An education and re-training program has established for different types of personnel operating and maintaining the REPAC system. An extra rack for maintenance of modules and programs has been procured. It can also be used for training and education.

11. DOCUMENTATION.

A good documentation is essential, both for the development and for the licensing of safety critical software. The total documentation must cover all aspects from requirement, through design, implementation and operation, together with a number of auxiliary topics, paying special regard to the documentation necessary for safety verification.

The maintenance and updating of the documentation is of highest importance. It is essential for the licensing of a system that the documentation correctly reflects the actual version of the system. A computer based documentation system is a useful aid in this respect, as well as in the organisation of larger projects, with participation of many persons.

All phases of the REPAC development were thoroughly documented. However, no computer based documentation system was utilised. This documentation was a basis for the review meetings between SKI and Ringhals. No particular documentation was made for the reviews, except minutes from the review meetings. Finally, however, a safety analysis report was made by Ringhals, as a basis for an approval by SKI.
12 CONCLUSIONS.

As stated in the beginning, the intention of the paper is to evaluate the experience on the practical applicability of the Guidelines gained during the review of REPAC, and not an evaluation of the REPAC system itself. Emphasis is put on how or whether the various points in the Guidelines were applicable to REPAC, and how the recommendations were fulfilled.

Most of the points were relevant. If they were not it was because of the development method (e.g. not use of formal methods) or in the structure of the SPEC200 Micro system.

Not all of the relevant points in the Guidelines were followed in the REPAC project. Some points relates to method which is not fully developed, as FMECA applied to the specification. In this connection one should mention that since REPAC is a one-to-one exchange, the safety analysis is not completely remade, but based on the existing one. For a new system, a complete safety analysis would have been made in agreement with this requirement.

Documentation of most of the points could be found in the safety analysis report. This documentation is, however, not detailed enough to be the sole basis for a safety analysis, but refers to other documents. This should in principle be sufficient, since the report is intended for SKI, who usually is neither performing nor checking the detailed analysis (this is the responsibility of the power companies). A computer based documentation system could, however, have been a useful tool in the review process.

One point we want to stress is that the Guidelines are guidelines, and not requirements. This means that SKI is free to follow its own judgement on the approval of a system, even if not all recommendations in the Guidelines are fulfilled.

Although the REPAC project concerned a real implementation of a safety critical system which required approval from SKI, it can also be considered a pilot project for the assessment of programmable safety systems with the use of the Guidelines. This has given useful experience, in particular on the practical aspects. The development of REPAC was made in cooperation between persons of different background, reactor engineers, experts on conventional electronics, software engineers, safety experts etc., which had to communicate to produce the safety report. As the Guideline was not a basis for the REPAC development, one had to adapt its recommendations to what was really done. This has given a feedback which can be used in future assessment of such systems. We also hope that this experience can be a useful contribution in the international cooperation on procedures for licensing of safety critical computer systems.

REFERENCES.


ISO 9001 "Quality System - Model for quality assurance of software in specification, design, development, test and maintenance". ISO standard 9001, 1988


DEVELOPMENT - PRODUCING A REVIEWABLE SYSTEM

SESSION 2 - AFTERNOON

DEVELOPMENT OF THE DIGITAL SAFETY RELATED SYSTEM - Mr. Hiroshi Yatabe, Hitachi Ltd., Omika Works

DEVELOPMENT OF COMPUTER-BASED PROTECTION AND MONITORING SYSTEMS FOR APPLICATION ON NPP, REACTOR APPLICATIONS AND EXPERIMENTAL STUDIES - Mrs. Janette Ann Baldwin, AEA Technology Winfrith

REVIEWABLE SOFTWARE SYSTEM OF DIGITAL BASED REACTOR PROTECTION SYSTEM FOR NEXT STAGE PWR PLANTS IN JAPAN - Mr. Toshihiro Aoyagi, Japan Atomic Power Company

CONTRIBUTION TO THE SAFETY ASSESSMENT OF I & C SOFTWARE FOR NPPS - M. J.Y. Henry, Institut de Protection et de Sûreté Nucléaire (IPSN)

TRENDS AND POSITION OF ITALIAN SAFETY BODY ON THE APPLICATION OF COMPUTER-BASED PROTECTION SYSTEMS IN NPPS - Mr. Fausto Zambardi, National Agency for Environmental Protection (ANPA)

DIGITAL INSTRUMENTATION AND CONTROL SYSTEMS IN U. S. NUCLEAR POWER PLANTS - Mr. Jared S. Wermiel, U.S. Nuclear Regulatory Commission

Development of the Digital Safety Related System

Hitachi Ltd.  Hiroshi Yatabe
Toshiba Co  Hiroshi Sakamoto

Recent advanced electronic technology has been making it possible for us to introduce digital systems aiming at higher system reliability and maintainability. The safety related system for Tokyo Electric Power Company Kashiwazaki Kariwa Unit No.6 and No.7, the first ABWR (Advanced BWR) plants in Japan, has also been digitalized based upon sufficient operational experience of non-safety digital systems.

Not only the production process, the documentation, the organization but also the hardware and software architecture selected are almost the same as those for non-safety systems. The major exception is so-called V&V. V&V has been additionally conducted conforming to the guide line for digital safety related systems, though the operational records shows that our usual development and manufacturing methods have sufficient reliability by itself. Regarding the software language we selected POL (Problem Oriented Language) because of its visual characteristic. The symbolic representation of POL makes it easier to confirm the correspondance between the software document and the Interlock Block Diagram which specifies the control function of the system, and thus makes V&V activity straightforward.
1. Subjects In Digitalizing the Safety Related System

We have applied and expanded digitalization on the step-by-step basis to the normal system for the control system at the nuclear power plant and realized digitalizing the safety related system for Tokyo Electric Power Co., Kashiwazaki Kariha Units No. 6/7, as the first ABWR plant in Japan. The problem of the system which is required of high reliability is how the requirement to the system is properly realized, with little difference between the hardware and the software. In the development of the upstream documents downstream documents (system requirement → system design specification → interlock block diagram → ECWD and logic diagram), the planning and manufacturing phase is not different from conventional ideas in respects of the process, the document system, and the work organization. About the task also, information is transmitted extremely smoothly by the engineers in the nuclear power engineering department.

A great difference between the digital system and the conventional hardwired system is a software design and manufacturing work at the manufacturing stage. In recent years the productivity, reliability and maintainability of the software are being discussed. Especially in applying the digital controller to the safety related system, high reliability relative to the software is required, so how to prevent inconsistency when developing to the program from the interlock block diagram is the most important subject. Accordingly the software itself must be easily understood by anyone other than designers.

For this subject solved problem of the visibility the software by using a symbolized language called Problem Oriented Language (hereafter referred to as POL)

The following chapters deal with the POL which is the software language we are using, the V&V (Verification & Validation) in accordance with the guide line (JEAG4609) provided in introducing digitalization to the safety related system and the concrete system design/manufacturing procedure.
2. **Problem Oriented Language (POL)**

Following are the features of the interlock of the safety protection system:

- Composed of simple combinations of logics such as AND, OR etc.
- Repetitive processing
- No interruptive processing
- Describable with some tens' symbols

(Detail is shown in Table 1)

From above features, it can be understood that the application of the symbolized language is extremely valid.

Next, an example of the POL (Problem Oriented Language) we are using is shown in Fig. 1. In the POL all instructions and signals are expressed in terms of function blocks and lines. For example, the instruction AND, whose signal is outputted when all conditions of input 1, input 2, and input 3 (NOT) are established, is expressed in terms of a function block as $\frac{1}{2} \cdot \left[ \begin{array}{c} A \\ N3 \end{array} \right]$. Its clarity can be understood when compared with the relay sequence corresponding to the instruction in Figure 1.

The software prepared by such a simple functional block as this can be easily reviewed by the third party other than designers.

The reliability of the POL itself is certified by fully performing the shop test on the functional block of the simple structure. Further application of the POL is carried out step by step in the process of the nuclear power, the waste processing system normal use digital system and safety related system. And during the period full plant operating hours (actual results of operating hours) are accumulated to confirm the reliability. For the operating time of the POL, we have had experiences of the operating the plant for 15 years and no trouble due to the POL itself has occurred thereafter.
3. JEAG4609 and V&V (Verification and Validation)

Next we will explain the technology called V&V applied in order to verify and validate the software in manufacturing the digital type safety related system for Tokyo Electric Power Co., Kashiwazaki Kariha Units No.6/7.

A standard in applying the digital controller to the safety related system was issued as ANSI/IEEE 7.4.3.2 in 1982 and subsequently RG-1, 152 was established in 1985. In our country also JEAG4609 which describes basically the same technology as ANSI/IEEE-7.4.3.2, the V&V (verification and validation) technologies was clarified. The following shows the outline of JEAG4609.

JEAG4609

- Clarify design and manufacturing process
- Ensure traceability of design and manufacturing of the software
- The third party who does not carry out design / manufacture verify conformance between any adjacent two phases of the design and manufacturing process.
- Document the results.

In accordance with this guide line, we verified each step of design(veri 1-5) and the validated manufacture as shown in Fig. 4.

Design manufacture are classified into the following five steps:

1. Standard and guide line
2. System requirement (system design specification)
3. H/W, S/W design specification (Equipment design specification, IBD (Interlock Block Diagram))
4. Software design / manufacturing (logic diagram)
5. H/W, S/W integration

For the above steps, the following verification shall be performed between two adjacent steps.

1. Verification of the system design specification.

Verify that the system design specification is specified to realize the safety design
requirements correctly as the total system.

2. Verification of the hardware / software design specification.
   Verify that the hardware / software design specification is specified correctly to realize the system requirement.

3. Verification of the design of the software.
   Verify that the design of the software is designed to realize the software design specification.

4. Verification of the manufacture of the software.
   Verify that algorithm specified by the software design is properly realized by the software language.

5. Verification of the integration of the hardware / software.
   It shall be verified that the signal take-in of the hardware / software agree in the form of the total system after the hardware / software integration.

The above is the contents of the V&V we performed and the results thereof shall be kept as documents respectively.
4. Manufacturing Procedure of the Software System and the Implementing Procedure of V&V

Finally we will introduce the procedure of the manufacturing and review we are performing, citing on the digital type safety related system for Tokyo Electric Power Co., Kasiwazaki kariha Units No. 6/7.

First stage
Requirements for the system are developed in the system design requirement specification (each system design specification). (Check & review) and Veri 1.

Second stage
The system requirement specifications are developed H/W, S/W design requirement specification (equipment design specification, interlock block diagram). (Check & review) and Veri 2.

Third / fourth stage
H/W, S/W design requirement specifications are developed on the developed on the hardware design (ECWD) and the software design (software drawing).

For the hardware, we purchase materials and parts in accordance with this ECWD manufacture and assemble them. All the parts, materials and their connections are expressed in the ECWD. (Check & review).

On the other hand, the softwares are expressed by the POL in accordance with the IBD, needless to say, all the instructions and connections are expressed on the drawing. (Check & review) and Veri. 3/4.

* (V&V in the design and manufacture stage can be performed very smoothly because the POL is used).

Fifth stage
Integration of the hardware and the software (completion of panel manufacture).

Veri 5.
Shop test: For the control panel in which the hardware/software are integrated in the above process, we conduct the following test based on the shop test procedure (shop test witness, inspection record) and validation.

General structural inspection, wiring inspection, sequence test, performance/characteristic test, instrument test.

Verifications of relays, etc., insulation resistance test, withstand voltage test, antiseismic verification test, and combination tests.

Semi-dynamic test: We consider that the quality can be fully verified with the normal quality assurance process as shown by the above steps. However because of the first application of the digitalization to the safety protection system, we conducted the semi-dynamic test using a simple simulator to verify the behavior of the control panel for the important event of the plant (LOCA etc.) as an referential additional test. Fig. 5 shows an example of this test.

Packing/Shipment: Packing/Shipment are conducted in accordance with the packing/Shipment procedure.

Site installation: Installation to the site are conducted in accordance with the prespecified procedure.

General structural inspection, wiring inspection, sequence inspection, performance/characteristic test, instrument test, insulation resistance test.

System preoperation test:

The test shall be conducted in accordance with the prespecified procedure.

Interlock test, actuation test, safety protection test, Combination test (cabinets, sensors, actuators).
Heat Up test at the site:

The tests shall be conducted in accordance with the predetermined procedure.

Thus, in applying the digital technology to the safety related system for Tokyo Electric Power Co., Kashiwazaki Kariha Units No. 6/7, we introduced processes to manufacture the software system from the system requirement to the control system.

Conclusion

By applying the digital system step-by-step to atomic power and using the POL, we have already implemented check and review in each stage of design and manufacture, with satisfactory actual results in operating.

Further in applying the digital system to the safety protection system for K-6/7 this time, we have manufactured a system reviewed by the V&V procedure.
### Example of POL Instructions

<table>
<thead>
<tr>
<th>Instruction</th>
<th>Relay Sequence</th>
<th>Function Block</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>A (AND)</td>
<td><img src="image" alt="A Relay" /></td>
<td><img src="image" alt="A Function Block" /></td>
<td>Logical product (AND) of all inputs</td>
</tr>
<tr>
<td>O (OR)</td>
<td><img src="image" alt="O Relay" /></td>
<td><img src="image" alt="O Function Block" /></td>
<td>Logical sum (OR) of all inputs</td>
</tr>
<tr>
<td>TON (Timer delay on)</td>
<td><img src="image" alt="TON Relay" /></td>
<td><img src="image" alt="TON Function Block" /></td>
<td>Turns on after time T.</td>
</tr>
<tr>
<td>OUT (Device output)</td>
<td><img src="image" alt="OUT Relay" /></td>
<td><img src="image" alt="OUT Function Block" /></td>
<td>Output according to input condition. Output to multiple device is possible.</td>
</tr>
<tr>
<td>LH (Latch output)</td>
<td><img src="image" alt="LH Relay" /></td>
<td><img src="image" alt="LH Function Block" /></td>
<td>Same as OUT except that the previous status is restored after power recovery.</td>
</tr>
<tr>
<td>O/A (OR—AND)</td>
<td><img src="image" alt="O/A Relay" /></td>
<td><img src="image" alt="O/A Function Block" /></td>
<td>After OR processing of an input. AND processing of its result and the previous logic result is executed.</td>
</tr>
<tr>
<td>A/O (AND—OR)</td>
<td><img src="image" alt="A/O Relay" /></td>
<td><img src="image" alt="A/O Function Block" /></td>
<td>After AND processing of an input, OR processing of its result and the previous logic result is executed.</td>
</tr>
</tbody>
</table>

Other easy-to-use instructions are also available.
<table>
<thead>
<tr>
<th>No</th>
<th>ITEM</th>
<th>Characteristics</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Expressibility in IBD with relatively small number of element kinds</td>
<td>Expressible</td>
</tr>
<tr>
<td>2</td>
<td>Weight on simple logic decisions or complex calculations</td>
<td>Simple</td>
</tr>
<tr>
<td>3</td>
<td>Event-driven priority structure</td>
<td>Fixed period based</td>
</tr>
<tr>
<td>4</td>
<td>Volume of input variables for SSLC initiation signal</td>
<td>About 60</td>
</tr>
<tr>
<td>5</td>
<td>Kinds of output variables</td>
<td>Mainly ON/OFF</td>
</tr>
<tr>
<td>6</td>
<td>Language used</td>
<td>Mainly symbolic language</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Symbol kinds :about 20</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Number of symbols:several</td>
</tr>
<tr>
<td></td>
<td></td>
<td>hundred</td>
</tr>
</tbody>
</table>

TEPCO, HITACHI, TOSHIBA
Difference between the Digital and Conventional Systems

H/W,S/W Design Specifications

- Equipment Design Spec.
- Interlock Block Diagrams

H/W Design

ECWD

S/W Design

S/W Diagrams

S/W Coding

Interlock Block Diagram

1

2

3

5

4

6

Conventional System

(Relay Sequence)

Digital System

(Logic Sequence)

1

2

N3

3

N4

4

5

R

6

TEPCO, HITACHI, TOSHIBA
Fig. 3  Steps of Design/Manufacture and Verification & Validation

System Design Requirement Specifications
(Functional Design Specifications)

Hardware and Software Design Requirement Specifications
(Block Diagram)

Software Design
(Algorithm and I/O Assignment)

Software Manufacture

Hardware/Software Integration
(Software Loading into Hardware)

Validation Test

Verification 1

System Design
(same as for conventional systems)

Verification 2

Verification 3

Verification 4

Verification 5

Validation

TEPCO, HITACHI, TOSHIBA
The specific procedures for Digital Safety Protection System

<table>
<thead>
<tr>
<th>Utility</th>
<th>Manufacturer</th>
<th>Design &amp; Manuf.</th>
<th>V &amp; V</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Check &amp; Review</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Check &amp; Review</td>
<td></td>
<td></td>
<td>V &amp; V</td>
</tr>
<tr>
<td>Veri-1 Report</td>
<td></td>
<td></td>
<td>master plan</td>
</tr>
<tr>
<td>Check &amp; Review</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Veri-2 Report</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **System Requirements**
- **System Design Specifications**
- **H/W,S/W Design Specifications**
  - Equipment Design Spec.
  - Interlock Block Diagrams
The specific procedures for Digital Safety Protection System

<table>
<thead>
<tr>
<th>Utility</th>
<th>Manufacturer</th>
<th>V &amp; V</th>
</tr>
</thead>
<tbody>
<tr>
<td>Design &amp; Manuf.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **H/W, S/W Design Specifications**
  - Equipment Design Spec.
  - Interlock Block Diagrams

  - **H/W Design**
    - ECWD

  - **Components Procurement**

  - **Fabrication**

  - **S/W Design**
    - S/W Diagrams

  - **S/W Coding**

  - **Integration of H/W & S/W**
    - (manufacturing completed)

- **Check & Review**

- **Veri-3/4 Report**
The specific procedures for Digital Safety Protection System

Utility

<table>
<thead>
<tr>
<th>Veri-5 Report</th>
<th>Check&amp;Review</th>
</tr>
</thead>
<tbody>
<tr>
<td>Witness Test at Factory</td>
<td>Test Data</td>
</tr>
</tbody>
</table>

Manufacturer

Design & Manuf.

Integration of H/W & S/W
(manufacturing completed)

Factory Test Procedures

- Visual Inspection
- Wiring Inspection
- I/O Characteristic Test
- Sequence Test
- Instruments Test
- $M\Omega$ Test
- Withstand Voltage Test
- Aseismatic Test
- Combination Test

Semi-Dynamic Simulation Test

Shipping Procedure

Packing Check/Delivery Check

V & V

Veri-5

Validation

TEPCO, HITACHI, TOSHIBA
The specific procedures for Digital Safety Protection System

<table>
<thead>
<tr>
<th>Utility</th>
<th>Manufacturer</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Shipping Procedure</td>
</tr>
<tr>
<td></td>
<td>Packing Check/Delivery Check</td>
</tr>
<tr>
<td></td>
<td>Installation Procedure</td>
</tr>
<tr>
<td></td>
<td>Installation at Site</td>
</tr>
<tr>
<td></td>
<td>Cabinet Test Procedures</td>
</tr>
<tr>
<td></td>
<td>Cabinet Test</td>
</tr>
<tr>
<td></td>
<td>- Reassemble Check</td>
</tr>
<tr>
<td></td>
<td>- Transmission Test</td>
</tr>
<tr>
<td></td>
<td>- Sequence Test</td>
</tr>
<tr>
<td></td>
<td>a. Algorithm</td>
</tr>
<tr>
<td></td>
<td>b. Alarm</td>
</tr>
<tr>
<td></td>
<td>c. Interface with other systems</td>
</tr>
<tr>
<td></td>
<td>d. Combination Test with the field instruments</td>
</tr>
<tr>
<td></td>
<td>e. D-I/O Check</td>
</tr>
<tr>
<td></td>
<td>- A - I/O Check (including Accuracy Test)</td>
</tr>
<tr>
<td></td>
<td>- Combination with actuators</td>
</tr>
<tr>
<td></td>
<td>- Integrated Test</td>
</tr>
</tbody>
</table>

Check & Review - Witness - Test Report - Check & Review - Witness - Test Data
The specific procedures for Digital Safety Protection System

- **Utility**
  - Check & Review
  - Witness
  - Test Report

- **Manufacturer**
  - Cabinet Test Procedures
  - System Preoperation Test Procedure
    - System Preoperation Test
      - Interlock Test (including interface with the other systems)
      - Actuator Test
      - Safety Protection Test
      - Combination Test (Cabinets, Sensors, Actuators)
  - Heat up Procedure
  - Heat up Test
Semi-Dynamic Simulation Test (1/2)

From the viewpoint that this system is the first digital Reactor Protection System, we confirm the validity of the system by simulating the changes of the process values, which is some of the analysis conditions required.

- Prepare the simulator which simulates the changes of the parameters used in the analysis conditions (LOCA and so on).

- Input the signals from the simulator to the digital controller, and record the corresponding behavior by the recorders.

- Verify if the system works as expected.
The simulation test cases:

- **Loss of Power Supply**: The plant AC power is lost because of the failure of the electricity feeding system device in the plant, etc.

- **Main Steam Isolation Valve Close**: When MSIVs are closed, the reactor pressure becomes high.

- **Failure of Pressure Controlling Device**: The reactor pressure level becomes abnormal in case of a failure of Pressure Controlling Device.

- **Loss of all the feed water flow**: The reactor water level becomes low in case of a failure of the feed water system.

- **Loss of the Coolant Water**: The pipes connected to the RPV break and the coolant flows out of the reactor and reactor water level becomes low.

- **Break of the Main Steam Pipe**: When the main steam pipe breaks, the coolant flows out of the reactor and reactor water level becomes low.

- **Fuel Treatment Accident**: When some fuel being exchanged is dropped, the fuel may break.
Configuration of the Semi-Dynamic Test

Div. I TLU
Div. I DTM

Div. I RPS/MSIV DTM RMU
Div. I ESF DTM RMU

N36-PT-026A B21-PT-025E
B21-PT-025A B21-LT-003A
E31-DPT-002A

CV Fast closure
Radiation rate of MS is high

Same as Div. I
Same as Div. I
Same as Div. I

Simulator

TEPCO, HITACHI, TOSHIBA
An Example of Semi-Dynamic Simulation Test
Requirements for Computer Systems Important to Safety at Forsmark NPP

Olle Andersson
Forsmarks Kraftgrupp AB
S-742 03 ÖSTHAMMAR
SWEDEN

ABSTRACT

Ageing, the lifetime of components and systems are issues which are given considerable attention within Forsmarks Kraftgrupp AB. Sustainability, that is, long-term maintenance of production capabilities, is a part of our business definition. In order to fulfil the requirement on sustainability, and to ensure that we are well equipped for the future, a modernization programme has been initiated and is expected to be completed in the year 2000. This programme entails an investment of around USD 300 million. The modernization of Instrument and Control equipment (I&C) accounts for about 30% of these investments.

Forsmarks Kraftgrupp AB generates electricity in ABB Atom-design boiling water reactors. Forsmark 1 and 2 were taken into commercial operation in 1980 and 1981. Forsmark 3, which belongs to a later generation, was taken into operation in 1985. The instrument, control and safety systems which were originally installed in the units were based on the tried and tested conventional technology of that time. In Forsmark 1 and 2, relay-based systems were used in combination with analogue technology. In Forsmark 3, conventional analogue and digital electronics were installed in combination with relay technology. The modernization projects which are now in progress, mainly apply to Forsmark 1 and 2. These units contain "old" technology and have control rooms which do not fulfil modern requirements with regard to ergonomics.

Digital, computerized technology will largely be used for the replacement and modernization of the existing I&C systems, which may also affect safety-related systems. To ensure that the units can be operated with the same high level of safety and availability as at present, the implementation of new technology is based on the following three basic principles:

- A structured approach must be applied which means stringent quality requirements during the design, manufacturing and implementation phases. International experience must be used and established standards must be applied.

- Equipment must be designed in accordance with well-established ergonomic requirements. The equipment must be functionally adapted to the operators' needs.

- Tried and tested, standardized equipment must be used. Customized solutions must be avoided.
INTRODUCTION

The currently planned stage of the upgrading of I&C systems at Forsmark 1 and 2 does not include the reactor protection systems. The replacement of the unit process computers, replacement and modernization of control rod manoeuvre and control rod indication systems as well as the modernization of alarm and signal monitoring systems are included in the first stage. Even if the reactor protection system is not included in the first stage, the upgrading of I&C systems means that the control room design will be affected to such an extent that it will be treated as a modification of safety-related equipment. The basic principles for how to implement the modernization projects have been established in the light of this. A description of the approach to be adopted has been submitted to the regulatory authorities. The approach is based on the following three basic principles:

- Structured approach.
- Control room design in accordance with established ergonomic requirements.
- The technical solution must be based on tried and tested, standardized technology.

The aim of the three basic principles is to ensure that, after the modernization programme has been completed, the same currently high level of safety and availability is attained at the units concerned.

STRUCTURED APPROACH

A structured approach must be applied to the planning, design and implementation of new I&C. A structured approach means that:

- Basic and specific conditions are established and documented.
- Established standards are used for the design and implementation of computerized functions.
- The control room function must be analyzed and documented. Task analyses must also be carried out with the aim of establishing the operators' requirements on the equipment.

The structured approach is illustrated in Appendix 1.

Establishment of Requirements

Basic Requirements on Standards

New equipment must fulfil at least the same requirement level as that which applies to the equipment which was originally installed. Basic safety principles which are already being applied must also be applied when new technology is introduced. Applied basic safety principles comprise: the single-failure criterion, the 30-minute rule, redundancy and diversification etc.

The requirements on electrical systems are function-oriented and the criteria used concern: redundancy, availability, control, single-failure criterion, independence, testing and separation. The classification of electrical systems is based on IEEE 308, including I&C functions. However, the classification has been developed by the original suppliers of the unit and, in
practice, follows the IEC 1226 classification. The classification principles will not change as a result of the introduction of new control equipment.

Diversified solutions may be necessary for safety functions.

**Special Requirements on Computerized Functions**

In addition to the general and basic requirements as described above, specific requirements apply with regard to computerized functions. The aim of the special requirements is to ensure high quality and to minimize the risk of CCF/CMF (Common Cause Failure and Common Mode Failure).

The following requirements have been established:

- Category A Systems (1E systems) must be developed and designed in accordance with the methods specified in IEC 880.
- For Category B and C Systems (2E and 3E applications), quality control standards must apply in conformance with ISO 9000-3.
- The systems must be load-independent.
- The systems must be fail-safe.
- Computer programs and configured data must be stored in permanent memory.

Established programmes for the verification and validation of equipment and computer programs must be developed in advance.

**Seismic Design Requirements**

Using realistic methods, all plant modifications which are introduced since a few years back, must be designed to withstand seismic events, provided that the equipment is required for the safe shutdown of the reactor or for maintaining the containment function. Depending on the safety classification of the control rod manoeuvre system and the system for control rod indication, the seismic requirement may have to be applied to these systems.

**General Design Bases**

In connection with the original design of the Forsmark reactors, the supplier established system-specific, general design bases. The document provides detailed information on the design of, for example, cables, power supply systems and alarm systems.

When new technology is introduced, these documents will have to be revised and supplemented. In the first instance, the design bases which are directly dependent on the current stage are revised (or re-written). In some cases, completely new design bases must be prepared.

The basic principles for separation, power supply, cables, failure signals etc. are not affected by the introduction of new technology.
Control Room Philosophy and Evaluation of the Control Room Function

An important stage in the preparations for the introduction of new control room equipment has been to document requirements and conditions for control room work. Extensive work has been carried out for Forsmark 1 and 2. This has resulted in two documents entitled “Control Room Philosophies”.

The preparation of these documents marks the introduction of an overall approach to the control room function. The philosophies provide guidelines for control room design both with regard to man-machine interface and support functions such as operating procedures, training, communication equipment. Various work stages, tasks and support functions have been analyzed. Requirements on control room design have then been made, based on the analysis results.

In order to further establish the requirements on control room design and on I&C systems, tasks have been analyzed. Current working methods are documented using the task analyses. The modifications which are necessary to improve the control room design and operator support must be based on the results of the analyses. As far as possible, the improvement proposals must be verified with new mock-up analyses before the proposals are taken to the design stage.

CONTROL ROOM DESIGN IN ACCORDANCE WITH ESTABLISHED ERGONOMIC REQUIREMENTS

In Sweden, there is no overall standard specifically for ergonomic nuclear reactor control room design. In the light of the current upgrading projects which, in principle, imply a total modernization of the control room, the need for such an overall document has been emphasized.

Man-machine interface at Forsmark must be based on a systematic approach and implemented in accordance with good ergonomic practice. In the light of this, an ergonomic handbook has been prepared. The handbook is based on NUREG 5908 “Advanced Human System Interface Design Review Guidelines.”

Since NUREG 5908 was written as an evaluation document and the guidelines prepared by Forsmarks Kraftgrupp AB are primarily requirements which can be used by purchasers of equipment, designers and evaluators, the structure of the documents are completely different. In order to evaluate whether important requirements and stages, in spite of differences, are included in the Swedish document, an independent comparative evaluation with the NUREG guide has been carried out by ABB Atom.

Handling of Ergonomic Issues in the Modification Process

The ergonomic handbook fulfills an additional function, in addition to that of establishing requirements, namely, that of serving as a “textbook” for how ergonomic issues are to be resolved in an optimal way in complex modification projects.
In the light of this, the ergonomics handbook contains “10 golden rules” for dealing with ergonomic issues:

**The Control Room in an Integrated Perspective**

Man-machine interface is only one of several support functions for the control room operators. All support functions must be harmonized with each other.

**Control Room Philosophy**

All modifications must be viewed in a larger perspective. A plan must be prepared for the long-term modification of the control room and all modifications must be considered in relation to this plan.

**Functions and Concrete Solutions**

A function analysis must be prepared and the question of exactly what information the operator needs and what he/she must control in order to attain the goals of the system must be answered. A clear definition must be given of what the operator must do and of what the technology must do. The administrative functions and barriers which must exist must be analyzed.

**Task Analysis**

A detailed description of what the operator does in order to complete his/her tasks during different operating conditions must be provided. How different solutions support the operator in his/her task must be analyzed. Task analyses must be carried out for existing systems as well as in connection with the preparation of alternative solutions.

**Ergonomic Expertise**

Specialists should be used when necessary, for example, lighting specialists, computer interface specialists etc.

**Experience Feedback**

How similar problems have been solved by others and the experience that has been gained should be established.

**Operator Involvement**

The operators’ views on the control room layout should be actively monitored. The wishes of everyone cannot be satisfied. However, everyone must be able to state their point of view and it is, after all, the operator who has the greatest experience of working in the control room.
Qualitative Risk Analyses

For each modification proposal, the possibility of making errors must be discussed. This must be achieved systematically and the barriers which exist against human error must be investigated.

Uniformity of Design

How the modification conforms to current design room principles must be established and any conflicts resolved. The control room must be designed in a uniform manner so that the same principles apply at all points.

Simplicity, Structure and Logical Design

It must be obvious and clear to establish the relationship between items in the control room as well as the function of different items. Items must be marked in a clear and logical manner. Instruments must be large enough to be easily identifiable. The risk of confusion must be investigated.

TRIED AND TESTED STANDARDIZED TECHNOLOGY

In connection with the initiation of the modernization projects, basic technical requirements for the I&C functions were established.

In addition to the basic requirements previously described, the following technical requirements were specified:

- Simple operator adjustment. The man-machine interface design must be uniform.

- The I&C functions must be available in the form of standard industrial equipment. Customized solutions must be avoided. This means that one supplier must be appointed for all I&C functions.

- Open system solutions must be applied for "specialist functions" such as core calculations, vibration monitoring etc. However, the operators' MM interface must be uniform for general functions as well as specialist functions. This means that the MM interface must be able to communicate with general equipment as well as with dedicated systems.

- To prevent undue impact on sensitive I&C functions, communication between shared Forsmark computer systems and the control room systems must be carried out via special gateways acting as "check-valves".

- A signal processing level must be established which is built up by automation processors and serial networks. On the signal processing level, signal exchange and required signal processing must be handled.

- It must be possible to adapt new equipment to existing processes with regard to analogue and digital incoming and outgoing signals.
Selected structures must allow for a gradual modernization and the detailed solutions must be standardized in order to simplify design, operation and maintenance as well as to limit the risk for CCF. The general requirements have been discussed with different suppliers which, on behalf of Forsmarks Kraftgrupp AB, have carried out structural investigations to establish how requirements can be implemented. On the basis of the investigations carried out, an overall agreement has been signed with a supplier.

The principles of the new I&C functions are presented in Appendix 2.
APPENDIX 1

IMPORTANT STAGES AND ELEMENTS IN THE MODIFICATION PROCESS

 Conditions & Goal descriptions

 Function analysis

 Task analysis for existing system

 Technical functional requirements

 Ergonomic functional requirements

 Proposed solution

 Task and consequence analysis

 Verification and validation
APPENDIX 2

NEW STRUCTURE OF I&C FUNCTIONS AT FORSMARK 1 AND 2
Development of computer-based protection and monitoring systems for application on nuclear power plant, reactor applications and experimental studies.

Wall D N, Baldwin J A, Grayson A R, Adams G N.
AEA Technology
Winfirth Technology Centre
Dorchester
United Kingdom

Introduction.

This paper is divided into two parts, the first part describes the development of and issues associated with the introduction of the Single Channel Trip System (SCTS), the first computer based protection system licensed for use on a civil UK reactor. The second part of the paper discusses some of the activities relating to the issues arising from the deployment of computer based systems, and in particular the DARTS (Demonstration of Advanced Reliability Techniques) experiment. The two pieces of work were completed in parallel and the very different nature of the developments prevented any exchange between them. However the conclusions drawn from the two exercises show a number of common features.

The information available from the SCTS development differs markedly from that coming from the DARTS experiment primarily because DARTS as an experiment, was focussed on collecting the necessary information to allow a comparative analysis to be completed, however, similarities in the grade of software developed, i.e. safety critical, allow some comparisons to be made. Two aspects of the SCTS development are considered, first the development of the SCTS safety critical software that had a well defined requirements specification and secondly the monitoring system that accompanied it and had a considerably less well defined requirements specification which evolved during the course of the development on the basis of customer assessment of a series of prototypes. The DARTS development, while being similar in nature to the Safety Critical SCTS development, has provided many more productivity and reliability metrics as a result of the fact that it was a research project. The paper considers the various approaches adopted for the development and justification of these programmes, highlighting in particular the difficulties caused by lack of clarity in the software requirements.

SCTS Development.

The potential for gag failure, which would restrict coolant gas flow and reduce channel cooling, resulted in significant limitations on power generation of the Dungeness B AGR Nuclear Electric decided to provide defence against the consequences of a gag fault rather than modify the reactor. The basic system requirement was for a simple high temperature trip on each of the 408 reactor fuel channels.

This type of trip function has traditionally been provided using analogue technology. This was not viable in this case because the large number of fuel channels and the need for multiple protection channels on each fuel channel renders the number of instruments excessive. The solution selected was to use a computer based protection system, SCTS, based on AEA Technology's Inherently Safe Automatic Trip (ISAT™) architecture to trip the reactor on
detection of the high coolant gas outlet temperature that would arise in the event of gag failure. The system consisted, by necessity, of four protection channels working on a two out of four vote. There are only two thermocouples available in each channel, so the more conventional two out of three vote is not possible.

The system was developed to meet a functional specification supplied by Nuclear Electric. The required functionality was simple and well defined, and consequently the specification was quite straightforward. It is noted that at no point were the safety issues conveyed to the designers nor was any attempt made to set the system requirements down in a formal manner, reliance instead being placed on the natural language specification.

The system software was produced to meet the specification by modifying an existing ISAT™ for high coolant temperature protection that had been developed and tested on the prototype fast reactor at Dounreay. This system had a pedigree extending back over 10 years with no software errors reported. The software was executed on commercially available processors and bus.

The ISAT™ architecture is designed such that all parts that make up the system are checked continuously throughout use. There is no collusion between disparate parts of the system and all parts process data independently. The data for an ISAT™ system is so arranged with interleaved test signals that all paths through the software are exercised each time a frame of data is processed. A pattern of pass and fail results are produced from the data resulting in a pattern that may be checked easily using simple and proven hardware.

The software, written in assembly language, was required to be developed in accordance with IEC 880. The work was performed by completing several iterations of a modification process derived from the model recommended for adaptive software maintenance by EWICS TC7. The software changes were performed in several (=7) planned iterations with manual development and verification at each stage. Implementation of the adaptive maintenance lifecycle ensured visibility of the software production process and aided greatly in the planning and resource control. All outputs from the process were brought under an electronic configuration system which provided version and access control. Each of the planned iterations in the development process can be described by the following activities:

1. Modification request by customer to developer via development manager.
2. Feasibility study carried out by developer.
3. Modification authorised by development
5. Modified of Design plus independent verification.
7. Modified of Test plan plus independent verification.
8. Test Report plus independent verification.

The system justification was based on three elements:

- the quality and rigour of the production process;
- analysis including a hardware failure modes and effects analysis and appeals to the self testing properties of the ISAT™ architecture;
- extensive testing.
In addition to local independent verification, developers provided the customer with all the necessary information concerning the system, its properties, safety features etc., in the form of specifications, drawings, summaries etc. These details were also passed to an external assessor for the purpose of quantitative reliability assessment and an qualitative software assessment. This information was processed internally by the customer to provide information for incorporation into safety submissions to NII.

It should be noted that the customer was required to submit the safety case for the system. The developer was required to submit details of the system to the customer for onward passage after editing to the regulator. This indirect interaction led in some cases to misunderstandings and additional work.

For the purpose of justifying the system to the regulators (NII), the onus was on proving that the software would not prejudice the reliability of the hardware, already assessed by traditional methods, including FMEA and fault tree analysis. In many cases the software also checks the operation of the hardware to ensure correct operation, and although this is not accounted for in the calculation of reliability, it does provide confidence in the system. No criteria were defined by the regulator to aid the developer, and no feedback occurred via the customer until the system was nearly completed. Obviously the risk in not meeting the criteria (assumed by the developer) was not assessable. Adverse feedback towards the end of the development could have resulted in serious delays and rework.

Again to provide confidence that no errors are present in the code and in light of the precedent-setting nature of SCTS two independent reviews of the system were conducted, one concentrated on the quality of the development process, and particularly that of the software, the second on the hardware and fault revealing properties of the system. The substantiation of the system was completed by a MALPAS analysis coupled with a source code comparison exercise. These last two processes are a feature common to the licensing of both the SCTS and Sizewell B nuclear reactor protection systems.

Monitoring the Status and Performance of the SCTS

The ISAT™ system performs the trip functions, but there is an additional requirement for system equipment monitoring and capture of the data relating to events associated with a demand on the system. These functions are performed using an external PC based monitor. The SCTS computers pass information via a single direction optical link to a buffer computer which forms the data into packets and broadcasts it on an Ethernet. The monitor computer captures the data and processes it to provide information on the equipment status. The main functions of the computer are to:

- indicate the status of all parts of the system in real time;
- display the temperature signals in real time;
- perform consistency and diagnostic checks on the data, particularly the thermocouples;
- provide analysis of trips;
- capture the thermocouple data in the period immediately before and during a system trip, to provide data for a trip analysis.

The form of the system is quite different from the main safety critical system, in that it is written in Pascal and uses the DOS operating system. The system is considered to be of lower
grade than the main SCTS system. The requirements and specification were not firmly established and additional functionality was requested during the course of the development. These changes were generally captured in conjunction with the customer and documented by the developers rather than being provided by those who produced the requirements specification.

In addition to the functional requirements, there were extensive non functional requirements, particularly relating to the nature of the display screens, that were not specified as part of the requirements but were produced in what was essentially a prototyping exercise for approval by the customer. This exercise required the assistance of those that would ultimately use the system. However, some decisions on the presentation of data to the operator were decided without consultation with the operator but handled by a representative from management. This did result in incorrect information being passed to the developer, as during trials of the system objections were raised by the operator resulting in extra development time being required.

The STARTS development model was followed with manual verification of the level 1 and 2 specifications, and the test plan documents, but not of the code itself. The production of code was carried through a tool (PDF) that generated a structure for the code from design input via Jackson diagrams. The validation and justification relied heavily on the testing of the system. Testing was carried out on successive builds of the software, each build incorporating increased functionality of the software. The tests were derived from the system specification and the structure of the code. They provided statement and branch coverage of the software and while extensive, they were by no means exhaustive.

The problems presented by this development were quite different from those of the main system and included:

- the significant number of important non functional requirements such as screen formats;
- the operation of the monitor hardware close to its performance limits;
- the relative informality with which the requirements were expressed.

The value of the monitor became readily apparent once the system was put into service and the availability of the data gave the potential for extensive analysis of not only the SCTS system equipment but also of the thermocouple and reactor performance.

The build and operation of the monitoring systems were very different from the processes followed for the safety system. The initial lack of clear an unambiguous requirements and the use of iterative customer approval of the system specification resulted in extensive rework. This was mainly due to the fact that the customer was unaware as to the quality and amount of information that could be deduced, both concerning the equipment and the signal data. This was in part also due to the fact that the safety system itself was a modification carried out on a mature system where considerable data and information were available, whereas the monitoring system was entirely new. As the customer was not completely aware of the potential of the monitoring system, the requirements could not be formulated without education of the customer and explanation as to what would or could be useful and what was available in terms of information from the system.

The capture of requirements was therefore a protated process, the problems being exacerbated by the fact that the customer had several faces in addition the end user who would
want to use the monitor to aid his work. Prototypes were used in the formation of the requirements, but as stated these prototypes were not responding to the person who in the end would make practical use of the system.

The DARTS experiment.

The ESPRIT II DARTS experiment [1] was a multinational project, part supported by the CEC that was established to investigate many of the problems encountered in the development of computer based safety critical systems. Some of the key issues and their effect on the cost of production and reliability of the product that were addressed and included:

- choice of computer language;
- choice of development methodology e.g. structured v mathematical formal;
- role of tool support;
- testing methodologies e.g. black box, white box, statistical & operational;
- assessment.

The experiment was started by producing a natural language requirements specification setting out the requirements for a four channel protection system requiring a vote of 2 out of 4 the same as for SCTS. This was given to four development teams who each produced a protection channel. The production methods adopted for the four channels were:

<table>
<thead>
<tr>
<th>Specification</th>
<th>Channel 1</th>
<th>Channel 2</th>
<th>Channel 3</th>
<th>Channel 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Design</td>
<td>VDM</td>
<td>FOREST/MAL</td>
<td>SADT/ASA</td>
<td>Yourdon(SA)</td>
</tr>
<tr>
<td>Code</td>
<td>Jackson PDF</td>
<td>FOREST/MAL</td>
<td>HOOD</td>
<td>Yourdon(SD)</td>
</tr>
<tr>
<td>Test Tools</td>
<td>Pascal</td>
<td>OCCAM</td>
<td>Ada</td>
<td>C</td>
</tr>
<tr>
<td></td>
<td>SPADE</td>
<td></td>
<td>Logiscope</td>
<td>VAX Set</td>
</tr>
<tr>
<td></td>
<td>Testbed</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The developments followed a basic waterfall lifecycle development model. This allowed comparison data to be recorded consistently. They were carried out in accordance with IEC880 as far as it was relevant. Other emerging standards were also consulted such as Interim Defence Standards 00-55 and 00-56 and IEC65A WG9 and WG10 (now IEC1508). Each channel was assessed by a single external organisation within the project and subsequently the whole system was assessed.

An extensive data collection exercise was completed during the development and assessment work. This exercise focussed on recording quantitative information, such as effort expended and the error history of the software, in a database.

Once each team was happy with the performance of their channel and had performed a series of acceptance tests, installed in a test harness for testing where each channel was tested back to back with the other channels. All outputs from each of the channels in response to a particular data message were recorded. Where a single channel disagreed with the other three, the developers of that channel investigated the problem against the requirements specification and other channel results. This usually resulted in repair work. A full set of tests consisting of tests derived from the requirements, statistical tests and actual profile data was carried out. A final set of testing was carried out and this provided the reliability figures for the software developed on each channel.
The test data sets and results were recorded in a second database. The contents of the two databases plus the results of the analysis of code provided the basis on which the first conclusions of the experiment are based.

**Effort analysis.**

The effort required to develop the channels varied significantly as did the assessment effort. The relative effort by phase for the four developments is shown in figure 1 below.

![Figure 1 Raw effort data from the DARTS experiment.](image)

The results shown in figure 1 are the raw data from the experiment. They show that the formally produced channels i.e. 1 and 2, expended more effort in the specification phase followed by a lower proportion of effort in the design phase. This effort profile is consistent with that expected for the use of a formal development method. The other noticeable feature of the results is that the specification effort for channel 2 is very large. As with many formal methods, the division between specification and design can be unclear and some of the channel 2 design effort was entered into the database as specification effort. Similarly for channel 1 validation effort was entered as implementation effort.

**Static Code Analysis.**

The DARTS project used static analysis of the code to allow comparison of the resulting complexities and certain productivity measures to be calculated. All four channels were analysed using a combination of tools, e.g. the QUALMS and PROMETRIX. The latter is backwards compatible with QUALMS and can be used to render the static measures compatible. The four metrics used were:
- *Nodes* is the number of executable statements in the program.
- *Compressed Nodes* is the number of nodes remaining when all sequences of nodes which have no branching in or out are modelled as a single node.
- *McCabe* is the complexity metric calculated for the whole channel.
- *Albrecht's EFP* (displayed as a line graph against the right-hand axis) was not measured directly by Prometrix, but calculated from the Prometrix measurement of *Nodes*, as described below.

The size and complexity values obtained for the channels are shown below.

![Figure 2 Static measures for the Channels](image)

The number of compressed nodes divided by the number of nodes gives the proportion of statements in each channel which involve branching in or out.

<table>
<thead>
<tr>
<th>Comp. Nodes/Nodes</th>
<th>CH1</th>
<th>CH2</th>
<th>CH3</th>
<th>CH4</th>
</tr>
</thead>
<tbody>
<tr>
<td>48%</td>
<td>67%</td>
<td>75%</td>
<td>77%</td>
<td></td>
</tr>
</tbody>
</table>

The lower ratio for Channel 1 shows that more than 50% of the code consists of simple, linear sequences of statements with no branching in or out.
The McCabe metric gives the number of out-branching statements in the code, which is effectively a count of the number of decision statements such as IF which may be disguised as CASE or other statements. The table below gives the number of decision statements per 100 lines of code and gives an indication of the programming 'style' with respect to the density of 'IF' statements.

<table>
<thead>
<tr>
<th>McCabe/hundreds_of_nodes</th>
<th>CH1</th>
<th>CH2</th>
<th>CH3</th>
<th>CH4</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>18</td>
<td>31</td>
<td>36</td>
<td>37</td>
</tr>
</tbody>
</table>

The figures for Channels 2, 3 and 4 are quite similar. The lower figure for Channel 1 again shows that the code 'style' is simpler with a lower density of decision points. This disguises the fact that Channel 1 actually has more decision points than Channels 2 & 4.

Different programming provide different productivities and reliabilities during the implementation phase. The source code size metrics were used with the declared effort to investigate this. The number of source statements per function point was assumed to be as follows:

- Pascal 91
- Occam 91 estimate
- ADA 71 estimate
- C 150

The number of 'Equivalent Function Points' has been calculated for each channel by dividing the 'number of nodes' by the language dependent 'expressiveness' number. The results are shown in figure 2. The numbers of Equivalent Function Points, which we would expect to be reasonably similar for each channel, show a range from 19, Channel 1, to 3.24, Channel 4, and seem to follow the other size metrics rather than factor them out. The two extreme values, which do not involve 'estimates', actually have a larger ratio between them than the relative size of these channels 1766 to 486. This large difference would seem to indicate that the attempt to normalise size with respect to source language is not very valid for the DARTS channels. However viewing the results very sceptically, Channel 1 is a lengthy solution, Channel 3 is a simple and long, rather than compact and 'clever' solution. On this basis Channels 2 and 4 seem to do a great deal with relatively few statements.

**Productivity Analysis.**

The productivity was assessed on the basis of the metrics and the development time. Three measures were used these were as follows:

- **Nodes per day** is the number of nodes divided by the total development effort in man **days** for each channel.
- **McCabe per day** is the McCabe value divided by the total development effort in man **days** for each channel.
- **EFP per Month** is the number of Function points (as calculated earlier) divided by the total development effort in man **months** for each channel.
The 'nodes per day' productivity measure shows values for Channels 2, 3 & 4 which are not too dissimilar. The value of 7.03 for Channel 1 perhaps is a result of the simple style of its coding.

The 'EFP per month' measure cannot be taken too seriously, given that we have expressed some doubt about its validity for the DARTS channels and the low figure for Channel 4 is a direct result of the high Albrecht value for the 'C' language.

The 'McCabe per day' measure is the most similar over the channels, and shows a high degree of consistency.

It is noted that the overall productivity for Channels 1 and 2 was slightly greater than for Channels 3 and 4 for all three measures which appears to contradict the accepted idea that a development involving formal methods will always be more 'expensive' than a traditional one.

These figures give some idea of the development time required for safety critical code. However, these figures do not show how the interaction with other bodies necessary for the
licensing of safety critical software affects the cost of production. It is these interactions that must be planned for and their impact estimated.

Conclusions

The brief examination of two similar safety critical developments and a safety related monitoring system shows that in many ways the production of code from a set of clean requirements can be estimated fairly accurately. It is the interactions with both customer and other bodies that in many cases cannot be estimated accurately with any degree of confidence.

Justification criteria or principles were not available from the regulator. A large risk area arises from the non-prescriptive nature of the justification of the system. It is noted that all systems are different in behaviour and properties, however it is felt that early communications with the regulator could be used to aid the justification process. Obviously, this process should not compromise the regulator in any way. In the DARTS experiment a description of assessment and licensing criteria was drawn up. It was also noted in the results from DARTS that interaction with the assessor (the regulator’s agent) was seen to have been cost effective if the feedback had occured during development.

In both projects problems arose with the communications between interested parties, ie. the customer(s) and the users of the system. Where people are involved the presentation of data is subjective, therefore this is a great risk area, especially if more than one customer is involved. It is essential to gain access to the people who will be using the system in their every day work and to educate them as to the behaviour of the system so that they may make informed judgement of their requirements for monitoring such systems.

Iterative development is an excellent way of bringing the customer in to the activity of defining the requirements. However, this can develop into a long drawn out process, not sufficiently taken into account when estimating the cost of development. A two phased approach would be better where the customer initially works with the developer to define the requirements in a non-ambiguous way, sharing the risk. The second phase the subsequent development from these requirements.

From the DARTS experiment it can be seen that both traditional and mathematically formal methods of development will allow the production of high quality software. It is hard to judge whether the use of formal methods can compensate for lack of experience and maturity in a software team using traditional methods. But, as can be seen the use of formal methods does not penalise the development of safety critical software, on this scale at least, in terms of cost. Additionally, the use of formal methods allows tools to be used that can provide confidence that errors are not present in the resulting code.

Where a new safety system is developed in response to an identified hazard it is essential that the customer understands the proposed system, its behaviour and properties in detail. It is only then that he can recognise its potential, terms of his responsibilities on the plant and in turn form these into requirements for the monitoring of such systems.

It is noted that it is far easier to specify a system that does not include the subjective input from the user on the method of data display. These systems lend themselves more easily to being expressed in formal terms.
References

OECD/NEA CSNI-CNRA INTERNATIONAL WORKSHOP on LICENSING ISSUES of COMPUTER-BASED SYSTEMS IMPORTANT to SAFETY

Munich, GERMANY 5 - 7 March 1996

Reviewable software system of digital based reactor protection system for next stage PWR plants in Japan

Y. Yamamoto Kansai Electric Power Co., Inc.
* T. Aoyagi Japan Atomic Power Co.
M. Utsumi Mitsubishi Heavy Industries, Ltd.
H. Funakoshi Mitsubishi Electric Corporation
1. Introduction

In the next stage PWR plant, totally digitalized I&C system including safety system and soft-operation control room are planned to apply.

In this paper, we summarize reviewability of the safety software we are going to apply to the next stage PWR plants.

Basically, methods which have been applied to the design and manufacturing of the software for non-safety system in operating plants will be applied to the safety software.

From the period of the development of these methods used for the non-safety application, future use for the safety application was also took into consideration.

So sufficient experiences in the non-safety application assure the reviewability in the V&V activity for the safety software.

2. Design of functional requirements

Functional requirements for the digital reactor protection system are the same as the conventional system.

Simple signal processing and voting logic almost similar to the operating plants will be also applied to the reactor protection system in the next stage plants.

In addition, from the point of system architecture, design concept such as redundancy (4-channel, 4-train) or functional assignment (signal processing, voting logic, sequence control, etc.) is almost the same as the design of conventional system.

So functional requirements for the digital reactor protection system will be written in natural language same as the conventional plants.

Therefore functional requirements will be readable and reviewable by plant designer and utility people.

3. Design of equipment specification

Documentation structure same as the conventional system will be applied to equipment specification of the digital reactor protection system.

Important specifications such as system functions, functional boundary between hardware and software, and system interfaces are described in
composite block diagrams.

Structure of these diagrams are similar to the non-safety digital applications in the operating plants and have sufficient experiences.

In this composite block diagram, function to be installed is described as the connection of graphical symbols which represent basic functional elements generally used in the design of I&C (lag-module, AND operator, etc.)

This approach enables easy understanding and review of the functions of equipment not only for experts of the digital I&C system but also for people who has experiences in the conventional I&C system including operational staff or maintenance people.

This documentation is widely effective in the use of design, manufacturing, testing, operation and maintenance.

For the system software which determine basic system operation, the same software is used as the non-safety system in the operating plants which experienced intensive verification tests and sufficient experiences.

There are no need to change the specification of the system software for safety application, so only required specifications for each safety system are described in the specification documents

4. Design of software specification

Software specification of the application software is directly created from equipment specification described in the composite block diagram.

Software modules (POL modules) which correspond to the graphical symbols in the composite block diagram are connected together.

Programming of the application software is achieved at a software design / programming tool like a manner as CAD drawing tool to select POL module symbols and connect them with lines.

Completed application software is readable and reviewable not only by the designer of the software but also by the system designer and utility people.

This approach have been applied since the design of the software of the non-safety application for the operating plants, and has sufficient experiences.
5. Testing of software

In the software verification test, the software should be executed on the system which has the same configuration as the target system to verify system function.

Modules of the system software and functional modules of the POL are tested in the V&V activity through structural tests (white box tests) and functional tests (block box tests) carried out at object code level.

Total system function of the application software is verified using test bed which has the same configuration as the target system using simulated input signals.

System software of the digital reactor protection system is the dedicated one which is designed for the use of I&C system in nuclear power plants, and this system software operate in single task and fixed time interval without any interrupt operation.

So test condition is able to be established easily without any consideration for the timing of related signals and the operation of the target system to perform sufficient verification test with relatively small numbers of test cases.

6. Maintenance of software

Maintenance and modification of the software should be considered in the software life cycle.

Readable structure of the application software of the digital reactor protection system as mentioned before enables maintenance and modification of the system function easy and sure.

Software design / programming tool is also used as a maintenance tool. This tool can monitor the operation of the application software on-line in the POL diagram used in the design stage shown in its display, and effectively used in maintenance and testing at site.

In addition, to avoid uncontrolled modification of the software at site, software is installed in the EPROM which can keep the software code even in a failure of the power supply, and modification request from the tool is restricted by password management.
Fig. 1: Software Architecture
CONTRIBUTION TO THE SAFETY ASSESSMENT OF INSTRUMENTATION AND CONTROL SOFTWARE FOR NUCLEAR POWER PLANTS

APPLICATION TO SPIN N4

J.-Y. HENRY

INSTITUTE OF PROTECTION AND NUCLEAR SAFETY
SAFETY ASSESSMENT DEPARTMENT (DES)
STRUCTURE AND EQUIPMENT ANALYSIS OFFICE (SAMS)
CEA - FONTENAY-AUX-ROSES NUCLEAR RESEARCH CENTRE

International Workshop on
Technical Support for Licensing Issues of
Computer Based Systems Important to Safety

CSNI-CNRA
5-7 March 1996, MÜNCHEN
ABSTRACT

The process of licensing nuclear power plants for operation consists of mandatory steps featuring detailed examination of the instrumentation and control system.

This examination takes account of the technology-related aspects (integrated circuits, software) which were selected by the manufacturer for the programmed systems performing the safety-grade functions.

Significant changes were introduced by the operator in the process of designing and producing 1400 MWe pressurised water reactor safety systems and, in particular, in the case of the Digital Integrated Protection System (French abbreviation SPIN).

The methodology applied by the Institute of Protection and Nuclear Safety (IPSN) to examine the software of this system is set out in this communication. This methodology led the IPSN to carry out research and development into tools with the prime objective of producing an aid to analysis.
1. INTRODUCTION

1300 MWe pressurised water reactors (PWRs), like the 1400 MWe reactors, operate with microprocessor-based safety systems. This is particularly the case for the Digital Integrated Protection System (SPIN), which trips the reactor in an emergency and sets in action the safeguard functions. The softwares used in these systems must therefore be highly dependable in the execution of their functions. In the case of SPIN, three players are working at different levels to achieve this goal:

- the protection system manufacturer, Merlin Gérin,
- the designer of the nuclear steam supply system, Framatome,
- the operator of the nuclear power plants, Electricité de France (EDF), which is also responsible for the safety of its installations.

Regulatory licences are issued by the French safety authority, the Nuclear Installations Safety Directorate (French abbreviation DSIN), subsequent to a successful examination of the technical provisions adopted by the operator. This examination is carried out by the IPSN and the standing group on nuclear reactors.

This communication sets out:

- the methods used by the manufacturer to develop SPIN software for the 1400 MWe PWRs (N4 series),
- the approach adopted by the IPSN to evaluate the safety softwares of the protection system for the N4 series of reactors.

2. METHODS USED BY THE MANUFACTURER TO DEVELOP THE SPIN SOFTWARES

2.1. Description of the SPIN

The protection system for 1400 MWe PWRs, like that for the 1300 MWe PWRs, consists of the Digital Integrated Protection System (SPIN), which is made up of:

- four redundant and independent Protection Acquisition and Processing Units,
- two redundant and independent Safeguard Logic Units.

These units trip scram circuit breakers and control the safeguard actuators when two of the four redundant measurements of a given physical parameter exceed a predetermined value.

In the case of the N4 series of reactors, each unit of the SPIN consists of a Motorola 68000 microprocessor. The software, the binary code of which is stored in REProm, is written in C and in 68000 assembler code. Information is exchanged between the SPIN units via local area networks:

- eight redundant protection local area networks, of the NERVIA type, exchange information between the Protection Acquisition and Processing Units and the Safeguard Logic Units,
- two redundant signalling local area networks, of the NERVIA type,
actuator networks internal to the Safeguard Logic Units which transport protection orders between the processors and the actuator cards.

Special units are incorporated in SPIN for periodic testing.

2.2. Development of SPIN softwares

The general approach behind this development work is set out in the Software Quality Plan of the manufacturer Merlin Gerin. The softwares were mainly developed on a computer assisted specification and code generation sets of tools (SAGA), by means of programming rules, and by separating the design and verification teams. The softwares made no use of interrupts.

The process of developing a software in this context consists of seven stages, each of which gives rise to one or more documents. These stages and the associated documents are as follows:

- writing the software specifications, with the associated Software Specifications,
- the preliminary design stage, with the associated Software Preliminary Design dossier,
- the detailed design stage, with the associated Software Detailed Design dossier,
- coding stage, with the associated programming dossiers for the lists of instructions for the various components of the software,
- component integration testing stage, associated with the Software Integration Test dossiers,
- the software validation testing stage, with the associated Software Validation Test dossiers.

Four of these documents (Software Specifications, Software Preliminary Design, Software Detailed Design and Software Validation Test) give rise to a review by the persons in charge of the design and verification teams and the quality assurance manager. These reviews come under the umbrella of “quality” actions during the development process.

Furthermore, the Software Specifications are submitted for approval to the designer of the nuclear steam supply system.

The process of development enters into its next stage after each satisfactory review.

The SAGA atelier takes part directly in several stages of the development cycle. It makes use of five tools:

- the specification tool,
- the code generation tool,
- the programming tool,
- the documentation tool,
- the administration tool.

The specification tool is used during the preliminary and detailed design stages. Its interactive graphical interface can be used to produce a top-down description of the software to be developed in terms of its components, which are in turn broken down into easy-to-program components.
The code generation tool is used to obtain the C code for the softwares mentioned above.

The programming tool is an aid to the programmer when designing and developing the component source program, which cannot be generated automatically using the previous tool; it does this by suggesting a standard template and by checking compliance with certain writing rules. The interface for this component is generated automatically at the design stage.

The documentation tool is used as the preliminary and detailed design stages progress, and formats the written documentation associated with the components described using the specification tool.

The administration tool is used to manage access to atelier resources by the different users.

The programming rules must, in particular, ensure compliance with the provisions of standard IEC 880 “Computer software in nuclear power plant safety systems”, and ensure that the programming remains uniform, thereby simplifying the task of testing and maintaining softwares.

Separate design and verification teams were used, as in the software quality plan for 1300 MWe PWRs, in order to increase the number of independent checks.

3. EVALUATION OF SPIN SOFTWARE BY THE NUCLEAR OPERATOR

The safety approach adopted by the operator consists in demonstrating compliance with the safety functions by studying those accident scenarios requiring the use of safety class systems or equipment which must satisfy design, manufacture and installation requirements.

In the case of the reactor protection system, the operator performs an independent validation of the SPIN safety software in addition to the provisions adopted in the manufacturer’s quality assurance plan. He approves the specifications contained therein and performs audits, especially during the tests carried out during the software validation stage in the manufacturer’s premises.

Furthermore, the tests carried out on each item of equipment in the SPIN with validated programs (Protection Acquisition and Processing Units and Safeguard Logic Units), and then on the SPIN and its interfaces with other systems, give rise to joint reviews between the manufacturer, NSSS designer and the operator.

4. METHODOLOGY USED BY THE IPSN FOR EVALUATING THE SAFETY SOFTWARES

The technical support body (IPSN) of the safety authority (DSIN) is responsible for carrying out any investigations they deem necessary in order to ensure that the methods and techniques used by the manufacturer and operator guarantee that the SPIN software reaches the expected level of safety and exhibits an adequate degree of testability and maintainability. In order to do this, the support body pay particular attention to the following issues:

- rational and thorough methods of developing softwares by following a specific quality assurance plan (documentation and code);
- strict programming rules for producing a testable and maintainable program (code);
tests carried out to ensure sufficient coverage both in the manufacturer's premises and on site (simulation).

The assessment carried out by the IPSN does not cover all the equipment which makes up the SPIN, in view of their relative complexity. It was decided to limit the analysis to:

- all documentation associated with the SPIN technical specifications,
- a representative set of protection system functions.

The representative set of functions chosen for the safety assessment consists of two channels, one relating to a trip request and the other to a safety injection request. They each involve the functional units needed to perform a safety task:

- two process data acquisition units,
- a processing unit for this data allowing a partial trip to be executed corresponding to a trip request or a safeguard action request,
- a unit in charge of the majority vote controlling the scram circuit breakers and safeguard actions.

The softwares associated with this representative set of functions process one or more items of data from the process, from acquisition to the input terminals of the actuators. The methodology adopted to analyse the SPIN N4 softwares proceeds in successive steps to evaluate the various technical solutions put forward by the operator. Currently, there are six different steps involved in the evaluation of safety software:

- step 1, critical examination of the documents (see §4.1),
- step 2, evaluation of the quality of the code (see §4.2),
- step 3, determination of the critical software components (see §4.3),
- step 4, development of test cases (see §4.4),
- step 6, consistency study (see §4.5),
- step 6, robustness study (see §4.6).

Some of these steps are carried out in parallel, as is the case with steps 2, 3 and 5. Steps 1 and 2 are more specifically focused on a so-called static analysis, because they do not require running the program. Steps 4, 5 and 6 are focused on a so-called dynamic analysis. Step 3 is the transition between the static analysis steps and the dynamic analysis steps. In order to ensure an acceptable approach for the tasks to be performed and to provide the analyst with technical elements, special tools have been developed:

- a tool for modelling text in natural language to evaluate the completeness and consistency of specification and design documents,
- a static analysis tool which is used to evaluate the quality of programming and which, with its semantic analyser, is an aid for generating test cases,
- a tool which is used to determine the critical components in terms of the safety objectives which the software must meet,
- a simulation atelier which consists of the following tools:
  - a simulation and testing tool for carrying out dynamic analyses,
- a tool for describing environment programs for the simulation,
- a tool for processing the results which gives a graphical readout of the results of dynamic analyses.

The analyses set out in the sections which follow will enable the IPSN to meet the objectives stated above.

4.1. Critical examination of documents

An evaluation of the safety of the programmed systems is leading the IPSN to pass judgement on the relevance of information contained in the software specification and design documents, in respect of technical knowledge of the project and the standards relevant to the facility using these systems.

The examination of the protection system carried out as part of this evaluation takes account of the safety requirements of the installation, the system architecture and its specifications. The examination therefore consists in verifying the presence of all the functions needed to ensure that the installation is safe and to comply with the functional diversity which will make it possible to protect against common mode failures.

These functions make use of protection signals, for instance water-level in the steam generators, and result in protection actions being taken (scram or safeguard). Several protection signals appearing during a single accident sequence must be processed in different functional units of the SPIN (principle of functional diversity). This was the case, for instance, with the signals indicating very low pressure in the pressuriser and very high pressure in the containment, which arise during a loss of coolant accident (large break LOCA) and which are processed in two different functional units of the SPIN.

The development of softwares corresponding to the functions of the protection system is organised in a Software Quality Assurance Plan which gives rise to a very much documentation, throughout the development cycle.

This documentation consists of documents written in natural language (specifications, design, tests) and the source program itself.

The documentation is produced over a long time-scale, owing to the extent of this type of softwares.

Each document is analysed, not just to understand the functions performed, but mainly to check that there is no superfluous information (causes for complexity), or information which is incoherent or missing from the software documentation.

The AVIS method and its AVISO computer tool are an aid to examination involving the application of a systematic and thorough approach.

The method uses linguistic analysis to compile graphs showing the information set out in the document.

This operation relates each text element with its corresponding point on the graph.
This sort of modelling is more useful than a discursive or mathematical language, because it shows the relationships existing between the information contained in the text. The resulting overview can be used to focus attention both on the meaning and on the details of each piece of information.

This representation simplifies the task of examining the completeness and consistency of this information.

Besides, the references drawn up between the text and the graphic allow any anomalies picked out during the modelling process to be linked to the original text.

So, analysis of documentation produced over a long period of time and modified with each version of the softwares is more powerful by the ability of this tool to store information contained in the different texts together with observations and comments raised by the analysis.

4.2. Evaluation of the quality of the code

The source code of the representative set of functions are analysed to:

- search for any constructions dangerous for the type of language used:
  - data flow anomalies (definitions and uses of variable values, types of variables etc.),
  - arithmetic expressions (parentheses, division by zero),
- search for an incorrect or over complex structure in the programs:
  - multiple input or output loops,
  - variable index loops,
  - inaccessible code,
  - unnecessary code,
- verifying compliance with those provisions of IEC 880 deemed important by the IPSN.

The components from which the programming anomalies were detected become so-called sensitive components. The testability and maintainability of these components are in turn evaluated. Some of these components, mainly those containing variable index loops, could be tested during the robustness study of the program, thereby allowing the verification of their behaviour under these conditions.

A first campaign of analyses was carried out using a static analysis tool (structural analysers of the MALPAS tool) on the program of one of the SPIN functional units. These results showed some features of the code which could affect the testability and maintainability of this program.

4.3. Determining the critical software components

The software of the chosen unit processes several channels. Those parts relating to the two channels selected (the essential components) must be distinguished from the representative set of functions.

Amongst these components, the so-called critical functions, whose failure is likely to cause a severe system malfunction must be identified.
This is carried out using the Failure Modes, Effects and Criticality Analysis (FMECA) adapted for software analysis. This is the first stage in the AFFUT approach, which is intended to determine the most important unit functions for the IPSN to test.

This approach consists in evaluating the effects of postulated failures on each function of the softwares in turn.

An index of relative importance can then be calculated for each function, by taking into account the number and severity of failures, and hence categorise them.

The second step in the AFFUT approach consists in studying the critical functions in detail by analysing all the tests performed by the manufacturer.

If these tests are not sufficient for ensuring that the postulated failures cannot occur, these critical functions performed by the components which are called up in turn will be the subject of additional tests as part of the consistency and robustness studies.

4.4. Developing test cases

This is a two-part stage. The first part is based on an examination of the manufacturer’s tests with a view to the consistency study, the second is currently based on results from the semantic analyser of the MALPAS tool and is focused on the robustness study.

In the first part, the analyst selects from the series of manufacturer’s tests those which correspond to specific system operating conditions in order to verify system behaviour.

In the second part, the study of the critical and sensitive components that the analyst adopted continue with the PEGASE tool. This is used to give all the functional paths which lead to the values which can be assumed by each output variable of the component in question. It can be used to find the ranges of values for input data by means of the conditional relationships which describe the functional paths.

Values are selected for the input data in order to activate the critical components during the tests.

This analysis can also be used to verify the ranges within which the data vary from their specification values.

Besides, this type of analysis shows the dependencies which exist between the input and output data, and makes it possible to verify that the program code and specification conform, if the software contains such information.

4.5. Consistency study

An evaluation of the programs of the representative set of functions gives rise to a dynamic analysis which can determine, in a first stage, how consistently these programs perform with regard to their specifications.

The consistency study can be used to verify, for the representative example cited earlier, the values assumed by outputs from these channels (for instance controlling a scram) when the inputs assume values selected by the analyst from the nominal operating range of the
protection system. This study verifies the most significant aspects of the behaviour of the binary program which is actually used at the site.

The IPSN has developed for the purposes of this type of examination a set of tools which can simulate operation by execution of a binary program without recourse to equipment (CPU card, peripheral cards etc.) used on site. These tools, which are supported on a computer, make it possible to:

- compile an environment which reproduces the exchanges between each microprocessor and the circuits (clocks, communication circuits, memory etc.) which are associated with it in each unit of the protection system installed on site,

- run the binary programs of the units of the protection system by means of a microprocessor simulator, generating special files which track all interactions between the microprocessors and their environments, with a statement of the run time,

- present, in mimic form (time diagrams, curves etc.), the values assumed by the different variables monitored, in order to analyse simulation results.

The environment of the binary program and the microprocessor which runs it is simulated by developing special programs which replace the equipment called up by these programs. This development was carried out mainly by using a graphical description based on the SADT method.

Programs are run to take account of the values of the input variables given by the series of tests designed for this consistency study.

The implementation of such a simulated system is currently in hand. In a first stage, the normal operating conditions of the protection system will be selected to ensure that the model obtained using the environment developed for the purposes of this study is adequate. In a second stage, the program will be run to check the behaviour of the system in specific operating situations (degradation of the two-out-of-four voting logic, for instance) provided for in the specifications.

The simulated system and the associated tests series will be reused to verify that each version of the softwares works as well as before.

4.6. Robustness study

The main purpose of this study is to judge the behaviour of the programs of the representative set subjected to series of tests, defines in advance, which represent abnormal situations for the protection system or of the systems which provide it with information. The series of tests are focused on the critical or sensitive components detected during the previous steps. It sets in place an analysis which covers one area not touched on in the manufacturer tests.

This study makes use of the simulation tools set out for the consistency study, in order to create a more complete environment, making it possible in particular to arrive at certain internal program variables which are representative of the abnormal situation selected.

The results of the simulations obtained using the different series of robustness tests must be analysed to identify the state of each output variable of the system representative set of
functions. This implies ascertaining the values which should be obtained for each test case. Special semantic analyses are carried out to calculate the expected values (Oracle).

An analysis of the simulation results is carried out to identify, for system outputs, the consequences of malfunctions introduced and to draw conclusions on the adequacy of system behaviour with respect of the missions it must perform.

5. CONCLUSION

The tools mentioned earlier are in operation or in the experimental stage and are based on technologies currently available. Improvements are being made continually in this area and could lead to the resources used to carry out one or more of these analyses being changed. The IPSN is devoting a considerable share of its efforts to develop research programs into these issues.

However, the evaluation methodology set out in the above sections can be considered to be a satisfactory basis for examining the various aspects of safety software. This is an evolutionary approach, and other possibilities have still to be explored. These cover, inter alia, the self tests included in the protection system equipment softwares, the exhaustive nature of which affects the dependability of this system. Similarly, the problem raised by the software common modes and the extension of this evaluation method to other types of “real time” system will be dealt with in greater detail in the near term.
BIBLIOGRAPHY

Bussac J.P.¹, Jover P.², Conflant M.²
International Symposium on Nuclear Power Plant Instrumentation and Control

Soubies B., Le Meur M., Henry J.-Y., Bouc'h. J.¹
"Evaluation Methods for the Instrumentation and Control Safety Softwares of Nuclear Power Plants"
IAEA Specialists’ Meeting on Software Engineering in Nuclear Power Plants:
Experience, Issues, Directions

Notes:

¹Institute for Nuclear Safety and Protection CEA CENFAR
²Nuclear Installations Safety Directorate (DSIN) CENFAR
OECD/NEA INTERNATIONAL WORKSHOP

TECHNICAL SUPPORT FOR LICENSING OF COMPUTER-BASED SYSTEMS IMPORTANT TO SAFETY

Munich, 5-7 March 1996

TRENDS AND POSITION OF ITALIAN SAFETY BODY ON THE APPLICATION OF COMPUTER BASED SYSTEMS IN NPP’s

F. Zambardi
ANPA - National Agency for Environmental Protection
V. V. Brancati 48, 00144 Rome, Italy. Fax: +39 6 5007 2044

Abstract

This report presents the position of the Italian Safety Body (ANPA - National Agency for Environmental Protection) related to the application of computer based protection systems in Nuclear Power Plants.
Current activity performed in the Agency is resumed, it is related mainly to the evaluation of the application of systems based on digital technology in future design plants and eastern plants.
A synthesis framework is given to underline general principles (including error avoidance and fault tolerance) valid for quality and reliability of the safety related computer system and software, the evaluation of the effective application of these principles is at the base of the assessment of the adequacy of the digital programmable systems employed in Nuclear Power Plants. Particular emphasis is given to the definition of an appropriate development cycle where the pertinent phases of verification and validation are identified, in addition to all other engineering activities concerning system design and implementation.
A) BACKGROUND.

ANPA has been instituted in 1994 and has the primary role to perform development, evaluation and verification activities aimed to protect the environment and to achieve the environmental compatibility of products and activities. ANPA has taken in charge also the role of National Authority for Nuclear Safety and Radiation Protection, in this regard it makes use of experience gained throughout the construction and operation of 4 nuclear power plants and many other technical activities performed in cooperation with other organizations.

After the Chernobyl accident the Italian Government decided to shutdown the operating NPP's and to stop the construction of new plants. However significant activities have been continued in the field of nuclear safety, they include studies and evaluations related to new accident scenarios, participation in the development of advanced and passive safety plants in foreign countries, as well as supplying technical assistance to eastern countries in the context of European programs established to improve the safety of VVER and RBMK plants.

Before the interruption of construction of new NPP’s, in 1987, large efforts were devoted in Italy to the development of digital technology for nuclear safety applications. In particular, a microprocessor based protection system (called IPS - Integrated Protection System) was planned to be used on the new PWR's that were to be built. The primary concept design of IPS was originated in Westinghouse, but a heavy contribution to the system development came also from Italy, trough Ansaldo and Enea.

In the meanwhile, because of the lack of mature design and licensing guidelines at that time (apart someone, like IEC 880, NUREG 0497), the Nuclear Safety Authority (now named ANPA) made an independent effort to include correctly in the licensing process the peculiarities of the new system (use of digital instead of well experienced analog equipment, in particular use of software programs).

From 1987 a significant effort has been exercised concerning the evaluation of problems deriving from the application of digital technology in NPP's. A follow-up activity has been performed about the development, at international level, of new engineering standards, safety evaluation methods, and licensing approaches.

This background allows to undertake safety evaluations related to the application of computerized systems in NPP's, as part of the overall ANPA activity in nuclear field.

Related to the field of digital instrumentation review activities have been performed related to:
- digital control system of the ITER fusion plant
- SBWR digital protection system (in cooperation with GE)

Current activities include:
- cooperation with NRC in the certification process of AP600
- participation with other European countries to the TSO study program for resolution of safety issues related to evolutionary PWR’s.

Further activities to be performed in the next future are:
- review of EUR Rev. B document (European Requirements for Evolutionary PWR’s)
- participation with other European countries to the TSO program of technical assistance to Russia, related to the digital instrumentation of the Kola NPP.
B) POSITION ABOUT APPLICATION OF COMPUTER-BASED SAFETY SYSTEMS

General Aspects

The increasing diffusion of computerized systems, also in crucial applications, has raised a relevant problem about the quality and reliability of these systems. In particular the software is considered a critical component of computerized safety systems of nuclear power plants, and many concerns exist about the possibility of common mode failures caused by software errors, therefore the adopted design and certification methods must guarantee an acceptable level of reliability of software for safety related applications.

In this regard, many technical guides have been issued (the most important one is IEC 880 "Software for computers in the Safety Systems of Nuclear Power Stations"), moreover important applications are already existing (for instance computerized protection systems are used in France and UK), or are foreseen (new plants designs include extensive applications of digital technology). On the other hand there is not yet general consensus on guidelines and procedures to adopt in the licensing of safety related software, furthermore limited operating experience is available, compared to hardwired technology, concerning the application of digital technology in safety related systems.

Italian Position

The position maintained up to now in Italy about the use of digital technology and software is resumed in the following criteria for protection systems (included in the PWR general design criteria):

- adoption of proven technologies, demonstration of system conformity to safety requirements conducted with well established methods, availability of significant operating experience of the system or, availability of significant operating experience of similar systems complemented with testing in simulated operating environment
- adoption of a back-up diversified protection system for the most critical functions; diversification concerns both functions and equipment (including monitoring devices, data acquisition, logic devices and actuation equipment)
- software used in microprocessor based protection systems must undergo technical and management provisions for correct design and verification.

According to the position indicated above, a microprocessor based protection system could not operate alone, instead it should be used in conjunction with a back-up system based on different technology, besides specific provisions should be applied to software. These provisions are identified in IEC 880 and many other standards and guides (IEC, IEEE, IAEA).

However, if new evidence could be gained about the acceptability of the performance of the existing microprocessor based systems, the above stated position could be changed.

C) EVALUATION OF DIGITAL SYSTEMS EMPLOYED IN NPP’S

The assessment of the adequacy of the digital programmable applications in NPP’s is based on an overall characterization of the digital systems in which the following items are taken into account:

1) Design basis and general information related to the main fields of application with a given order of priority:
- protection system
- main control room (i.e. safety monitoring system, operator support systems)
- control systems.
The collected information is in terms of:
- reference rules, safety criteria and technical standards
- system architecture (functional diversification, function segmentation, redundancy, electrical independence and physical separation, interfacing and support systems, operator interface and treatment of manual commands)
- functional and performance considerations (functional requirements, accuracy, time response)
- reliability considerations (hardware/software fault tolerance, defense against common mode failures, comparison between design figures and operating experience data)
- aspects related to system integrity (abnormal environmental conditions, hazards)
- environmental qualification, immunity to electromagnetic interference
- use of diversified back-up protection systems.

2) Aspects related to digital technology (hardware side)
- data acquisition, logic and computational functions, data storage, data links
- fail-safe and self-test features
- other features related to improvement of reliability (i.e. simplified operating system and reduced number of interrupts in microprocessors, data integrity protection, deadman signals, watch-dog, communication robustness characteristics, back-up modules)
- use of optic signal for data transmission and to realize electrical separation.

3) Aspects related to software lifecycle
- software development cycle
- use of formal methods for software requirement specification
- engineering techniques and automated tools used for software design and testing
- verification and validation (software reliability targets and assessment)
- independence of Verification & Validation team
- use of diversification in software
- treatment of safety critical software (in particular the software for critical common functions like communications).
- validation of proprietary or commercial software to nuclear grade
- software maintenance

4) Quality, design and realization
- quality assurance program in design and construction
- documentation
- achievement of the nuclear grade quality level for digital equipment with respect to the quality level of commercial grade equipment
- design verifications and prototypes.

5) Operation considerations
- operating data records- significant operational events
- periodic testing
- availability figures.

6) References regarding licensing activities
- licensing approaches
- special analysis performed and documentation supporting the licensing process
- evidence requirements about software quality (or correctness).
D) A FRAMEWORK FOR SAFETY RELATED SOFTWARE

Software quality and reliability.

At the state of the art the attainment of reliable software is based on error avoidance (software errors generated during the development cycle are kept to a minimum), and fault tolerance (faults due to residual errors do not cause unacceptable degradation of software performance).

Important concerns with software quality and reliability come from operation considerations: the ultimate goal is software availability, thus the integrity against wrong inputs and unauthorized access must be assured, furthermore a design oriented to maintenance and change (following possible plant modifications) is to be addressed; these exigencies involve the attainment of specific attributes during the entire software lifecycle, such as security, robustness, maintainability and modularity.

Error avoidance and Fault tolerance.

The error avoidance and, ultimately, the software correctness, involve the application of adequate engineering techniques during the development cycle, as well as Verification & Validation activity (V&V). With error avoidance faults are prevented by means of reduction of residual design errors. At present time the free-error condition (to avoid any fault) is very hard to attain and to demonstrate for software of significant size (this is the case of protection systems).

Fault-tolerance is based on fault recognition and on the limitation of their degradation effects. Design provisions like diversity, independence, and separation allow to limit the effects of faults by confinement (limitation of the fault boundaries). Recognition of faults is based on built-in autotest capabilities. For instance rationality tests, cross checks, and elapsed time checks on computation results allow certain categories of faults to be recognized.

The provisions for the limitation of degradation effects of S/W faults include capability of reconfiguration, abort and retry, alarm, and fail safe response.

Development cycle

The identification of significant development phases and relevant milestones are at the base of the definition of the software development cycle.

The first step is the definition of the development program, which aims to identify activities to be performed, responsibilities, and documentation to be issued. V&V planning, configuration management, and quality assurance approach are also concerned.

The next step is the establishment of the initial technical references for the software product. These include functional and interface requirements or constraints from plant design. Applicable rules and standards are identified, like IEC-880, IEEE standards. Usually they are translated and increased with more technical details in engineering practices and technical handbooks.

The following phase is the software conceptual design. Plant requirements and constraints are translated into software requirements. Use of formal development methods and of automated tools can assist in the generation and verification of software requirements. Software architecture is defined in terms of main functional blocks and service modules. Functional specifications and design requirements are established enunciating "what" and "how" to do. V&V plans are also provided. The mentioned activity interacts with similar activity related to hardware.

The further design phase is at detail level. The software structure is divided in lower level modules. Test and integration of software sub-units into higher level units is performed. An important aspect concerning the effectiveness of this activity is the development environment, it consists of the set of
resources and tools available for software design and test. At the final stage software and hardware are integrated together. Prototype testing supports or complements this integration stage.

The progress of the design, besides the parallel process of V&V, is accompanied by several verifications performed at various levels and with different aims. Design reviews are the ordinary verifications performed by the design team at the completion of any relevant phase, before the related product is released for V&V. The overall software cycle undergoes QA activity and review, furthermore Configuration Management provides the control of the intermediate and final products.

Verification & Validation

The aim of V&V is to gain confidence on the correctness of the software system. An important condition for effectiveness of V&V is that activity must be performed independently from design: the V&V team is composed by people not involved in design activity. Verification is performed, according to established criteria, on the base of a plan which relates aspects like checklists, environment, etc. Basic verification activities are document verification and software testing.

Document verification is a top-down, step-by-step, checklist driven process aimed to verify that appropriate inputs are correctly incorporated into documents. Software testing includes structural and functional tests. The first ones aim to cover instructions, branches and paths of the program. The second ones check the performance in front of inputs derived from functional specifications. An interface with hardware test can be expected (related to data link configurations, communication functions). Acceptance criteria, for the evaluation of test results, include aspects like coverage degree, performance requirements. The effectiveness of the test strategy depends on the combination of structural and functional tests. The execution of tests and the evaluation of test adequacy takes strong advantage of automatic tools. Evaluation of test effectiveness is very important in regard of the rate of coverage of all possible errors.

Verification regards also system integration. Both document verification and software testing (at subsystem and system level) are performed in the integration verification. Special verification effort is needed for those software parts performing critical functions (common functions like autodiagnostic and communication), since they could represent main contributors to common cause failures.

Validation activity is based on functional tests performed on the integrated system. Test cases are derived from system functional and performance specifications, in this regard special constraints may derive from accident analysis. The use of plant simulators can represent an important component of the test environment. Full validation of the complete hardware and software system can imply and be supported with the construction of a system prototype to be tested in the real operating environment of the plant for a period of some years.
DIGITAL INSTRUMENTATION AND CONTROL SYSTEMS IN U.S. NUCLEAR POWER PLANTS

JARED S. WERMIEL, CHIEF, INSTRUMENTATION AND CONTROLS BRANCH, USNRC
IMPLEMENTATION OF DIGITAL I&C SYSTEMS

- Operating Plant Modifications
  - Westinghouse EAGLE 21
  - General Electric NUMAC
  - Babcock and Wilcox STAR
  - Nonsafety-related systems

- Advanced Reactor Designs
  - General Electric ABWR
  - ABB-CE SYSTEM 80+
  - Westinghouse AP-600
BASIS FOR ACCEPTANCE

- Quality
  - Hardware architecture meets IEEE-279/603
    - Single Failure / redundancy
    - Environmental qualification
    - Testability
  - Software built to rigorous design process - IEEE 7-4.3.2
    - Software lifecycle
    - V&V / testing
    - Configuration management

- Defense-In-Depth / Diversity
  - Alternate means of function in the event of software common-mode failure
DEMONSTRATION FOR LICENSING

- Quality
  - Documentation of process and products

- Defense-in-Depth / Diversity
  - Analysis of software common-mode failures
  - Provision for diverse functional capability
    - Manual or automatic
    - Digital or analog
UPDATE OF STANDARD REVIEW PLAN
CHAPTER 7

- Codify basis for acceptance and licensing demonstration
  - Endorse IEEE software design standards in new Regulatory Guides to support IEEE 7-4.3.2
    - IEEE 1012 & 1028, V&V Plans, Reviews, and Audits
    - IEEE 828 & 1042, Configuration Management Plans and Guidelines
    - IEEE 829, Test Documentation
    - IEEE 830, Requirements Specification
    - IEEE 1008, Unit Testing
    - IEEE 1074, Life Cycle Process
- Additional guidance in new Branch Technical Positions
  - Software reviews
  - Realtime performance
  - Defense-in-Depth / Diversity
  - Programmable Logic Controllers
  - Self test and surveillance testing
  - Commercial-off-the-shelf software

- Guidance aids regulatory staff and licensees/applicants
Regulatory Requirements and Safety Related Aspects
Concerning the Implementation of Modern I&C
in German Nuclear Power Plants

H.P. Berg, Th. Schaefer, F. Seidel
Bundesamt für Strahlenschutz (BfS), Salzgitter
Federal Republic of Germany

Poster paper to be presented at:

CNRA/CSNI International Workshop,
"Licensing Issues
of Computer-Based Systems
Important to Safety"

Munich, 5-7 March 1996
Table of contents:

- Abstract
- Poster figures:

1. Introduction
2. Licensing aspects of digital I&C implementation
3. Regulatory framework
4. General description of qualification concept
5. Safety relevant aspects of I&C modernization, specification
6. Points under discussion
7. Further investigation activities

- Explanations to poster figures
Abstract:

Worldwide investigations and first applications show technical and economical advantages of digital I&C use. In the Federal Republic of Germany there are at present intensive research and development activities, to introduce digital I&C in nuclear power plants for both operational and safety purposes. The development of such a system is in a far advanced state and the proposed concept is accordant to a first expert assessment in compliance with presently valid German regulations.

German nuclear power plant utilities are now starting to upgrade their currently used analog I&C systems. As a consequence, implementation of digital I&C in nuclear power plant safety systems is of increasing importance from the regulatory point of view. In the necessary procedure for their installation it has to be shown that the developed I&C system has fulfilled the functionality and reliability requirements as prescribed by the regulations. For its qualification a comprehensive concept is being developed by German expert organizations and the manufacturer which takes into account the properties of digital technology.

In this context the qualification status as well as a brief description of regulatory aspects in case of digital I&C introduction are provided. Beside the technical development intensive efforts are made to adapt and extent the regulatory framework for safety application of digital I&C. The current state is outlined. Furthermore a systematic set of safety issues regarding the introduction of digital systems to safety I&C is provided. Finally, the paper contains a listing of points which are still under discussion regarding digital I&C qualification including comments which indicate potential approaches towards their solution.
1. Introduction

Current status of digital I&C use:
- Non-critical safety applications:
  - e.g. power limitation function (Gundremmingen, KRB)
- Operational applications, e.g. PRISCA

Digital I&C projects:
- Modernization:
  - Neckarwestheim (GKN 1), Unterweser (KKU),
    stepwise upgrading is intended; status: feasibility study
- Future nuclear power plants:
  - European Pressurized Water Reactor (EPR)

Development status:
- Far developed status of digital safety I&C system, TELEPERM XS
  - First expert review, GRS/ISTec (1992):
    Implementation is possible in compliance with current German regulations
    From the safety point of view implementation is considered to be feasible

Basic Licensing Requirements:
- KTA 3501: I&C shall not determine the unavailability of the safety system
- Modernization:
  - Safety state of the plant must at least remain on the same level
- Future nuclear power plants:
  - According to extended regulatory framework (in preparation)

Functional requirements:
- According to safety analysis

Reliability requirements:
- According to safety significance of definite I&C functions
  (inherent design errors, Common Cause Failures, have strengthened importance and must be taken into account)

Choice of an adapted qualification concept covering all development phases
## 2. Licensing aspects of digital instrumentation and control implementation

<table>
<thead>
<tr>
<th>Phase of the Digital I&amp;C Modernization Process</th>
<th>Steps of Licensing Procedure</th>
<th>Remarks, Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Definition of Requirements</strong></td>
<td>Safety categorization of I&amp;C functions; early expert participation to find licensing prerequisites</td>
<td>DIN/IEC 1226</td>
</tr>
<tr>
<td><strong>Concept</strong></td>
<td>Choice of adequate qualification measures; evaluation of earlier operational experiences</td>
<td>IEC 880, 987 and draft of extended RSK guidelines (in preparation)</td>
</tr>
<tr>
<td></td>
<td>The reliability proof of all safety relevant I&amp;C functions shall be generally and sufficiently possible during all development phases</td>
<td>ISTec review of the SILT I&amp;C system with basically positive result (1992)</td>
</tr>
<tr>
<td></td>
<td>Feasibility study</td>
<td>GKN-1 modernization project regarding I&amp;C for power limitation</td>
</tr>
<tr>
<td><strong>Specification</strong></td>
<td>In case of modernization: Determination of the extent of I&amp;C alteration and influence on remaining/unchanged I&amp;C parts</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Derivation of reliability goals of I&amp;C functions and associated proof Design of safety I&amp;C functions against Common Mode Failure (kind and degree of diversification)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Establishing of licensing documentation (safety analysis report)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Licensing application</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Expert review of safety specification General proof that the reliability goals can be met</td>
<td></td>
</tr>
<tr>
<td><strong>I&amp;C Design</strong></td>
<td>Consideration of all specified reliability goals for I&amp;C functions, system and equipment</td>
<td>Development of hardware independent software Application of standard hardware (INTEL 80486) and standard operating network software</td>
</tr>
<tr>
<td><strong>Implementation/Integration</strong></td>
<td>Expert review of components and software modules regarding the several I&amp;C functions Laboratory test of I&amp;C functions, involvement of experts</td>
<td>Hardware/Software type test Laboratory test of TELEPERM XS components</td>
</tr>
<tr>
<td><strong>On-site installation</strong></td>
<td>On-site safety goal oriented proof of the safety relevant I&amp;C functions by experts In case of I&amp;C modernization: Partial license for several modernization packages Actualization of the safety specification</td>
<td></td>
</tr>
<tr>
<td><strong>Operation/Maintenance</strong></td>
<td>Supervision about the specified tests and maintenance procedures by the competent federal state authority Supervision about hardware and particulary software changes; associated documentation is required</td>
<td></td>
</tr>
</tbody>
</table>
3. Regulatory requirements

Listing of guidelines, KTA safety standards and other standards, essential for digital I&C application in Germany; short characterization, indication of relevance to hardware/software (HW/SW)

National regulations and KTA standards:

RSK-guidelines for PWR: general requirements to reactor safety; adaptation to digital I&C in preparation; HW/SW

KTA 1401: general quality assurance requirements; covers development and operation; HW

KTA 2201: protection against earthquake; HW

KTA 2206: protection against lightning; HW

KTA 3501: defines safety philosophy of reactor protection system; also valid for digital I&C; HW

KTA 3502: requirements for incident instrumentation; HW

KTA 3503, 3505: type testing of electrical modules, sensors and transducers for reactor protection; interpretation needed; HW

KTA 3508: system test of the I&C equipment of safety system; interpretation needed; HW

KTA 3507: factory test; operational experience on electrotechnical modules, equipment and system components of the safety system; interpretation needed; HW

KTA 3508: new safety standard will cover the topic of digital I&C application comprehensively, currently only a preliminary report is available, HW/SW

KTA 3901: standardization of alarms, HW

KTA 3904: requirements for control room design, HW

Selected international standards:

DIN/IEC 680: software qualification requirements, SW

IEC 987: hardware qualification requirements, HW

IEC 1226: categorization of functions systems and equipment; national version as DIN/IEC in preparation, HW/SW

Further relevant standards:

ISO 9001: quality assurance, qualification requirements; SW

DIN ISO 9000/3: quality management of software development; SW
4. General description of qualification concept

Procedure to fulfill protection goals

* Analysis of basic design / redesign
* Derivation of safety functions
* Derivation of associated safety I&C functions

Reliability requirements

* Categorization of the I&C functions regarding their safety significance, e.g. with respect to DIN/IEC 1226 and extended RSK-guidelines, in preparation
* Derivation of reliability goals for the definite I&C functions
  - deterministic: starting with the basic design and the deterministic requirements of regulations and technical standards (particular KTA 3501)
  - probabilistic: by reliability analysis based on operational experiences and reliability data of components

Qualification measures

Establishing a qualification methodology with:

* Choice of procedure model for software and hardware qualification; this model shall define the single steps of verification and validation, e.g. the life-cycle-model for software qualification, which covers all phases (development and operation); see explanations

* Choice of appropriate methods and measures for
  - fault avoiding:
    e.g. use of formal methods, graphic specification language, design tools with automatic documentation
  - fault detection and removal
    e.g. self test capabilities, self adjustments, application of simulators and special test and service stations
  - fault tolerance
    e.g. appropriate redundant architecture with diversification, use of fail safe principles (e.g. for reactor shut down system)

* Graduated association of these measures in correspondence to the safety categorization and phases of the qualification procedure; e.g. for the software life-cycle-model:
  - fault avoidance measures particularly important for the specification phase; specification faults are difficult to detect and can remain unidentified through the whole qualification procedure
  - fault detection measures are dominant during the integration phase; a test strategy has to be established in detail (e.g. with the following main steps: application unspecific type test, partly off-site integration test, full size integration test on-site)
  - fault tolerance should be introduced as ultimate measure to deal with the loss of safety critical I&C or electric functions (e.g. shut down system)
<table>
<thead>
<tr>
<th>Step</th>
<th>Task</th>
<th>Step</th>
<th>Task</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Verify completeness</td>
<td>2.</td>
<td>Verify completeness</td>
</tr>
<tr>
<td>2.</td>
<td>Verify completeness</td>
<td>3.</td>
<td>Verify completeness</td>
</tr>
<tr>
<td>4.</td>
<td>Verify completeness</td>
<td>5.</td>
<td>Verify completeness</td>
</tr>
</tbody>
</table>

5. Safety relevant aspects of instrumentation and control modernization, specification
6. Points under discussion

1. Existing German regulations do not cover digital I&C in detail: Extend of checks and scope of review have to be stipulated particularly for safety critical I&C

2. Quantitative reliability goals for digital I&C systems: Can reliability be proven sufficiently or is a diverse back-up system needed?

3. First step in life-cycle-model from requirements to specifications: Which specification language gives the best support for interdisciplinary communication?

4. Interfaces to control room design and alarm initiation system: How can the reliability goals of modern alarm initiation systems be defined (adaptation of control room design to advanced I&C concept, consideration of safety I&C categorization scheme)?

5. International harmonization of reliability requirements: This should be supported by international exchange of experiences regarding I&C qualification and operation

7. Further investigation activities:

Main discussion points are examined within accomplishing research activity initiated by BMU/BfS

Investigation programmes are under way covering the following issues:

1. "Qualification tools of digital I&C for NPP's" evaluation of tools for hardware/software qualification

2. "Safety and qualification requirements for digital I&C in future NPP's" major topic: EPR, harmonization of requirements, development/comparison of test strategies

3. "Safety related aspects of backfitting digital I&C" major topic: Estimation of possible backfitting concepts
Explanations to poster figures

1. Introduction

Up to now, there are only few digital I&C applications in German Nuclear Power Plants (NPP’s) with safety significance (e.g. local core monitoring system in Gundremmingen NPP, digital calculation component in the reactor protection system of Mülheim Kärlich plant). There are more frequent examples for non-safety applications, e.g. the process information system, PRISCA. As a consequence, there is no comprehensive licensing experience available in Germany regarding digital I&C systems.

Since the end of the nineteen-eighties German electronic manufacturers are developing digital I&C systems for safety as well as non-safety applications. Purpose of this digital I&C systems firstly is the substitution of conventional I&C systems in existing NPP and secondly the application in future nuclear power plants. For the I&C system designed for safety application, TELEPERM XS, being in an advanced developed state, a plant independent first expert review has been established in 1992. As a general result it has been stated, that an implementation should be realizable in compliance with persisting German regulations.

Additionally, the German Reactor Safety Commission (Reaktor-Sicherheitskommission - RSK) has recently stated that from the viewpoint of safety an implementation of digital, computer aided systems in German NPP’s shall be feasible.

On a worldwide prospect analog I&C equipment is in the process of being replaced by digital I&C and in the near future nuclear power plant utilities will be faced with problems as plant manufacturers will not guarantee I&C spare parts supply over the scheduled lifetime of the plants. Prompted by this and recognizing significant technical and economical advantage from worldwide first applications in countries like Canada, France, United Kingdom and USA German nuclear power plant utilities intend to upgrade their currently used conventional I&C equipment by digital systems. This procedure has started although positive operational experience with analog I&C has been gained in Germany.

The question of implementation computer aided I&C in NPP’s for safety application has therefore also reached increased importance from the viewpoint of licensing in Germany.

The case of replacement or modernization of safety I&C equipment in existing plants is considered as a substantial plant modification and, as a consequence, requires a licence in accordance with the German Atomic Energy Act. During the qualification process it has to be demonstrated that the safety state of the plant remains at least at the same level using digital I&C systems.

For licensing of future nuclear power plants which are expected to be equipped completely with digital I&C systems the national regulatory framework should be supplemented, particularly taking
into account relevant IEC-standards, extended RSK-guidelines and standards of the Nuclear Safety Standard Commission (Kerntechnischer Ausschuß - KTA) (s. also 3. and associated explanations).

There is actually no doubt that, due to the higher performance of computer aided I&C, functionality requirements stipulated by the operational objectives can be fulfilled as with conventional systems or even much better. Furthermore significant safety improvements are obvious, e.g. improvements concerning signal processing, presentation of information, decoupling of components as well as improvements by automatic documentation and self test. Concerns regarding the implementation of these systems in NPP's safety system are mainly caused by the necessary prerequisite for licensing to demonstrate that the required reliability of execution of the I&C functions is guaranteed.

Design errors in software or hardware parts of computer aided I&C systems will produce Common Cause Failures (CCF). As these complex systems have a limited testability the application of an adapted qualification concept with enhanced qualification measures is required to reduce the probability of undetected design errors.

2. Current practice of licensing

Fig. 2 gives an general overview on the licensing procedure associated with the several phases of safety I&C development.

Essential parts of the Atomic Energy Act are implemented by the federal states acting on behalf of the federal government. In particular, licenses for the construction and operation of nuclear facilities are granted by the responsible authorities of the federal states where the facility will be constructed. The competent regulatory authorities of the federal states are subject to supervision by the federal government, which is executed by the Federal Ministry for the Environment, Nature Conservation and Reactor Safety (Bundesministerium für Umwelt, Naturschutz und Reaktorsicherheit - BMU). BMU is responsible for developing the general safety strategy and for establishing the national policy on nuclear safety issues. The federal ministry is advised by expert committees, e.g. Reactor Safety Commission (RSK), which consist of independent experts. Furthermore BMU is competent to coordinate and harmonize the requirements concerning nuclear safety. In this context BMU issues its safety policy in the respective guidelines, safety criteria etc. In the licensing procedure for nuclear facilities the licensing authorities ensure that the safety requirements are met. Usually they consult experts, e.g. the Technical Inspection Agencies (Technischer Überwachungsverein - TÜV) and in special cases institutions like the Company for Plant and Reactor Safety (Gesellschaft für Anlagen- und Reaktorsicherheit - GRS/Institut für Sicherheitstechnology - ISTec). The Federal Office for Radiation Protection (Bundesamt für Strahlenschutz - BfS) beside its responsibility, to supervise construction and operation of facilities for the final disposal of radioactive wastes, gives
administrative and technical support to the federal ministry in matters of nuclear safety.

Safety significant backfitting measures like complete upgrading of digital I&C - as well as construction and operation of new plants - are subject to the provisions of the Atomic Energy Act and associated ordinances and therefore require a license. Normally the license is granted in stages e.g. by taking a decision about - normally three - partial construction permits and finally the operation permit.

In the case of modification (modernization or upgrading) of safety significant systems the license can be divided with respect to several modernization packages, each of which is to be separately realized during an outage period. This procedure seems to be particularly favourable to structure complex projects like I&C modernization.

Regarding the case of modernization I&C systems with low safety significance the authority has to decide either to establish a licensing procedure or a procedure in the frame of an approval. The last procedure is characterized by lower scope and deepness of required documents.

3. Regulatory framework

The process of international harmonization of qualification requirements on computer aided I&C for NPP application is still under way. E.g., the IAEA is currently reviewing its guidebook "Nuclear Power Plant Instrumentation and Control". The IEC is establishing a "chapeau" paper which shall generally cover several IEC standards and is developing two supplements to the software qualification standard IEC 880, dealing with issues as CCP, formal methods, use of tools and standard software, software maintenance, security, communications and self supervision. A supplement to IEC 1226 will deal with the use of probabilistic safety assessment in I&C categorization.

German nuclear guidelines and safety standards traditionally follow industrial developments and experiences and are issued if a common understanding between all participants in the licensing procedure has been achieved. At present, this process regarding safety critical digital I&C is still on the way and licensing is not finalized.

The persisting regulatory framework, in particular the standards of the Nuclear Safety Standard Commission (KTA), has been formulated for I&C systems based on analog technique. No requirements concerning the qualification of software are given so far. Generally these standards are still valid but have to be interpreted regarding digital technology.

In particular KTA 3501 defining the safety philosophy of the reactor protection system essentially holds also for digital I&C.
Fundamental stipulations regarding the architecture and reliability are given, e.g. for the degree of automation, kind of diversity and self supervision. But even more, KTA 3501 and the persisting version of the RSK-guidelines have already formulated key requirements necessary for qualification of digital I&C, e.g. the general demand to realize safety I&C systems in an as simple as possible structure. Also the basic qualification strategy now applied to digital I&C (operational experience or type testing) has been pointed out.

Recent international developments and the intention of German nuclear power plant utilities to stepwise upgrade currently used analog I&C systems have increased the importance of the process towards completion of a suitable national as well as international regulatory framework.

As a first step in 1991 one of the German TÜV, TÜV Bayern-Sachsen, has published a review guideline concerning licensing of digital I&C. Bavor publication, the guideline has been subject to comments of the expert community.

As a further essential step the German RSK is extending its guidelines with respect to digital safety I&C. Subjects as categorization of digital function and qualification are covered. The RSK-guidelines are recommendations to the superior licensing authority in Germany, BMU, and by this way become a mandatory requirement in the licensing procedure of NPP’s.

The KTA has established a preliminary report, "Computer aided I&C in NPP". The scheduled safety standard to this topic will cover the reliability requirements and qualification measures in more detail. Moreover it will provide interpretations for application of the valid persisting regulatory framework. But due to discussions concerning definite requirements on safety digital I&C qualification progress on this safety standard is currently postponed.

Following the aim of an international harmonization of regulations the BMU and its experts on behalf are actively participating in the proceeding of international standards and guidelines, e.g. IAEA guides, OECD guidelines, standards of IEC. In comparison to the national regulations the international regulatory framework is more detailed regarding the topic of digital I&C, but from the first the application of international standards is not mandatory in the German licensing procedure. However, in its recent draft of extended RSK-guidelines RSK has given hints to selected international standards. Relevant international standards are adapted to national conditions, e.g. DIN/IEC 1226 which is in an advanced state of draft supplemented by national foot-notes regarding for instance the German 30 minutes criteria; while IEC 1226 gives the option for "category A" human actions promptly following an accident.
4. General description of qualification concept

Generally, it is the aim of the qualification concept to qualify a computer aided I&C system which can be adapted to the conditions of different plant concepts/plants which have been constructed at different times. Therefore the German system under qualification can be applied for backfitting older plants currently in operation as well as for application to future NPP's.

Pursuant to the German Atomic Energy Act it has to be ensured, that the public and the environment are protected against damage resulting from nuclear radiation. This overall safety goal implies that at least in case of design basis accidents the four protection goals, namely reactivity control, core cooling, enclosure of radioactive material and limitation of radiation exposure have to be fulfilled. I&C functions are assigned to the functions of process technique of the facility. Accordant to the German defence-in-depth concept I&C functions are structured in control, limitation and protection.

Starting from the basic design phase in the case of new/advanced nuclear power plants or from a redesign phase in the case of modernization older plants the I&C functions are derived by means of an accident analysis. As a necessary prerequisite for qualification in a next step the I&C functions have to be categorized according to their safety significance. The German defence-in-depth concept provides the basis for the categorization of the I&C functions. In principle this procedure follows DIN/IEC 1226.

In a next step graduated reliability requirements have to be established accordant to the safety categorization of I&C functions. For I&C functions of the highest safety category the highest reliability requirements have to be stipulated. Therefore the reliability can only be checked by establishing this I&C functions in a very simple structure. But this is not a new aspect, e.g. compare the established structure of the analog I&C for reactor shut down.

To a first priority in Germany the reliability requirements are established in a deterministic way. The deterministic qualification concept (hardware and software) analogous to the persisting regulatory framework (s. Fig. 3, KTA 3503, 3505) is based on type testing, which performs a proof that specified properties/requirements are fulfilled. It is a new approach in the framework of qualification that the concept of type testing is extended also to the software. Prerequisite is a sufficient modularisation combined with clear defined software interfaces. This is one basic approach to restrict significantly the probability of occurrence of software CCF. If sufficient data for hardware components are available, alternatively a qualification by means of operational experience shall be possible. Concerning the qualification of software it is an essential prerequisite that the development process is directed by a software life-cycle-model. A suited model has been proposed (recent draft of RSK-guidelines, s. end of this chapter).
Additionally the qualification concept is supported by a stringent probabilistic assessment, in particular it is stipulated (KTA 3501), "that safety I&C shall not determine the unavailability of the safety system".

The feasibility of a respective analysis is to a great extent based on the modular structure of the system, i.e. the final system is established of a few number of software modules with defined interfaces. The correctness of these software modules can be proven nearly completely. As a prerequisite for qualification it is in particular assumed that reliability quantification for the software modules can be stipulated in the sense of conservative minimum values (number of failures per lines of code, ISTec expert review).

Concerning the quantification of software correctness further research activities are needed (Concerning this topic, which has to be discussed in the context of diversification and need for back-up systems see Fig. 6 and associated explanations).

Software-life-cycle model

- **requirements**
- **specification**
- **design top-down**
- **implementation**
  - coding
  - software-integration
- **installation**
  - system-integration (SW/HW)
  - construction, commissioning
- **operation, maintenance**

*protection goals / assumed events: malfunctions, design basis accidents beyond design basis accidents*
5. Safety relevant aspects of I&C modernization, specification

In order to provide a systematization of safety relevant aspects, it is intended to assign safety requirements for digital I&C and respective quality assurance measures to the several phases of I&C modernization. For the special application in the case of I&C modernization a first proposal is shown in figure 5 restricted to the specification phase. A complete proposal of this systematic - covering all life-cycle phases - is under development in the frame of a BMU/BfS/ISTec investigation programme (s. Fig. 6, third programme).

6. Points under discussion

Discussion regarding the implementation of digital I&C into the safety system of German NPP is not finished yet. Some of the major issues are listed in the following. Additionally potential approaches towards their solution are outlined.

Discussion points:

1) Current German regulations do not cover the digital I&C qualification process in detail. International rules and guidelines are not mandatory with respect to the German licensing procedure. As a consequence there are open questions concerning the extend of necessary checks respectively the extend for reviews of these systems.

In principle the existing regulations offer the possibility to demonstrate sufficient reliability by a suited combination of type test, operational experience and suitability test.

At present, there is only little operational experience regarding computer aided safety I&C available in Germany. Therefore the reliability proof preponderately shall be performed by test. Extent and scope of tests are determined by the safety significance/categorization of the I&C functions.

It should be possible to limit the qualification effort for digital I&C systems with low safety significance to a practical level. With modernized I&C systems of low safety significance qualification and operational experiences can be collected which are essentially necessary for further modernization steps towards safety critical digital I&C.

It is the current strategy in Germany to develop a variably applicable I&C system; firstly it should be supplied to I&C systems with low safety significance, e.g. to power limitation.

2) The quantitative reliability assessment is still under discussion, e.g., can reliability be proven sufficiently or is a diverse back-up system needed?
Generally a kind of diversification of safety critical I&C functions seems to be necessary. But different options for realization should be kept open, e.g. functional diversity, analog or digital back-up. The necessary prerequisite in this context is, that for the definite solution the required reliability is demonstrated by quantitative check.

From our point of view we consider a tool supported software development according to a software-life-cycle model as a necessary condition for successfully completion of type test and suitability test.

3) The first step in the modernization respectively life-cycle-model is the transition from requirements to specification. In this phase language problems between process engineers, I&C engineers and software engineers can occur. The application of an abstract, mathematic (formal) specification language can strengthen this problem.

However, it seems to be reasonable not only to accept the abstract and stringent specification languages, but also further classic formal tools as check lists for process engineers, I&C engineers and mathematicians respectively software engineers. First experiences with graphic specification languages show that they are very useful tools to avoid specification errors. In these tools often rules/restrictions and checks for combinination of the definite components are involved.

4) The interfaces of digital I&C systems to control room design and the alarm initiating system have to be adapted accordant to the categorization of the I&C functions in order to make the safety significance of information and alarms recognizable to the operator. In comparison to persisting analog technique, especially, the common processing of non-safety relevant and safety relevant process information by a computer network must be ruled out.

To solve this problem further investigations have to be performed. It is necessary to adapt the information and alarm concept to the design properties of the existing plant.

5) Worldwide harmonization of reliability requirements is still an open question.

As an important bilateral German activity, harmonization efforts in the framework of the French/German joint venture project, European Pressurized Water Reactor (EPR), should be mentioned. Exchange of operational experience collected with the safety relevant use of digital I&C is helpful in this context.
7. Further investigation activities

To cope with the persisting points under discussion BfS on behalf of BMU, has initiated 3 major investigation activities, which are accomplishing the current development and qualification process of safety digital I&C.

The investigation programmes cover the following issues:

- Evaluation of tools for hardware and software qualification of digital I&C; in this programme also the manufacturer is involved.

- Safety relevant aspects of I&C modernization in case of older nuclear power plants by implementation of digital components are systematically collected.

- Requirements on safety digital I&C; regarding to the French/German EPR project requirements shall be harmonized and test/proof strategies for qualification developed.
SESSION 3 - MORNING

HIGH INTEGRITY SOFTWARE - IS IT SAFE? COULD IT BE SAFER? - Prof. Nancy LEVESON, University of Washington

EXPERIMENTAL RESULTS FROM APPLYING INVERSE INPUT DISTRIBUTIONS TO VARIOUS SOFTWARE CONTROL APPLICATIONS - Dr. Jeffrey M. Voas, Reliable Software Technologies Corp.

TESTING OF COMPUTER-BASED SYSTEMS - METHODS, TOOLS AND RESULTS - Mr. Günter Glöe, TÜV Nord

VALIDATION OF TRANSFORMATION TOOLS - Dr. Johannes Brummer, ISTec

TOOL VALIDATION, MAINTENANCE AND OPERATIONAL FACTORS OF DIGITAL BASED REACTOR PROTECTION SYSTEM FOR NEXT STAGE PWR PLANTS IN JAPAN - Mr. Hisashi Funakoshi, Mitsubishi Electric Corporation

OPERATING & MAINTENANCE EXPERIENCE WITH COMPUTER-BASED SYSTEMS - Dr. Helmy Ragheb, AECB, PWG1
HIGH-INTEGRITY SOFTWARE:

Is It Safe? Could It Be Safer?

Nancy G. Leveson
leveson@cs.washington.edu
Safety is a property in interaction between components

Not in individual components

Emergent
Failure Models

- Accidents are the result of failures of system components.
- Advantage that failure rates in hardware are quantifiable.
- Drawback is that very narrow and does not include important factors.
- Prevention measures focus on eliminating component failures (including software).
Systems Theory Models

- Accidents arise from interactions among humans, machines, and the environment.

- Do not usually specify single variables or factors that account for accidents. Look at what went wrong with the system’s operation or organization to allow accident to take place.

- Safety is an emergent property that arises when components of system interact with an environment.

  A set of constraints related to behavior of components in system enforces that property.

  Accidents when interactions violate those constraints (a lack of appropriate constraints on the interactions).

- Software as a controller embodies or enforces those constraints.
Facts about accidents:

- Most accidents originate in system interfaces. Caused by complex, unplanned interactions between components of the system.

- Accidents often involve multiple failures of different components.

- Accidents intimately intertwined with complexity and coupling.

Computer-Involved Accidents: Almost all due to inadequate design foresight and inadequate requirements specification.

- Incomplete or wrong assumptions about operation of controlled system or required operation of computer.

- Unhandled controlled-system states and environmental conditions.
CAUSALITY

Accident causes are often oversimplified:

The vessel Baltic Star, registered in Panama, ran aground at full speed on the shore of an island in the Stockholm waters on account of thick fog. One of the boilers had broken down, the steering system reacted only slowly, the compass was maladjusted, the captain had gone down into the ship to telephone, the lookout man on the prow took a coffee break and the pilot had given an erroneous order in English to the sailor who was tending the rudder. The latter was hard of hearing and understood only Greek.

Larger organizational and economic factors?

A cause is a set of sufficient conditions (each is necessary but only together are they sufficient).

Common oversimplifications in ascribing causality:

- Human error as the cause of accidents
- Technical failures as the cause of accidents
- The legal approach to causality
- Looking only at single causes, ignoring root causes
Hierarchical model of accidents

Accidents must be understood and analyzed in a framework containing multiple hierarchical levels including technical, human, organizational, and regulatory perspectives.

Level 1: Mechanism of accident

Level 2: Conditions or lack of conditions allowing events at level 1 to occur.

Level 3: Constraints or lack of constraints that allowed conditions at level 2 to occur.

Constraints on:

- technical and physical conditions
- social dynamics and human actions
- management system and organizational culture
- government or socioeconomic policies and conditions

An accident is not understood until it has been examined on all these levels.
Root Causes of Accidents

1) Flaws in the Safety Culture

Safety Culture: the general attitude and approach to safety reflected by those who participate in an industry or organization: management, workers, and government regulators.

- Overconfidence and Complacency
  
  Discounting risk
  Overrelying on redundancy
  Unrealistic risk assessment
  Ignoring high-consequence, low-probability events
  Assuming risk decreases over time
  Underestimating software-related risks
  Ignoring warning signs

- Low Priority Assigned to Safety

- Flawed Resolution of Conflicting Goals

Downstream vs. upstream efforts
2) Ineffective Organizational Structure

- Diffusion of responsibility and authority
- Lack of independence and low-level status of safety personnel
- Limited communication channels and poor information flow

3) Ineffective Technical Activities

- Superficial safety efforts
- Ineffective risk control

  Failing to eliminate basic design flaws
  Basing safeguards on false assumptions
  Complexity
  Using risk control devices to reduce safety margins

- Failure to evaluate changes

- Information deficiencies
  Collection and recording
  Dissemination and use
History of Safety Engineering

Started in 19th century

Industrial Safety Pre-WW II

- Increase integrity of individual components
- After-the-fact accident investigations
  
  "fly-fix-fly"

Post-WW II

- Appearance of new hazards
- Increasing complexity
- Increasing exposure of public to risk
- Discovery and use of high-energy sources and nuclear energy
- Increasing automation of manual operations
- Increasing centralization and scale
- Increasing pace of technological change — less chance to learn from experience
SYSTEM SAFETY

The application of systems theory and system engineering approaches to prevent foreseeable accidents and to minimize the result of unforeseen ones.

- Emphasizes building in safety rather than adding it on to a completed design.
- Looks at systems as a whole rather than just subsystems or components.
- Takes a larger view of hazards than just failures.
- Emphasizes analysis and control.
- Emphasizes qualitative rather than quantitative approaches
- Recognizes tradeoffs and conflicts.
- Is more than just system engineering or safety engineering.

Cost and effectiveness
DEFINITIONS

Accident: An undesired and unplanned (but not necessarily unexpected) event that results in (at least) a specified level of loss.

Incident: An event that involves no loss (or only minor loss) but with the potential for loss under different circumstances.

Hazard: A state or set of conditions that, together with other conditions in the environment will lead inevitably to an accident (loss event).

Note that a hazard is not equal to a failure

C.O. Miller: "Distinguishing hazards from failures is implicit in understanding the difference between safety and reliability engineering."
**Hazard Level:** A combination of severity (worst potential damage in case of an accident) and likelihood of occurrence of the hazard.

**Risk:** The hazard level combined with the likelihood of the hazard leading to an accident plus the exposure (or duration) of the hazard.

**Safety:** Freedom from accidents or losses.
GENERAL APPROACH

- Identify system hazards (PHA).
- Evaluate or prioritize hazards.
- As define required functionality and allocate to components:
  - Apply formal hazard analysis to emerging design.
  - Optimize design for safety and other constraints.
  - Identify and resolve potential conflicts.

Results in identification of particular behaviors of individual components that could contribute to a system hazard. These hazards could not be eliminated at the system level so must be eliminated or controlled in design of components.

- After completing allocation of functionality, do system (SHA) and subsystem hazard analysis (SSHA) to identify causal factors, provide information for component hazard elimination and control, and ensure defined component behavior (requirements specification) is consistent with system safety constraints.

- Design and build components with safety constraints in mind (design safety into system and components).

- Verify safety of constructed system and evaluate operational feedback.
Management’s Role

- A sincere commitment to safety by management is the most important factor in achieving it.

- Top management’s participation in safety issues is the most effective activity in controlling risk and reducing accidents.
  - personal involvement
  - assigning capable people and giving them appropriate objectives and resources
  - setting up appropriate organizational structures
  - responding to initiatives by others.

- Responsibilities:
  - setting policy and defining goals
  - defining responsibility, fixing accountability, and granting authority
  - establishing communication channels
  - setting up a system safety organization
HAZARD ANALYSIS

Used for:

Design
Management
Licensing

1) Preliminary Hazard Analysis (PHA)
   - Identify, assess, and prioritize hazards
   - Identify safety design criteria

2) System Hazard Analysis (SHA)
   Performed on subsystem interfaces to evaluate safety of system working as a whole.

3) Subsystem Hazard Analysis (SSHA)
   - Determine how subsystem design and behavior can contribute to system hazards.
   - Identify hazards associated both with component failure and non-failure behavior.

4) Operating and Support Hazard Analysis
   - Evaluate hazards associated with environment, personnel, procedures, and equipment.
   - Should include the human-machine interface.
### HAZARD ELIMINATION
- Substitution
- Simplification
- Decoupling
- Elimination of specific human errors
- Reduction of hazardous materials or conditions

### HAZARD REDUCTION
- Design for controllability
- Barriers
  - Lockouts
  - Lockins
  - Interlocks
- Failure minimization
  - Safety factors and safety margins
  - Redundancy

### HAZARD CONTROL
- Reducing exposure
- Isolation and containment
- Protection systems and fail-safe design

### DAMAGE REDUCTION
SOFTWARE SYSTEM SAFETY

- Software is simply the design of a machine.
  
  Hardware components may fail.
  Software may have logic errors:
  
  — Written from incorrect requirements
  — Implementation does not match requirements (coding errors)

How to deal with software errors?

1) Build "correct" software
2) Make software fault tolerant using redundancy

But software may be highly reliable and "correct" and still be unsafe:

- Correctly implements requirements but specified behavior is not safe from a system perspective.

- Requirements do not specify some particular behavior that is required for system safety.

- Software has unintended (and unsafe) behavior beyond what is specified in requirements

3) Apply standard system safety techniques to eliminate or control hazardous software behavior
Computers do not produce new sorts of errors. They merely provide new and easier opportunities for making the old errors.

— Trevor Kletz

*Wise After the Event*
SOFTWARE SAFETY TASKS

• Trace identified system hazards to the software-hardware interface. Translate the identified software-related hazards into requirements and constraints on software behavior.

• Show the consistency of the software safety constraints with the software requirements specification. Demonstrate completeness of the software requirements with respect to system safety properties.

• Develop system-specific software design criteria and requirements, testing requirements, and computer-human interface requirements based on the identified software safety constraints.

• Trace safety requirements and constraints to the code.

• Identify the parts of the software that control safety-critical operations and concentrate safety analysis and test efforts on those functions and on the safety-critical path that leads to their execution.

• Identify safety-critical components and variables to code developers, including critical inputs and outputs (the interface).
• Develop a tracking system within the software and system configuration control structure to ensure traceability of safety requirements and their flow through documentation.

• Develop safety-related software test plans, test descriptions, test procedures, and test case requirements and additional analysis requirements.

• Perform any special safety analyses such as computer-human interface analysis, software fault tree analysis, or analysis of the interface between critical and noncritical software components.

• Review test results for safety issues. Trace identified safety-related software problems back to the system level.

• Assemble safety-related information (such as caution and warning notes) for inclusion in design documentation, user manuals, and other documentation.
Goals of the MURPHY Project

- Develop a theoretical foundation for safety of complex systems composed of
  
  Hardware  
  Software  
  Humans

- Build a methodology upon the foundation.

- Develop techniques to support the methodology.

- Evaluate the techniques using prototype tools.
Features of the Methodology

- Integrates system and software safety efforts.

- Applies software hazard analysis and control procedures throughout software development.

- Stresses building bridges between disciplines.
  
  System engineering
  Software engineering
  Human factors and cognitive psychology
  Organizational Sociology

- Stresses system-level approaches and viewpoints

  Many important properties arise in the interfaces between and interactions among components.

- Considers properties of the organization within which system is being designed, built, and operated.
The Safeware Methodology


- Management structures and procedures
- System hazard analysis
- Software hazard analysis
- Software requirements modeling and analysis
- Software design techniques (including human-computer interface)
- Safety verification (testing and code analysis)
- Change analysis
- Operational feedback
- Safety information system
SOFTWARE HAZARD ANALYSIS

- A type of Subsystem Hazard Analysis for software

- Goal is to determine how the software could contribute to system hazards.

- Identifies "software safety constraints"

- Need both a model of the behavior of the software and a model of the environment.

- Determine:

  1. If software behaves as specified, will it lead to a system hazard?

  2. What if there are failures in the environment?

  3. What type of software behavior could lead to a system hazard?
Experimental Results from Applying Inverse Input Distributions to Various Software Control Applications

Jeffrey Voas
Reliable Software Technologies Corporation
21515 Ridgetop Circle, Suite 250
Sterling, VA 20166
(703) 404-9293
jmvoas@RSTcorp.com, http://www.rstcorp.com
Software Fault-Injection

A "What-if" analysis technique, not statistical testing, not proofs.

The more "what-if" games you play, the more confident you become that your software can overcome anomalous situations.

Assesses robustness, tests firewalls, tests for the effect of imperfections (code/inputs).
The Big Picture:

What Happens Here?

Program

Output Events
“Problem” Spaces

All potential problems
(infinite and unknown)
All potential problems (infinite and unknown)

Existing problems (finite and unknown)
Anomalies We Inject (finite and known)

All potential problems (infinite and unknown)

Existing problems (finite and unknown)

Future problems (finite and unknown)
Simulate Anomalies

- Mutate Code
- Perturb Computational States
More Than Mutation: Propagation!

What Happens Here?
What Happens Here?

Program

Input

x = f(a,b,c,d,e,...)

corrupt(x)
Extended Propagation Analysis (EPA) Algorithm

1. Set variable count to 0.
2. Randomly select an input $x$ according to $D$.
3. Alter the value of variable sensor input variable $a$ or program variable $a$ at location $l$ and execute the succeeding code.
4. Increment count if the output satisfies $PRED$.
5. Repeat steps 2-4 $n$ times.
6. Divide count by $n$ yielding an extended propagation estimate: $\psi_{alPD}$.
   $1 - \psi_{alPD}$ is the fault-tolerance.
\textbf{PRED: Redefining Failure}

- Not all failures of a software system are equally hazardous.

- In certain situations, (1) if a particular variable becomes corrupted in \textit{any} manner, it is hazardous. In other cases, (2) it might be a set of conditions, e.g., if \texttt{variable\_1 > 5} and \texttt{variable\_2 > 100}. And in other cases, (3) it might be dangerous only when \texttt{variable\_1 = 103}.

- \textit{PRED} is our language for formally specifying hazardous output states.
Mechanisms to Perturb Data

```c
int flipbit(int x, int y)
{
    return (x ^ (1 << y));
}

int allbitshigh()
{
    return((int)~0);
}

int allbitslow()
{
    return((int)0);
}

int offbyone(int x)
{
    if (lrand48() & 1)
        return (x+1);
    else
        return(x-1);
}

int flipnbits(int x, int n)
{
    int bits = 0;
    int bitpos = 1;
    for (int i=0; i<n; i++)
    {
        bits |= bitpos;
        bitpos <<= 1;
    }
    for (int j=0; j < sizeof(int) * 8; j++)
    {
        int xbit = lrand48() % (sizeof(int) * 8);
        if ((!(bits & (1 << xbit))) != (!(bits & (1 << j))))
        {
            bits = flipbit(bits, xbit);
            bits = flipbit(bits, j);
        }
    }
    for (int k = 0; k < sizeof(int) * 8; k++)
    {
        if (bits & (1 << k))
            x = flipbit(x, k);
    }
    return(x);
}
```
Balls and Urn:

$\Psi_{alPD}$

- Our analysis can be viewed as selecting different colored balls from an urn where:
  - Black ball = input on which program output satisfies $PRED$.
  - White ball = input on which program does not satisfy $PRED$. 
Relating the PDF to Ball Density
Ball Stringing

- *Fault size* (really *anomaly size*) represents the number of inputs that satisfy *PRED* for a specific simulated fault class.
- The following urn represents five different anomalies in the program, each of size *one*.
Ball Stringing (cont.)

- This urn has five inputs that satisfy $PRED$ that are all caused by one anomaly in the program.
- This anomaly is of size *five*.
Inverted Distributions: Rare Inputs

- **Definition**: a "rare" input value is one that is unlikely to be selected according to $D$, where $D$ is the operational profile.
- Numerical input data.
- In general, by sampling from $D^{-1}$, you are sampling rarely observed input values.
- If you perform the EPA algorithms with $D^{-1}$, you assess the fault-tolerance of the program under unusual operating circumstances. Since you are very unlikely to test according to $D^{-1}$, i.e., select the high-probability inputs of $D^{-1}$ during test, this is interesting information.
Magnitudes of Inputs and Probabilities of Failure

Probabilities of failure

Order of magnitude numbers of tests
yawdamp.c

- several thousand SLOCs.
- analyzed with $D$ and $D^{-1}$
- identified one region of yawdamp.c where the fault-tolerance was on the order of $10^{-1}$ with $D$ and 0.0 with $D^{-1}$ which is a significant difference.
Histogram for Random Input Values Generated for YAWDAMP_STATE Variable
<table>
<thead>
<tr>
<th>Browse Level</th>
<th>Fault Tolerance (Original Gauss)</th>
<th>Fault Tolerance (Inverted Gauss)</th>
</tr>
</thead>
<tbody>
<tr>
<td>main.c</td>
<td>* 0.43478261</td>
<td>* 0</td>
</tr>
<tr>
<td>yaw.c</td>
<td>* 1</td>
<td>* 1</td>
</tr>
<tr>
<td>YAWDAMP</td>
<td>* 0.43478261</td>
<td>* 0</td>
</tr>
<tr>
<td>WOUT</td>
<td>* 0.72</td>
<td>* 1</td>
</tr>
<tr>
<td>FOLAG</td>
<td>* 1</td>
<td>* 1</td>
</tr>
<tr>
<td>lim</td>
<td>* 1</td>
<td>* 0</td>
</tr>
<tr>
<td>Line 280</td>
<td>0.43478261</td>
<td>0.51</td>
</tr>
<tr>
<td>Line 282</td>
<td>0.77</td>
<td>0.53061224</td>
</tr>
<tr>
<td>Line 286</td>
<td>0.43478261</td>
<td>0</td>
</tr>
<tr>
<td>Line 288</td>
<td>0.71428571</td>
<td>0.52941176</td>
</tr>
<tr>
<td>Line 292</td>
<td>0.98181818</td>
<td>* 1</td>
</tr>
<tr>
<td>FCAS</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>LINTERP</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Applications

- Magnetic Stereotaxis System
- Nuclear Regulatory Commission Pilot Trials (January, 1996)
- Bay Area Rapid Transit
COTS Abstraction: Someday?
Summary

- Fault-injection methods have worked well for years in the *physical* world.
- Realize that the results of fault-injection are *limited* guarantees.
- The quality of the anomalies injected *is* very important, but as we have discovered time and time again, even *default* classes of anomalies detect deficiencies.
- *Inverted operational profiles* can be used to analyze the quality of a program’s output when it is operating in the rarest of modes.
- Software fault-injection mechanisms *complement* formal methods.
Testing of Computer-Based Systems

Methods, Tools, and Results

Dipl.-Phys. Günter Glöe
Dipl.-Inform. Dipl.-Ing. Gereon Pillmann
Dipl.-Phys. Ronald Westhäußer
Technischer Überwachungs-Verein (TÜV) Nord e.V.
Hamburg, Germany
February 1996

1 Introduction

About the year 1970 we started with the assessment and testing of safety related real time computer systems. The very first system was a computer system for use as a reactor protection system. At that time writing software was not yet an engineering discipline but seemed to be a little artistic. Assessing and testing of computers was very new. Neither a lot of rules or literature nor people with a lot of experience could be found.

To manage this problem we took into account nearly every method of assessment, asked the suppliers of the computer systems to test their systems according to any method and then argued that they neither used a good method nor did a good test. This was not very efficient for getting a license — that computer-based reactor protection system was never installed for closed-loop operation —, but very efficient for learning.

Still this is a good basis for our work.

Starting from that point during the following years in several type and system tests we learned to assess systems rather important to safety, but not always as extremely important as a reactor protection system:

- monitoring and trouble logging systems (10 and more computers),
- conventional instrumentation and control (I&C) systems,
- limitation and other protection systems,
- medical and railway electronic equipment.

During this quarter of a century we more and more felt, that the growing abilities of computer-based systems could help us in the assessment and testing of other computer-based systems. And so we started to develop own tools and even some systematic methods suitable to support the testing procedures of hardware and software.

Our today's state concerning methods and tools development will be mirrored within the next few chapters. It has been divided up according to the following three headlines:

- hardware testing (chapter 2),
• testing of software near to hardware (chapter 3.1),
• software development procedure (chapters 3.2 and 3.3).

Additionally we present some failure numbers from operational experience with nuclear power plant systems, which might give a little impression of the effectiveness of the methods and tools developed (chapter 4).

2 Testing of Hardware

In this chapter the testing procedure developed at TÜV Nord for a type test of highly complex hardware components for the construction of computer-based I&C systems will be described. The concept of this procedure is shown in Figure 1 (see next page).

2.1 Method for Hardware Testing

In order to make the type test practicable and understandable the testee has to be reduced to simple and separately testable entities. These entities we named test objects. There are three kinds of test objects:
• all hardware components in the List of Assembly Groups,
• all Papers Required by Standards,
• all functions in the List of Functions.

The functionality of the components must be derived by the assessor during the testing procedure using the functional information contained in the required papers.

For each kind of test object possible tests are specified in a Frame Testing Table. The suitable tests to be done are listed in the Testing Table for Required Papers or in the Testing Table for Specified Functions. Test procedures are chosen from the List of Test Procedures.

For each required paper and for each specified function a file (Procs/Results for Required Papers or Procs/Results for Specified Functions) is created containing information on the test object itself, the tests to be done, the test procedures chosen, and the test results.

As an essential result of this theoretical test procedure a Requirements Specification for Practical Tests is created.

2.2 Experience with Hardware Testing

The test method described has been developed to realize the necessary depth of tests in testing highly complex hardware components. But the reachable depth of tests has shown to be limited. On the one hand the reduction of highly complex components to simple test objects is insufficient or even impossible. On the other hand the number of test objects seems to increase exponentially with complexity. A greater depth of tests can be reached only by conceptually and constructively limiting the complexity of components to the actually necessary extent: 'Keep it simple and stupid!' (KISS).
Figure 1: Test Scheme for Hardware Type Testing

Delivered Papers
- List of Papers
  - List of Missing Papers
  - Delivered / Required Papers Correl.

Test Standards
- List of Standards
  - Papers Required by Standards

Expert Knowhow
- Test Procedures

Assembly Groups
- List of Assembly Groups

Frame Testing Table

Testing Table for Required Papers
- Procs/Results for Required Papers
  - Results of Theoretical Tests

Testing Table for Specified Functions
- List of Functions
  - Procs/Results for Specified Functions
  - Requirements Spec. for Practical Tests

- Test Report and Certificates
- Results of Practical Tests
3 Testing of Software

3.1 CATS

The Code Analyzer Tool Set is a set of tools envisaged for quality assurance and safety evaluation of software. Its main application area is software for embedded controllers or similar usage based on fairly small microprocessor systems. CATS was designed for use with dependable systems.

It was envisaged for four areas of assignment:
• supplying an overview on the investigated program,
• making weaknesses of the investigated program visible, in order to give priority to these when evaluating quality,
• collecting information to facilitate the decision on which methods to use for evaluating functionality,
• first interpretation of control flow and dataflow as preparation for analysis of functionality.

The concept of CATS is shown in Figure 2 (see next page).

3.1.1 Method of CATS

CATS implements static analysis. It is starting from machine code. By means of a special document, the hardware definition, the hardware environment directly connected to the processor (e.g. memory and ports) is taken into account for software analysis. While some parts of the disassemblers are processor specific, the interface files created by the disassemblers and the analyzers themselves are independent of processor and programming language used.

This approach guarantees that the software really used is analyzed and not any other version. Patches, if any, are taken into account. It provides independency of program documentation, its availability and its errors. Starting from machine code provides — may be the only — reliable basis for run-time computation of the software tested.

After disassembling (by DisCAT) CATS breaks the program down into its routines (by ProCAT). For purpose of control flow analysis (by CoCAT) and time computation (by RealCAT and TimeCAT) each routine is represented by a graph. May be different from other analyzers, CATS does not use matrices to represent the graph but chained lists or a special vector (the CATS—Vector). Advantages in comparison with adjacency matrix represented graphs are:
• less memory,
• improved time behaviour,
• the same algorithm applicable for computation of run time and number of paths of the testee,
• possibility to represent dead loops (important elements of process control computers),
• possibility to represent control flow abnormalities as 'jump to the next statement’ or 'conditional branch to the next statement’.

In [Zus91] precise definitions of a lot of software complexity measures are given. Based on these MeCAT does some metric computation.
3.1.2 Tools of CATS

The Code Analyzer Tool Set consists of seven tools, four of them shown in Figure 2:

The processor specific disassembler DisCAT acts as a front processor. It operates program path related and produces the source data base for the CATS analyzers and some global metric data about the code processed.

ProCAT is provided for the analysis of the global controlflow structure. The tool analyzes the composition of the code in terms of programs, interrupt routines, and subroutines. For these components the relocation and the interaction by call or code sharing is evaluated.

CoCAT does the controlflow analysis of routines defined by ProCAT. It provides well documented and reproducible quantitative and qualitative informations on formal quality criteria as well as on realized functions.
Accordingly DaCAT analyzes the dataflow of the whole program and of routines defined by ProCAT, providing informations in the same manner.

Besides these main CATS tools there are three further tools: RealCAT and TimeCAT which analyze the runtime amount for programs, interrupt-routines and subroutines, and MeCAT, which computes metrics and arranges statistical data gained by control flow and data flow analyses.

3.1.3 Experience with CATS

We have been using our Code Analyzer Tool Set for about five years. We have applied it in the assessment of some ten safety relevant computer systems. It has been conceived as an aid for quality assurance and testing, not as a tool providing automated assessment. And that is what CATS really does: it helps to identify potential weak points in the testee. So quality assurance staff and assessors do not need to deal with the huge amount of code well done, but may concentrate their limited effort on points which may be critical.

3.1.4 Results of Analyses Using CATS

An overview about the software used in some safety related application obtained with CATS is shown in the table below.

<table>
<thead>
<tr>
<th>ID</th>
<th>Size [bytes]</th>
<th>Number of Routines</th>
<th>Recursive Routine Calls</th>
<th>Hints on Weaknesses</th>
<th>Number of Interrupt Inputs</th>
<th>Code Sharing</th>
</tr>
</thead>
<tbody>
<tr>
<td>1a</td>
<td>875</td>
<td>13</td>
<td>no</td>
<td>33</td>
<td>4</td>
<td>no</td>
</tr>
<tr>
<td>1b</td>
<td>941</td>
<td>13</td>
<td>no</td>
<td>37</td>
<td>4</td>
<td>no</td>
</tr>
<tr>
<td>2a</td>
<td>123346</td>
<td>229</td>
<td>no</td>
<td>6383</td>
<td>1</td>
<td>yes</td>
</tr>
<tr>
<td>2b</td>
<td>123404</td>
<td>229</td>
<td>no</td>
<td>1348</td>
<td>1</td>
<td>no</td>
</tr>
<tr>
<td>3a</td>
<td>2724</td>
<td>48</td>
<td>no</td>
<td>82</td>
<td>6</td>
<td>yes</td>
</tr>
<tr>
<td>3b</td>
<td>7098</td>
<td>151</td>
<td>no</td>
<td>380</td>
<td>6</td>
<td>yes</td>
</tr>
<tr>
<td>4</td>
<td>445</td>
<td>2</td>
<td>no</td>
<td>2</td>
<td>0</td>
<td>no</td>
</tr>
</tbody>
</table>

System with ID 1a is the software of a commercial available PLC not specific for nuclear or safety applications. Because of insufficient time behaviour it has been tuned to System 1b. This tuning resulted in some additional code and an increase of potential weaknesses.

System 2a seemed to be not really acceptable. It has been improved. So the hints on weaknesses decreased by a factor of about 4.7, and there was no more code sharing detected.

System 3a has been designed for safety usage in nuclear application. Because of additional functionality a standard mathematics library was added. The resulting System 3b has a much larger number of routines (this is not surprising) but also increased number of hints on weaknesses.

System with ID 4 is the software of a single functional block for a new control system which is conceived for usage in safety systems as well.
3.2 TASQUE

TASQUE is the abbreviation of Tool for Assisting Software Quality Evaluation. The main development work on this tool is based on definitions done within the EUREKA project EU 240 (German part funded by the German Ministry for Research and Technology, BMFT). Partners in this project have been CEP (France), ENEA, ETNOTEAM and ISMES (Italy).

The TASQUE aim is to provide an easy-to-use way to:
- define software quality attributes from user’s point of view derived from the requirements of risk-based standards and special project needs,
- translate and refine these quality attributes to attributes relevant to programmer’s and quality engineer’s point of view (change attributes to internal ones),
- propose and manage metrics and checklists for the different steps of life cycle,
- transform metric values (including checklist scores) to quality attribute scores and generate reports.

The evaluation procedure consists of a derivation and an integration phase. The whole procedure can be divided into the following steps:
- definition of quality goals from user’s point of view ("capabilities"),
- translation and refinement of quality attributes into "properties" and "features",
- proposition of metrics and checklists,
- transformation of metric scores and checklist answers.

The informations which are necessary to run the derivation and integration path are defined in a quality model. The model consists of:
- a set of capabilities, properties, features, and metrics (including checklists),
- link intensities between them,
- the risk-class groups, risk classes, and risk parameters,
- requirements of different standards with respect to the capabilities and properties.

TASQUE is able to process different quality models. Up to now only the models based on the TASQUE definition phase and ISO 9126 have been used. It is, however, possible to tailor these models according to the needs of the company or a special project.

The concept of TASQUE is shown in Figure 3 (see next page).

3.2.1 Method of TASQUE

Methodological basis of TASQUE is the concept of quality models. These models are mainly concerned with description of the non-functional attributes of control systems, computers or software as for example reliability or safety. Since about 1980 some of these quality models have been published, e.g. /Bow85/, /Deu88/, and /Art85/. Most of them do not describe a complete model but only selected levels.

Quality models start with quality characteristics as usability, efficiency, reliability or safety which are characteristics from the user’s point of view. They support deriving measurable attributes which have to be met in realization to achieve the required quality characteristics. (These quality models are concerned with product quality, not with process quality!)
Figure 3: Concept of TASQUE

We start to use these models to identify quality related user requirements which are in contradiction to safety, e.g. the requirement for very high storage efficiency. And the models are applied to identify those characteristics of the realized software which should be checked to decide on the level of safety and reliability reached.

TASQUE uses a four level quality model. Top level are the quality characteristics from the user's point of view. Bottom level are characteristics which may really be measured against an existing software. Mapping between the levels is via matrices.
Composing a complete quality model from the parts published in literature was accomplished within *TASQUE*.

### 3.2.2 Experience with *TASQUE*

Until now *TASQUE* was applied to two non-TÜV projects, an industrial one and a research project. From these applications we have learned:

- Participants in a software project — user, manufacturer, assessor — assume quite different goals for the non-functional quality characterististics. And even in the development team — e.g. project leader and software engineers — the goals may not be the same.
- Based on quality models and guided by a moderator it is possible to identify within a few hours talk those quality characteristics where an agreement in the goals has not been achieved.
- For most characteristics after a short discussion the reason for the discrepancy becomes clear and the participants agree in the necessary quality goal profile.
- Within such a quality goal profile there may be inconsistencies from safety point of view or unnecessarily high requirements from commercial point of view. It saves money, time and a lot of trouble to identify such problems in the very first phases of a project.
- Whilst the two top levels of *TASQUE* quality models seem to fit well to practical use, the bottom level of *TASQUE* quality model still has drawbacks.

To improve the theoretical basis for software product quality considerations including safety and reliability aspects and to broaden the scope from product quality — which is the only concern of *TASQUE* — to process quality was the reason to participate in "Further Projects on Software Quality".

### 3.3 Further Projects on Software Quality

#### 3.3.1 *SQUID*

*SQUID* is the abbreviation of *Software Quality in Development*. It is an ESPRIT project. Partners are ENGINEERING and ENEA (Italy), CAP VOLMAC (Netherlands), DELTA (Denmark), NCC (UK), and TÜV Nord (Germany). The *SQUID* method and toolset will allow a software development organisation to:

- monitor, control and predict product qualities at various stages in the development life cycle by the use of objective measurement,
- demonstrate conformance with ISO/IEC 9126 and related standards for product quality,
- learn in a deeper and more systematic way about the software process and product, and to feed the appropriate experience back into current and future projects,
- assure final product qualities.

The project *SQUID* will end in April 1996.

In order to get some experience with the application of *SQUID*, a new project has been born:

#### 3.3.2 *VALSE*

*VALSE* is the abbreviation of *Validating SQUID in Real Environments*. It is a project within the Technology Validation and Technology Transfer Projects of the EC. The aim of *VALSE* is to demonstrate the suitability of the *SQUID* method and toolset. Partners are ENGINEERING and ENEA (Italy), DELTA (Denmark), Keele University (UK), and TÜV Nord (Germany).

*VALSE* consists of three validation actions:
The Mediterranean validation action will apply SQUID for development of high quality software products.

The Anglo-Saxon validation action will apply SQUID for object oriented development and re-use of reusable software components.

The Westcontinental validation action will apply SQUID for third party software product quality assessment.

The aim of FALSE will be pursued through the following objectives:

- Validation
  - to establish effective interaction with a base of target user organisations,
- Adaptation
  - to establish a practically proven and directly applicable method for applying the SQUID approach to managing product quality in client organisations,
  - to gather significant real life projects data to improve the Experience Base which SQUID estimation and advice capabilities rely on,
- Dissemination
  - to spread SQUID method and the experiences gained from its practical adoption to the international community of IT-practitioners and consultants,
- Exploitation
  - to establish a common product quality configuration applicable to transnational development and delivery in the common market.

The project started in early 1996 with the definition phase. This phase will end in mid 1996 and hopefully be followed by the demonstration phase.

4 Results

4.1 Results from Testing

- Several examples show that new software may be without significant problems.
- Elder software (even libraries) causes problems.
- Commercial hardware with excellent operational experience and references may cause problems in case of specified but unusual operational conditions, e.g. worst case voltage or unusual protocol situations.
- Assessment of new software for systems important to safety causes less problems than assessment of commercially available hardware. This can be quantified in number of topics needing clarification between producer and assessor.
- As a result to assess the hardware part of a computer-based safety system causes more effort (that means more time and money) than assessment of software — which is in contradiction to the usual assumption.
- For assessment of a new control system which is conceived for usage in safety systems as well, 852 documents have been presented, 633 for hardware, 219 for software.

4.2 Results from Operation

The operation experience of some computer-based systems important to safety in some European plants — mainly from Germany — is shown in the table below (see next page). Except the 'Test Aid for Neutron Flux 3' these systems have been assessed by us.

Testing of Computer-Based Systems — Methods, Tools, and Results

398
<table>
<thead>
<tr>
<th>System</th>
<th>Number of Systems</th>
<th>Operation</th>
<th>Number of Failures</th>
</tr>
</thead>
<tbody>
<tr>
<td>BWR Protection System</td>
<td>2v3, 2v2</td>
<td>1976 – 1982</td>
<td>0</td>
</tr>
<tr>
<td>Control Rod Control Computer</td>
<td>7x 1v1</td>
<td>1977 –</td>
<td>1</td>
</tr>
<tr>
<td>Radiation Monitoring 1</td>
<td>200</td>
<td>1986 –</td>
<td>0</td>
</tr>
<tr>
<td>Neutron Flux 1</td>
<td>100</td>
<td>1988 –</td>
<td>0</td>
</tr>
<tr>
<td>Radiation Monitoring 2</td>
<td>400</td>
<td>1986 –</td>
<td>3</td>
</tr>
<tr>
<td>Neutron Flux 2</td>
<td>3</td>
<td>1987 –</td>
<td>0</td>
</tr>
<tr>
<td>Neutron Flux 3</td>
<td>15</td>
<td>1994 –</td>
<td>0</td>
</tr>
<tr>
<td>Test Aid for Neutron Flux 3</td>
<td>1</td>
<td>1994 –</td>
<td>3...4</td>
</tr>
</tbody>
</table>

5 Conclusion

At the very beginning of our work on assessment and testing of computer-based systems important to safety we were not very successful in the qualification of such systems. A license for closed-loop operation could not be reached at that time. The development of methods and tools within the last 25 years has now led to the situation, that assessment and testing of safety-relevant computer systems, even reactor protection systems, has become reality. A type test of hardware and software components able to build a reactor control system or a reactor protection system as well, will be finished within the next few months. The chances for an application in a nuclear power plant are very good: orders already have been placed.

6 Literature

/Art85/ Arthur, L. J.; Measuring Programmer Productivity and Software Quality
John Wiley & Sons, New York, 1985

/Bow85/ Bowen, T. P., Wigle, G. B., Tsai, J. T.; Specification of Software Quality Attributes
Rome Air Development Center, New York, 1985

/Deu88/ Deutsch, M. S., Willis, R. R.; Software Quality Engineering
Prentice Hall, Englewood Cliffs, 1988

/Zus91/ Zuse, H.; Software Complexity, Measures and Methods
de Gruyter, Berlin, New York, 1991

We would like to thank Hartmann & Braun AG and Siemens AG for making available data about operation of computer-based systems important to safety and failures observed.
Validation of Transformation Tools

Johannes Brummer, Manfred Keraken
Arndt Lindner, Horst Miedl

Institut für Sicherheitstechnologie ISTec GmbH
Forschungsgelände
D-85748 Garching bei München

1 Introduction

In the course of the reorganisation and modification of the I&C equipment in nuclear power plants, the analogous technique is in many cases replaced by modern digital systems. This increasing use of digital technology, even in safety-critical applications, is on the one hand a consequence of the obsolescence of analog systems, on the other hand is based on the need to enhance the operational performance of I&C systems. Software based devices and systems offer several advantages, such as greater precision, higher availability due to self testing and the ability to implement complex functions.

Other characteristics of software driven systems, however, have raised safety and licensing issues and caused a world-wide discussion amongst experts concerning effective qualification procedures to enable the application of software in a safety critical environment. The general consensus, represented in the important guidelines and standards, is, that the quality of software is mainly based on a strictly ruled and methodically fundamented development and assessment process ("software life cycle"). There is the agreement that well-established methods and procedures of the up-to-date software engineering technology guarantee a level of software quality sufficient for safety critical applications.

Most of the standards and guidelines concentrate on requirements for newly developed application software. Some of the manufacturers of I&C systems, however, offer available software packages. The application specific programs are constructed by configuration of prepared generic software modules (elementary functions for a certain technical field). Besides economical reasons, this type of software solution has one essential quality related aspect: The construction (configuration) process is usually performed or at least supported by an automatic (code and system) generating tool and all error-prone (manual) coding activities are avoided.

Nevertheless, there is the question, how to achieve confidence in the correctness and reliability of such types of (configured) software, especially: how to qualify the configuration process respectively the "configurator" tool. In the following we will firstly characterise this
software type and point out the advantages and disadvantages in comparison to traditionally developed application software (section 2), we will also discuss the baselines for qualification of configurable software (section 3) and finally show the approach chosen for the qualification of the Siemens TELEPERM XS software environment (section 4).

2 Automated vs. manual software development

The goal of all software development processes is to design a product, that works on a class of well-defined hardware and fit the system requirements specification. In the past, a lot of methods were developed, to assist the software development process (Jackson, Meior&Ward, structured programming, modularization, object oriented programming, etc.). Another, different way uses automatic tools to transform special kinds of formal (e.g., graphical) specification into programming languages. Up to now, no method fits in all fields of software development.

Generally, software development methods can be classified as manual programming and automatic program generation.

![Software life cycle diagram](image)

Fig. 1: Software life cycle (development part)

Manual programming allows the design of all kinds of software. The development process splits into several steps (Fig. 1). The result of the development process depends strongly on
the experience and caution of human programmers. The methods to assist manual development processes give hints and guidance to formalise software development. Tests and proofs for correctness are necessary and must mostly be done by human beings. Besides errors in the development process, errors in the evaluation phase can occur.

An automatic code generation tool mostly fits only one application field. The result of the software development process depends strongly on the tool and is independent from the user of the tool, the human programmer, in a wide variety (Fig. 2). All generated programs follow the same programming style. Therefore, it is possible to create tools for program tests and correctness proofs, too.

Typical application fields for manual programming are system software like operating systems, I/O drivers, etc. On the other hand, typical application fields for code generation are classes of application programs, like software for safety critical applications in nuclear power plants (I&C functions).

Fig. 2: Software life cycle including software generation

Such software should be free of errors, which must be proved during the licensing procedure.

The two classes of software development processes lead to software products with different properties. The most important shortcomings of manual designed software consist in the number of transformation steps from the system specification to the program, which are performed by human beings. Each step is a potential source for introducing faults into the resulting software. Besides the high costs to implement the software, there is a huge effort needed for verification and validation of the product.
The automatically generated software will be highly formalised code. This is due to the necessary strictly formal procedure of code generation. It may be very hard to analyse such code manually. Nevertheless it seems to be possible to design an independent tool for this task.

The most important source of design errors in software is the transformation of the system specification to the software specification /BIS 88/. The front-end of this transformation is based on the view of plant engineers, whereas the back-end is based on the view of software engineers. Due to the different ‘cultural’ background of both parties and the resulting potential for misunderstanding, the transformation process from system to software specification is highly error prone.

Therefore it is desirable to develop system and software specification in an application specific precise notation (syntax) with well defined semantics. A precisely defined syntax and semantics also renders as a ‘by-product’ the opportunity to generate code automatically from a specification.

Thus the two factors in software development, i.e. precise (formal) and process-specific specification together with automatic code generation form a good basis to establish error-poor software. For safety-critical software, however, there is still a need to validate the code generation process.

3 Qualification of configurable software

3.1 Guidelines for the qualification of configurable software

Configurable software consists of pre-existent software modules (basic elements) which are configured to an application. If this application is an I&C function in a NPP, these basic elements are often graphical elements coming from ladder logic or functional diagrams. Normally they exist in the form of a graphical symbol and a piece of code in a programming language. Both possess a well-defined syntax and semantic if an appropriate symbolic and language are chosen. Also appropriate rules for combining the basic elements must be in place.

With all these elements provided, it is possible to establish formal specifications of I&C functions and to generate automatically code from this specification. For the qualification of I&C software which is developed in this way the problem areas of the V&V of

- formal methods
- automated transformation tools
- use of pre-existent software

are relevant.
The world-wide most recognised standard which sets requirements for software to be used in NPP's is IEC 880 /IEC 880/. With respect to these three topics, however, this standard contains only few, fairly general expressed requirements like the following:

- The use of a formal specification language may be a help to show coherence and completeness of the software functional requirements. Automatic tools may be used for this purpose (see IEC 880, 4.10).

- The use of automatic tools is recommended (see IEC 880, 5.2.6).

- As far as possible suitably qualified automatic development aids should be used (see IEC 880, B1.b.bi).

- Where standard software from a manufacturer or supplier is used, it should be shown to have operated satisfactorily (see IEC 880, B2.c).

Therefore a standard /IEC 94/ is in preparation which tackles, among others, these specific topics.

In the context of this paper, mainly requirements towards automated transformation steps during the development of software are of interest. The draft standard /IEC 94/ gives the following requirements for transformation tools.

(1) Where the output of a transformation tool is used without further review, the tool development shall conform to the same quality assurance requirements as for the development of the target software.

(2) For transformation operations, certified and validated tools should be used. When an expression of requirements is prepared or transformed by a tool, the intermediate expressions should be verified by inspection or use of a verification tool. The final output of the transformation process will be executable code or configuration data for use with that code.

(3) For all Category A (highest safety category) systems the final output of a transformation tool shall be subjected to verification and validation in accordance with IEC 880. If the final output cannot be subjected to verification and validation for some functions, then the transformation process itself shall be validated to IEC 880, and the tools used should be prepared to the processes of IEC 880. The use of a transformation tool without further review of its output should not be accepted unless special justification is provided.

(4) For all Category B (lower safety categories than A) systems the final output of a transformation tool should be subject to validation to IEC 880. The output of a tool may be used with validation of representative samples of output, if the tool has been previously validated or is validated.
If the transformation tool is considered to produce software of the highest safety category, i.e. category A according to the standard IEC 1226 /IEC 937, the most stringent requirements are contained in the third clause. Figure 3 illustrates these requirements.

Fig. 3: Qualification of transformation tools for category A systems
The first requirement is to subject the final output of a transformation tool, e.g. the source code, to V&V in accordance with IEC 680. In many cases this will cause problems, because normally the transformation process skips one or more development steps against which V&V of the output would be performed (as required by IEC 880). Therefore the first requirement of the third clause can be replaced by two others, i.e. to develop the transformation tool according to IEC 880 and to validate the transformation process itself.

3.2 Main steps in the qualification procedure

For the transformation a set of well-defined rules must be followed, and there is a temptation to take advantage of these rules when validating the transformation process, e.g. by some kind of inversion of these rules. With this approach, however, a certain danger for common cause failure in the forward (transformation) and backward (retransformation) process is introduced. Therefore the retransformation process as part of the validation must be performed independently from the transformation process. The results of the retransformation have to be brought into a form which enables its comparison with the input representation (the specification) of the I&C function, thus demonstrating its equivalence, or otherwise.

Of course there is also a need to verify the correctness of the coded basic elements from which an I&C function is constituted. As these basic elements are normally very simple ones, e.g. logical functions (AND, OR, ...), choice of a minimum out of few values, adder, etc. their correctness can be proven by means of rigorous mathematical proof or exhaustive testing.

A transformation tool is meant to be widely used to produce I&C functions. Therefore it is useful to incorporate the validation of the transformation process into the tool, thus enabling immediate validation of I&C functions generated.

4 Example: The TELEPERM XS code generation process and its qualification

4.1 Overview

TELEPERM XS is a software package, designed for applications important to safety in nuclear power plants by SIEMENS/KWU. The software package consists of several components (Fig. 4) /HOF 94/.

The real time operating system, I/O drivers, etc. depend strongly on the hardware used. These components are prefabricated standard software (developed according to IEC 880). The application software is hardware independent to a great extend. This part of the software is generated for each application.
Fig 4: Software package TELEPERM XS

The real time environment schedules up to two functional diagram groups, which consist of one or more functional diagrams. The function blocks are the primitives of the application software. They carry out simple logical (AND, OR, etc.) and algebraic (ADD, MUL, DIV, etc.) operations, threshold triggering, flip-flops, m-out-of-n voting, but also more sophisticated tasks like PID controllers. There are approximately 75 function blocks available.

The functional diagrams are designed using a graphical specification, which is transformed to the source code automatically [GRA 94].

Fig 5: Graphical software specification
The graphical specification based on the SPACE-tool (SPECification and Coding Environment). This tool comprises a graphical editor, a database which holds all specification data and, a code generator.

Verification and validation of the small software primitives has been performed exhaustively by developers and independent assessors. The generated code is strongly formalised and linear. Most important for the correctness of the code is the correct sequence of function calls (calls of function blocks) and the correct reference to data (signals, parameters, state memories).

4.2 Retransformation Procedure

The basic idea is to analyse and process the generated source code to reconstruct the inherent operation of the underlying safety function in a form suitable to prove the functional equivalence with the SPACE database tables containing its specification [MID 96].

This is done by means of the standard UNIX tools lex (lexical analyser) and yacc (parser generator). Essentially they extract the

- function block identifiers,
- their calling sequence and
- their parameters (e.g. input signals, ranges, ...) which describe the connections within the functional diagram, i.e. connections between function blocks or connections from or to other functional diagrams.

There is of course further data to be considered like number and type of parameters, or version information in order to examine whether the version of the source code concurs with that of the actual database.

The entire information is stored in files in a form which supports the remaining work of the tool. In the next step this information is compared with the data from the SPACE database tables containing the specification of the functional diagram. The access to the database is realised with the aid of the database language ESQL (embedded SQL).

If the comparison yields any discrepancies between the generated source code and its specification, then the tool has to analyse and record these accordingly.

For the analysis of the comparison results the entire stored source code information is considered so that the tool can provide an analysis trace. The comparison results are both printed on screen and written to a file. The tool user can select between different levels of detail.
Example:

The following example illustrates in a simplified manner the operation of the tool.

Let the following diagram be the specification of a "safety function":

![Functional Diagram](image)

**Fig 6: Example of a functional diagram**

There are three function blocks, an "adder", a "minimum" and a "constant". Signal 1 and signal 2 are added to signal 3. Signal 3 and signal 4 are compared and the minimum is assigned to signal 5 if the minimum is greater than the constant. Otherwise constant is assigned to signal 5.

The generated source code of the above functional diagram looks like that:

```plaintext
struct 1 = {signal 1, signal 2, signal 3};
struct 2 = {constant};
struct 3 = {signal 3, signal 4, signal 5};
add (struct 1);
const (struct 2);
min (struct 3);
```

For each function block the parameters are packed into a structure. After that the function blocks are called with the respective structure as argument.

The tool analyses the source code and extracts at first the identifiers and the calling sequence of the function blocks which is a relatively straightforward task. More complicated is to "break up" the data structures although in this example it seems quite simple. Since the tool needs to "understand" every structure component, i.e. whether it is an input or output signal, what type of signal it is, etc.
The extracted information then has to be compared with the related database tables. The following diagram illustrates the structure of a data record for a connection.

<table>
<thead>
<tr>
<th>functional diagram identifier</th>
<th>local identifier</th>
<th>source type</th>
<th>source identifier</th>
<th>target type</th>
<th>target identifier</th>
<th>signal type</th>
</tr>
</thead>
</table>

In our example the connection between signal 1 and the function block 1 (add) looks like

| 1 | 1 | external signal | signal 1 | function block | add | analogue signal |

where the functional diagram is assigned the identifier 1.

A main problem was to match the information extracted from code with the database entries and vice versa. This was due to the basic concept of the tool to be independent of the rules of the code generator. The tool therefore does not invert the transformation rules from database entries to source code. Rather it works with the description of the source code structure and the database model.

5 Summary

I&C functions in nuclear power plants are implemented increasingly by software using tool supported graphical specifications and automated code generation. The correctness of the specification must be proven by collaborative efforts of plant and I&C engineers.

This paper concentrates on the qualification of the code generating process. Existing requirements in the relevant standards are interpreted towards techniques which can be used during qualification. An example is given of an automatic production process and its validation by means of a tool which is existent as a prototype at the time being.

6 Literature

/BIS 88/  Bishop, P., G.  


IEC 93/ IEC Publication 1226, Nuclear power plants - Instrumentation and control systems important to safety - Classification, 1993.


OECD/NEA CSNI-CNRA INTERNATIONAL WORKSHOP on LICENSING ISSUES of COMPUTER-BASED SYSTEMS IMPORTANT to SAFETY

Munich, GERMANY 5 - 7 March 1996

Tool validation, maintenance and operational factors of digital based reactor protection system for next stage PWR plants in Japan

Y. Yamamoto Kansai Electric Power Co., Inc.
T. Aoyagi Japan Atomic Power Co.
M. Utsumi Mitsubishi Heavy Industries, Ltd.
* H. Funakoshi Mitsubishi Electric Corporation
1. Introduction

Tool function to support development of software, which is important to secure safety of nuclear power plants (e.g. environment setting, programming, compiler etc. mentioned as engineering tools below) is configured, based on operating experience of existing plants, reflecting requests for various improvements in its operation and recent drastically advancing technology. In software of the safety grade instrumentation system, the engineering tools serve as an important system to be applied to design and production of the safety grade software and its maintenance during plant operation. Therefore, the engineering tools themselves or the way of their use require a sufficient reliability. This chapter explains about

(a) Verification and validation of the engineering tools
(b) Maintenance and operation management of the engineering tools

2. Verification and validation of the engineering tools

In the safety grade instrumentation system, two types of software are employed; basic software specialized for nuclear power plants, introduced in the previous chapter, and application software which is activated periodically by the basic software. The combination of these types of software has already have a sufficient operating experience in the non-safety grade instrumentation system of existing plants in Japan. As far as the safety grade instrumentation system is concerned, each function is implemented by combining some application software, for which easily-debugged visible programming language (POL; Problem Oriented Language) is employed to ensure reliability.

This section explains about the method of verification and validation of the engineering tools specialized for design and production of application software of the safety grade instrumentation system.

i) Function of the engineering tools
The engineering tools are equipped with supporting functions related with design, production and maintenance of application software as showed in the Fig-1.
Fig. 1 Function of the Engineering Tools

1) Standard POL Module
   A series of standard sub-routine process codes, such as AND, OR logic, which are activated by application software.

2) POL input and output interface definition

3) Application software compilation
   Function to generate object codes operated under the instrumentation system.

4) Visual input and display of software
   Function to realize coding of application software combining pictorial elements prepared for each standard POL Module such as AND, OR logic.

5) On-line display of operating conditions
   Function to display operating conditions of application software on-line

ii) Verification and Validation of the engineering tools

The engineering tools have a sufficiently reliable experience in application to the non-safety grade instrumentation system of existing plants in Japan. Thus, basically there will not be any problems in application to the safety grade instrumentation system as well.

Following is the explanation of verification and validation method of the engineering tools through generation, verification and validation process of application software.

1) Verification and validation of software configuration
   Standard POL Module employed for manufacturing application software consists of a minimum calculating element which is impossible to divide, such as AND, OR logic, addition and subtraction. Each POL Module is manufactured by easy program. Verification
and validation of the modules in relation with software configuration is realized by checking source codes and object codes after compilation.

② Verification and validation of software function
The engineering tools' function of manufacturing application software consists of calculating function of standard POL and combining function of each POL Module. These functions are verified along with the verification of manufactured application software regarding object codes. Concerning application software, the object codes are equipped to the safety grade system (the target machine) as explained above and simulated data which exercises all the pass of software is input to exercise all the pass of software is input to verify and validate.

Through the steps ① and ② as explained above, verification and validation of the engineering tools function as well as application software function is completed.

Fig. -2 Concept of V&V
3. Maintenance and operation management of the engineering tools

Computer based system of nuclear power plants (safety /non-safety grade instrumentation system) is expected to undergo several improvements and changes during plant operation life cycle. As safety application software is verified and validated in each improvement, therefore, there will be no problem regarding assurance of plant safety. However, the engineering tools still play an important role in efficient improvement procedure and changes well as in safety control. It is considered that the engineering tools have to be equipped especially with the two functions below to eliminate positively human errors.

i) Protection against misuse and illegal invasion of software during maintenance.

ii) Standardization of strict resetting procedures and software installing (and management) procedures to prevent errors in operation.

This section explains basic policy about software maintenance and management procedure by the engineering tools.

① Prevention of human errors by standardization of the tool operating procedures

In maintenance and examination at site, such as improvements or changes of software, operation mistakes influence plant operation directly. Therefore, some measures need to be taken to prevent human errors such as misuse of tools as much as possible. As for the engineering tools, operations in design, production, maintenance and examination at site are standardized to prevent simple human errors such as operation mistakes.

---Standardization of operation procedures of the tools
---Management function to maintain conformity of software versions

② Prevention of software changes out of control

According to changes of plant characteristics, software tuning is sometimes made at site. In such a case, software changes which the operator does not expect are liable to occur. Besides, there will be a possibility that software is changed to the one which would impair plant safety by an unqualified operator (including a person to do that on ill-intention). Therefore, the engineering tools support application software change under control with the functions.
mentioned below.

--- Function to record automatically operation history at site
--- Function to limit tool users by a password

The software which the safety/the non-safety grade instrumentation system is equipped with, is operated under ROM to impose more strict limitation on out-of-control changes.

3 Prevention of mistakes in resetting procedure

There is a demand for the function to easily confirm implementation of operations in the process of design, production and maintenance, examination at site. The engineering tools are equipped with the functions below to ensure completion of system resetting procedure.

--- Function to display visually application software (on-line display of operation conditions)
--- Function to control procedures in each operation step (display of alarm in operation mistake)

As explained above, the engineering tools implement efficient improvements and changes with their functions ①～③. Besides, the tools play an important role in safety management. Considering the tool operation, they can be applied without any problems to improvements of the safety grade system during plant operation life cycle, by double checking with manuals resulted from the experience in the non-safety grade instrumentation system of existing plants.
OPERATING AND MAINTENANCE EXPERIENCE
WITH COMPUTER-BASED SYSTEMS

HELMY RAGHEB
Atomic Energy Control Board, Canada

1. Introduction:

Continued expansion of digital technology in nuclear power reactor has resulted in new licensing issues, since the existing licensing review criteria were mainly based on the analog devices used when the plants were designed. On the industry side, a consensus approach is needed to help stabilize and standardize the treatment of digital upgrades while ensuring safety and reliability. On the regulatory side, new guidelines and regulatory requirements are needed to assess digital upgrades.

Upgrades or new installation issues always involve potential for system failures. They are addressed specifically in the "hazard" or "failure" analysis, and it is in this context that they ultimately are resolved in the design and addressed in licensing. Failure Analysis is normally performed in parallel with the design, verification and validation (V&V), and implementation activities of the upgrades. Current standards and guidelines in France¹, U.S.² and Canada³ recognize the importance of failure analysis in computer-based system design. Thus failure analysis is an integral part the design and implementation process and is aimed at evaluating potential failure modes and cause of system failures. In this context, it is essential to define "System" as the plant system affected by the upgrade, not the "Computer" system. The identified failures would provide input to the design process in the form of design requirements or design changes for the new installation or the upgrade. Procedures, training, and other practices may be affected by the failure analysis because administration controls, periodic calibration and surveillance procedures may all be used to provide defense against potential failures.

An important input to the failure analysis activities comes from the feedback of operating and maintenance experience. Feedback of operating experience in nuclear power plants has long been recognised as a valuable source for improving system design, procedures or human performance to achieve safety and to prevent recurrence of failures. This is particularly true in the case of complex systems such as computer-based systems. The process of feedback would provide designers with information on systems failures, unforseen scenarios, or unanalyzed configurations.

The review of operating experience and the identification of causes of failures is also essential for the regulators in performing their safety assessments. Currently, the NRC reviews the electromagnetic compatibility (EMC), software reliability, and the human-machine interface when it performs a safety evaluation of digital upgrades to ensure that the digital system failures resulting from the identified causes are within the acceptable level of a system’s reliability.
Operating experience with computer-based system is one of the topics raised in the SESAR report. The CSNI Bureau of the OECD has requested NEA Working Group No.1 (PWG1) to review this topic. A task group led by Canada was therefore formed within PWG1, including France, Japan, U.K., and U.S.A. The study of the task group has just begun, and in the interest of providing input to this workshop, the group have expedited their initial review, and decided to present here some preliminary observations.

The purpose of this paper is to summarize the initial observations and some preliminary findings related to the operating and maintenance experience, based on the initial contributions from France, U.S.A. and Canada. Additional information from the review of the open literature is also included. Two of the operational incidents, selected as case studies, one from the U.S. and the other from Canada are presented as examples in Appendix 1. In addition, this paper presents an example of an evaluation by the USNRC of the digital upgrades introduced at an operating nuclear power plant.

In the U.S.A., the nuclear industry and the NRC have performed numerous studies of digital systems to identify safety concerns and to minimize the failures of Computer-base systems. These studies were in response to the replacement of analog instrumentation and control (I&C) systems with computer-based digital I&C systems by the utilities. Reference1 describes the results of one of these studies on computer-based digital system failures and evaluates the NRC's review of analog-to-digital conversions.

The U.S. study focussed on the current operating experience of computer-based systems in the U.S. nuclear industry as reported to the NRC. The purposes of the study were (1) to identify the types of digital system failures and (2) to ascertain how the NRC reviews digital updates. The first purpose involves reviewing LERs involving digital failure events experienced in 1990-1993, and categorizing digital system failures. The second purpose involves reviewing SERS for analog-to-digital upgrades, one for a General Electric plant and one for a Westinghouse plant. The review of 79 applicable LERs resulted in four categories of failures: software error, human-machine interface, Electromagnetic Interference (EMI), and random component failure. Software errors were further divided into software V&V failures and configuration control failures. A description of each category follows in the analysis below.

In Canada, an in-depth review of the Canadian operating experience with CANDU computer-based systems was initiated to address the safety issues arising from the use of these systems5. This review was based on the experience collected from the Atomic Energy Control Board's (ABCB's) analyses and reviews of CANDU significant events over the past 13 years. The review of 459 significant event reports from 22 reactor units, related to computer-based control, monitoring and safety-ystems, was undertaken to identify the types of failures encountered to date, the lessons learned through operating experience, the measures taken to address known software/hardware weaknesses, the impact of human interaction on software/hardware performance and the effectiveness of past computer expansion, upgrade and replacement programs.
While the review covered a relatively large number of computer related events, it should be recognized that they do not necessarily include all computer failures. They include only those events which resulted in consequences that meet reporting criteria of either the AECB or the utility. Most hardware, for example, is duplicated, and a single failure of a redundant component generally would not be reported formally to the AECB but would be reported internally in the utility.

In France, the experience of IPSN in evaluating the safety submissions from EDF related to the assessment of C&I systems were reviewed\(^5\). The review presented the methodologies used, including approaches for the identification of failures and management of modifications.

2. Observations:

The U.S. study\(^4\) found two results. First, electromagnetic interference (EMI), human-machine interface error, and software error caused significant number of digital system failures during the period 1990 through 1993. Fewer failures were caused by random component failures. Second, the NRC reviews digital I&C systems for electromagnetic compatibility (EMC), software reliability, and human-machine interface when it performs a safety evaluation of digital upgrades submitted by a licensee.

Table 1 lists the total number of events by category. The table shows that software errors (30 failures), human-machine interface errors (25 failures), and EMI (15 failures) are the dominating causes of the digital system failure events. However, only 9 events were caused by random component failure.

<table>
<thead>
<tr>
<th>Cause of Events</th>
<th>Number of Events</th>
</tr>
</thead>
<tbody>
<tr>
<td>Software error</td>
<td>30</td>
</tr>
<tr>
<td>Human-machine interface error</td>
<td>25</td>
</tr>
<tr>
<td>Electromagnetic interference</td>
<td>15</td>
</tr>
<tr>
<td>Random component failure</td>
<td>9</td>
</tr>
</tbody>
</table>

The evaluation of the U.S. data found that software failures, human-machine interface errors, and EMI caused more than 89 percent of the digital system failure events. The root causes of these failures were (1) poor software V&V, (2) inadequate plant procedures, and (3) inadequate EMC of the digital system for its environment. Most of these failure events did not cause a significant safety event; however, these failures could cause common-mode or cause failure, which can lead to significant safety events.
The ABCB study in Canada is not complete. Preliminary results are shown in Table 2. The following observations are based on the data analyzed to date, which is regarded as incomplete at the time of writing:

a. Almost all trends, in the investigated failures were either decreasing or they were flat, except for those attributable to inappropriate human actions and jumper-related faults. The data indicate that the incidents of inappropriate human actions has shown a marked increase in the last five years.

b. Software faults are still decreasing, which is an indication of the presence of latent faults left over from development, and not yet discovered.

c. Most failures were associated with the Digital Control Computers (DCCs); this is because most of the computer related events deal with DCCs. The DCCs have been in use since the 1970's and perform a complex and continuous task, so that software failures would tend to be more prevalent.

d. The control and shutdown computers are designed to be fault tolerant and, on their own, cause few failures due to hardware. However, the plant hardware associated with the computers is extensive and many events that involve computers are caused by ancillary device failures. Hardware failures should therefore be assessed not only from the point of view of unavailability but also in terms of their impact on the software and the resulting consequences.

e. Software problems are sometimes corrected with a temporary change installed outside the programmable part of the software, often referred to as “patch”. Even if put in correctly, the patch appears to cause further problems.

f. Programmable logic controllers (PLCs) are being introduced as a cost-effective method of replacing older analogue or digital controls. PLCs have resulted in a number of incidents within the plants and it must be recognized that they are themselves digital computers. The hardware and software for PLCs should be subjected to the same controls as with the DCC and the shutdown system (SDS) computers.
<table>
<thead>
<tr>
<th>HARDWARE PROBLEMS</th>
<th>SOFTWARE PROBLEMS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Processor</td>
<td>Executive/Os</td>
</tr>
<tr>
<td>Memory</td>
<td>Application</td>
</tr>
<tr>
<td>Interface CCA</td>
<td>Database/Table</td>
</tr>
<tr>
<td>Internal PSU</td>
<td>IO Routine</td>
</tr>
<tr>
<td>Connection</td>
<td>Software Other</td>
</tr>
<tr>
<td>Ancillary</td>
<td></td>
</tr>
<tr>
<td>Peripheral</td>
<td></td>
</tr>
<tr>
<td>Hardware Other</td>
<td></td>
</tr>
<tr>
<td>HUMAN-MACHINE INTERFACE PROBLEMS</td>
<td>EXTERNAL</td>
</tr>
<tr>
<td>Inappropriate Human Action</td>
<td>External Power</td>
</tr>
<tr>
<td>Operating Manual</td>
<td>EMI</td>
</tr>
<tr>
<td>Bad Communications</td>
<td>Other</td>
</tr>
<tr>
<td>Procedure Other</td>
<td></td>
</tr>
<tr>
<td>UNASSIGNED</td>
<td>MECHANISM OF THE FAILURE</td>
</tr>
<tr>
<td>COMPUTER SYSTEM</td>
<td>Requirements</td>
</tr>
<tr>
<td>DDC</td>
<td>Design Error</td>
</tr>
<tr>
<td>SDS1</td>
<td>Coding Error</td>
</tr>
<tr>
<td>SDS2</td>
<td>Manufacturing Error</td>
</tr>
<tr>
<td>SDS - Both</td>
<td>Part Failure</td>
</tr>
<tr>
<td>FM</td>
<td>Workmanship</td>
</tr>
<tr>
<td>SORO</td>
<td>Other</td>
</tr>
<tr>
<td>PLC</td>
<td></td>
</tr>
<tr>
<td>Computer - Other</td>
<td></td>
</tr>
<tr>
<td>EFFECT ON THE PLANT</td>
<td>STATUS OF THE PLANT</td>
</tr>
<tr>
<td>Service Interruption</td>
<td>Steady State</td>
</tr>
<tr>
<td>Service Degraded</td>
<td>Transient</td>
</tr>
<tr>
<td>Inconvenient, not def</td>
<td>Maintenance</td>
</tr>
<tr>
<td>Minor, deferrable</td>
<td>Commissioning</td>
</tr>
<tr>
<td>Safety Related</td>
<td>Upgrade</td>
</tr>
<tr>
<td>Not Specified</td>
<td>Run Up</td>
</tr>
<tr>
<td>CORRECTIVE ACTION TAKEN</td>
<td>Shutdown</td>
</tr>
<tr>
<td>Replace</td>
<td>Office</td>
</tr>
<tr>
<td>Redesign</td>
<td>Not Specified</td>
</tr>
<tr>
<td>Restart</td>
<td></td>
</tr>
<tr>
<td>Rerain</td>
<td>Jumper in effect</td>
</tr>
<tr>
<td>Revise</td>
<td></td>
</tr>
<tr>
<td>Not Specified</td>
<td></td>
</tr>
</tbody>
</table>

**Table 2**

Total records in survey 459
3. Operating Experience

**Computer Hardware and Ancillaries:**

In the Canadian study of Ref. 5, the computer hardware, consisting of memory, processor, interface circuit cards, internal power supply unit (PSU), and peripherals, shows a lower failure rate relative to other leading categories, most likely because of the design of the equipment, which is fully duplicated and, therefore, is fault tolerant. In other words, a single hardware fault should not affect the operation of the computer system or the plant. It was also found that the number of computer hardware faults has decreased steadily with time (Fig. 1).

The most common hardware problem was associated with ancillary devices. This category refers to parts and sensors outside of the computer 'box'. For example, common elements in this category are sensor and relay failures. Given the nature and complexity of some of the sensors used in nuclear power plants, this finding is not unexpected. Even if utilities did not use digital computers, these sensors or their equivalent would still be required to monitor the state of the plant. Ancillary failures showed also a decreasing trend with time (Fig. 2).

The second highest failures within the hardware are the faults with the circuit card assemblies (CCAs), connections and peripherals (see Table 2). Problems with CCAs and the decreasing trend of hardware faults over time agrees with the observed experience of the CANDU plant, Wolsong-1, operating in Korea for more than 10 years. Ref. 6 reported that computer systems encountered a high number of problems in the early stage of operation. Problems arose from oversights in design, manufacturing defects or installation. The majority of failures that have occurred on the control computer systems were attributed to the input-output systems. Bulk Memory System also at Wolsong-1, (Fixed Head Discs) experienced, in the mid 1980's, major failures and were subsequently replaced with dynamic random access memory units. On a restart, all of the core memory is saved on the disc and a fresh version is transferred to core. The saved data is then used to analyze the cause of the initial stall.

Many of the reported printer failures are caused by mechanical failures. Although printers may be viewed as one of the computer peripherals, their failures could cause computer stalls leading to major reactor transients. A recent event occurred at Bruce-A unit 4, where jamming of a computer printer ribbon caused its buffer to fill and stop the execution of a program. This caused the control computer to stall and close the cooling flow supply valves to the fuel tube. The event prompted recommendations for software and hardware changes to
ensure cooling flow to the fuel is maintained at all times in case of computer stalls.

Computer Software:

The Canadian study of Ref.\textsuperscript{4} reported that software faults represent a significant number of failures. The number of computer software faults has been decreasing slightly with time. A larger decrease in the number of failures can be expected in this area (Fig. 3), because once the fault is rectified, it is permanently rectified, i.e., it should not experience wear-out trends.

The U.S. study\textsuperscript{4} categorized any event caused by software failure as a software error. Each software error was further categorized as either (1) a software verification and validation (V\&V) failure or (2) a configuration control error. Configuration control is the process by which changes to the products of software development are controlled, including the configuration baseline.

The U.S. study found that software errors is the largest failure type among computer-based system failures (Table 1). Failures in the software V\&V process caused most of the software error events. A similar observation is made in the Canadian study. Table 2 shows that failures corresponding to the V\&V failures, which were further broken down into categories such as "requirement", "coding error" and "design error" , represent a high percentage of the software failures.

Human Machine Interface:

The definition adopted in the U.S. study (Ref.\textsuperscript{4}) states that human-Machine Interface includes all interfaces between the digital system and plant personnel, including:

Operators - alarms, status displays, and control interfaces.

Maintenance technicians - test and calibrations interface, diagnostic information displays, and data entry terminals for setpoints.

Engineering personnel - configuration workstations and terminals

Human-machine interfaces also include unauthorized computer data entry, deviation from procedures, and inadequate procedures from plant personnel.

For the purpose of presenting observations in this paper from different participants in this study, this broad definition will be adopted.

Both the U.S. study and the Canadian study indicate that the inappropriate human actions contributed to about 25\% of the computer-based system events. Preliminary results from the Canadian study indicate an increase of these events in the last five years (Fig.4)
External Power:

The Canadian study has shown that problems with external power supplies represent a not-insignificant number of failures, despite the fact that power supplies are designed to be fully duplicated and uninterruptable. However, they decreased with time as deficiencies were removed from the systems (Fig.5).

Environmental Conditions:

Environmental conditions such as temperature and humidity are critical for the proper functioning of computer components. At Wolsong-1, high failure rate was reported on the integrated circuits of the shutdown computers, for many years. Failures were reduced by providing heat sinks to the IC's, installing air conditioner, and making holes in the cabinet. In addition, alarms were installed to draw the attention of the operators to the high temperature condition.

Within the operating environment of the plant, systems may produce random electrical noise known as electromagnetic interference (EMI). Digital equipment, which operates at higher speeds and lower voltages than the analog equipment it replaces, is specially vulnerable to EMI. The EMI was the cause of 3 events in the Canadian study and 15 events in the U.S. study. The relatively high number of events in the latter appears to be due to the broader definition of EMI in the U.S. study which includes poor grounding and poor connections.

The ability of equipment to function satisfactorily in its electromagnetic environment without introducing intolerable disturbances to that environment or to other equipment is known as the electromagnetic compatibility (EMC). In reviewing EMC, the USNRC requires a licensee to perform tests and measurements to demonstrate that the replacement digital system is qualified for its environment.

4. Maintenance Experience

Maintainability of computer-based systems depends largely on the quality factors of traceability, completeness, consistency, simplicity, modularity and testability. The maintainer needs to be able to fully understand the software before it can be changed. Improving upon the deficient quality factors is required in order to improve maintainability. Experience and lessons learned in improving these quality factors through the adherence to standards, implementing modifications, or improving documentation are discussed below.
Adherence to Standards:

A recent rehabilitation program, described in Ref.\(^7\), recognized the need for the software to conform to more stringent software quality assurance standards. The Canadian approach used in the reactor rehabilitation program, to achieve this, is to use the new Ontario Hydro/ABCL Software Engineering standards (OASBS)\(^2\). This is a family of standards where each individual standard corresponds to a defined level of nuclear safety. For example, Category II is used for safety related software such as the Digital Control Computers (DDCs) software changes and the category III standard is used for the Plant Display Systems (PDS) and Safety System Monitoring Computers (SSMC) design. In complying with the standards the utility found the following problems:

- it makes the up-front cost appears to be higher and the development duration longer.
- in attempting to categorize software, guidelines were found to be subject to interpretation; different people may reach different conclusions
- more and better knowledge of plant operation is required in order to apply the guidelines than what most computer staff possess.

Maintain or Replace?

Decisions taken by the utilities on whether to maintain or replace components or systems are based on the operating experience throughout the life of the plant. When opting for replacement, two problems may arise, which could impact on the quality of the procedures:

- Off-the-shelf products my not be fully adequate, which results in complicated procedures. This is in contrast to previous computer projects, where automated tools were not used, but the procedures were much more straightforward.
- Computer hardware and software lifecycles are becoming increasingly shorter, in the order of 2-3 years. This creates a problem for maintainers since they have to follow a methodical and slow process.

Procedures for Modifications:

Configuration control has been widely recognized and identified as a crucial area. In France, EDF considered it necessary to institute a general modification procedure and implement it before fuel loading, as well as specific rules which must be put in place to control modifications after fuel load\(^4\). The aim is to make sure that each change (be it error correction or evolution to specification) is carried out properly avoiding unnecessary destabilization and the risk of regression. In order to maintain confidence in the updated version, the adequacy of the requalification should be justifiable. This is particularly relevant when the possibilities for on-line testing are reduced once the reactor has gone critical.

IPSN recommended a formal impact analysis procedure which the licensee implemented. Each modification is considered on a case by case basis by a panel with diverse and independent interests (programmer, operator, etc.). The impact of the correction or not for each error or modification request is considered and supported from the points of view of safety, functionality, operator nuisance, probability of occurrence, diagnosis, implementation and validation. There are categorizations made concerning the gravity of a non-corrected anomaly and an attempt to quantify globally the impact of corrections made to a new version by indicating the number of software modules altered. A summary of the impact analysis is
presented to the safety authority. The utility consider it essential to have a rigorous approach which is accountable and visible to the safety authority. The onus is on the licensee to make it work.

In France, the process for modification management up to this date is relevant to pre fuel-load. Once the reactor is loaded and has gone critical, the implications of any modification are evidently to be taken very seriously. In the first instance, it is clearly the aim of everyone concerned that the C&I will be sufficiently validated and proven such that modifications will not be necessary between periods of reactor shutdown and fuel unload. However, to be realistic and cover all contingencies, a procedure for safe reinstallation of software has been prepared. The safety case relies ultimately on the ability to operate the reactor safely from the diversified back-up panel. There is also the possibility to reinstall a known proven version if ever a modification does not work out.

Although an update of the software is not normally envisaged during reactor operation, the plant configuration data, the "programmable" part of the system, is designed to be modified on line. This is considered as a function of the system and as such has necessitated a particular effort for validation to ensure the absence of interaction between data and system.

The validation of data changes are carried out upstream in the CAD. However, the conditions for data modification during the reactor operation must be subject to prior function analysis to identify plant availability and define special operating conditions. A dry run must be carried out on an off-line configuration before each data modification.

**Patching:**

The Canadian study\(^5\) examined the difficulties encountered with temporary modifications to the software known as "Patching" which is a modification to the software performed outside the program, eg. to a data table, that causes the bypass of a sequence performed by the program. Because patches are temporary; their installation and removal is controlled by administrative procedures using a record known as a "jumper". A recent serious reactor incident at Bruce-A, in 1992 [IRS report No. 1360.00] highlighted the risk associated with patching and with the inadequate control of the software "patching" practice. In this incident, the patch was installed to force the software to operate correctly at a very low reactor power as some power sensors become irrational during reactor start-up. Because the patch was not removed when the reactor power increased the software operated incorrectly and caused a power excursion that was terminated by a reactor trip.

The number of events which involved temporary software modifications or 'jumpers' was found to decline until the late 1980s, but these are now on the increase (Fig.6). Care should be taken in drawing conclusions in this area, because of the relatively small number of failures in some years, and the random pattern of the faults.

**Human and Organisational Factors:**

The use of human factors engineering (HFE)
Principles is becoming increasingly important in the software design and upgrade processes. Utilities in Canada incorporate the HPB program and also solicit significant input for the new operator interface from the operating staff7. An example of this is the process used for capturing the plant display system (PDS) design requirements by directly involving operators.

In France, involving future user staff in certain design tasks is also viewed by EDF as necessary7. Users would be part of a team that writes the operating data, design images and charts, proposes computing algorithm or prepare training documents.

**Design Authority and Staff Qualification:**

Production pressure may cause the utilities to rely on the expertise of their local technical staff in the plant, rather than the original designer or the design authority, to modify the software. The design authority is the entity responsible for recommending changes to the design and maintaining historical records of the design changes and the subsequent verification and testing activities. A dedicated and appropriately qualified staff should facilitate the traceability of changes and minimize the likelihood of software errors.

**Testing the Upgrades**

Experience has shown that, following software changes, formal testing should be performed in adherence to stringent standards, similar to the standards required in the development phase. These tests should be as automated as possible. The test procedures should be written such that the specific steps to be done by the tester and the expected results are completely documented. One of the root causes of the incident which occurred at Bruce -A in 1990 [IRS No. 1118.00 ] (see summary of the event in Appendix1, case study 2), involving unintended movement of the fueling machine while it was attached to the reactor face, was that the software error was not evident during testing.

Another important aspect is the development of test facilities7. Although development tools are very important, maintenance staff believe that testing tools have often not been given sufficient attention on many computer upgrade projects. One plant reported the development of a process I/O emulator and plant simulator (IOBS) testing tool for the project.

**Documentation:**

In contemplating major software changes, compliance with modern standards may make the volume and effort of documentation appear far greater than expected7. However, maintenance experience indicated that adequate documentation is essential in reliably performing changes to the software and in performing thorough evaluation9. Because modern software standards were not in common use in the mid-1970's there were no software requirements listed in the currently used software which was installed then. The software requirements should specify what the software is functionally required to do and these software requirements form the basis from which thorough tests can be designed. The software requirements are also needed for assessing software quality through measures such as traceability and testability. Other information which was found necessary to perform changes was the description of interfaces between hardware and software, such as where certain inputs can be read by the software. Examples of inadequate software design documentation are:

- no revision number,
- variable listed as coming from the wrong routine,
- a routine does not list any outputs but it was found that another routine indicates that it
  gets a signal from it,
- a symbol is not listed as being used when in fact it is used for control purpose,
- routines not described,
- a function which has been removed long ago is still described in the design
  documentation, or
- documentation describing the overall software design and interactions of control
  programs are not up-to-date.

Without up-to-date documentation, the process of configuration control may be managed
mainly through the skill and experience of the technical staff. This could also cause testing to
be done in an ad hoc manner, often relying on the tester to know how to stimulate the
software program and what responses to expect. Following the implementation of the change
it is equally important that the step-by-step test procedures and the expected responses be
documented. It was also recommended that event reports should be made traceable, i.e., any
changes made to the software which are the result of a problem identified in an event report
should reference the event report number and that the event follow-up report should reference
the software change.

When the decision is made to maintain obsolete equipment rather than replacing it, it was
found that the success of maintenance was attributed to the availability of good
documentation. At the CANDU plant KANUPP documentation proved invaluable in
performing maintenance, despite the presence of new staff.

5. Regulatory Evaluation

The NRC reviews EMC, software reliability, and human-machine interface issues when it
performs a safety evaluation on each digital upgrade submitted by a licensee. The NRC
evaluated the licensee's proposal to replace 7100 analog process protection system with a
computer-based digital process protection system. The digital issues that the NRC staff
reviewed for its safety evaluation included the reliability of software and EMC of the system
and training for human-machine interface. The staff stated that it took the following steps to
review the reliability of the software:

1. performed a detailed review of the system design process and software V&V program
2. reviewed available information on the software and hardware history including
   previous software and hardware failures
3. reviewed the specific plant application, including any special features that were
   required
4. reviewed the V&V performed on the software used in the licensee application. This
detailed review included
   - following the code development
   - examining the vendor/licensee interface and response process
   - reviewing software problem and error reports and resulting corrections
   - comparing the V&V to ANSI/IEEE-ANS-7-4.3.2-.1982
   - interviewing personnel involved in the process
   - verifying the independence of the software verifiers
   - reviewing the development of the fractional requirements and subsequent
     software development documents
   - reviewing the software life-cycle and future vendor/licensee interface
5. performed a "threat audit", which consisted of picking a sample of plant parameters and tracing the manner in which the licensee used these parameters in writing the purchase specifications and functional requirements for the software and in writing and testing the codes.

At the end of the review, the staff collected all of the information to establish a benchmark for assessing the performance and reliability of the software safety system.

In the SER, the staff described its review of the EMI qualification in the following sequence:

1. evaluated the plant environment to find potential EMI sources, including the effect of open doors during surveillance, the types and strengths of plant radios, the location and direction of microwave sources, and the location and effect of other equipment within and immediately surrounding the installed location.
2. reviewed and evaluated the vendor test methodology, frequency susceptibilities based on the vendor tests, and vendor system modifications to compensate for those susceptibilities. This review included comparing the as-tested and as-installed configuration.
3. reviewed the licensee's on site testing and analysis.
4. assessed the system EMI qualification based on all of the above mentioned reviews and evaluations.

The NRC completed its evaluation by considering the licensee's training and procedures. The SER states that an important part of assimilating the computer-based digital process protection system into the station environment is ensuring that all procedures affected by the modification are correctly updated and that the operators and technicians have sufficient training in the use and repair of the new system. The documents reviewed to address this issue included surveillance, channel calibration, annunciator response, abnormal operating procedures, and administrative procedures for plant modifications. The licensee also committed to incorporate into its procedures the detailed operation and maintenance manual received from its vendor.
Case Study 1

**Human-Machine Interface Error (LER 91-006-00)**

On February 17, 1991, Unit 1 at the Limerick Generating Station experienced an actuation of the primary containment and reactor vessel isolation control system, an ESF resulting in the generation of a radiation isolation signal for the drywell and suppression pool purge supply and exhaust valves. This event was caused by selecting an "undefined" wide range accident monitor (WRAM) channel item number at the safety-related data access panel (RM-23) in the main control room.

An investigation revealed that personnel error caused this actuation. While interrogating the radiological meteorological monitoring system, the shift technical advisor (STA) and STA trainee selected a channel item number (i.e., 100) that is not listed in procedure RMMS-301. When the STAs selected this channel item number, the WRAM microprocessor repeatedly searched for nonexistent data. The WRAM microprocessor was not able to complete the task of locating the requested data in the allotted time period, resulting in the error message. This error message caused the WRAM to initiate a system reset by momentarily disconnecting and connecting power. When the system reacted, the relays in the WRAM lost power. One of these relays is associated with the actuation of the high radiation isolation signal for the primary containment and reactor vessel isolation control system. Therefore, when this relay momentarily lost power, it failed to its safe position (closed) and initiated the high radiation isolation signal for the drywell and suppression pool purge supply and exhaust valves.

Case Study 2

**Software Coding Error Caused Pressure Boundary Damage (IRS No. 1118.00)**

a. **Background:**

The refuelling system used on nuclear reactors is complex, so a detailed description of the fuelling equipment will not be provided. In summary, fuelling on the CANDU reactors can be carried out while the reactor is on line. A fuelling machine (FM) moves into place near a reactor. There is one fuelling bridge at each end of each reactor. The bridge picks up the FM, positions the FM at the reactor face, and the FM head locks onto a fuel channel. The fuel channel carries both the fuel, and pressurized heavy water (D2O). The D2O removes the heat from the fuel and transfers it to the boilers. This system is called the primary heat transport system (HTS). A FM head must lock onto each end of the channel, one handled by each bridge. The FM heads then must be pressurized to the pressure of the HTS. End plugs are then removed and some new fuel is pushed in from one end, and spent fuel is pushed out of the other end. End plugs then are replaced.

The FM is computer controlled. There are three computer systems and three FMs. Each computer system consists of two or three computers, with a total of eight computers. In general, one computer system controls one FM. However, as each FM can be positioned at any reactor (with some limitations), it is clear that each computer system must have the capability to control any of the bridges.
b. Event description:

This event occurred on 23 January 1990. A FM was clamped onto a channel of unit 4 and was in the process of being filled and vented but it was not pressurized. The bridge at the east end of unit 4 unexpectedly moved downwards about 40 cm which caused damage to the fuel-channel end fitting, onto which the FM was clamped. There was a loss of D2O, which is the primary HTS. The reactor was shut down and D2O had to be transferred from other units to maintain the HTS inventory of unit 4. A valve failure complicated the cool-down procedure. Shut-down was achieved safely and subsequently the split D2O was recovered.

c. Cause of the Event:

The primary cause of the event was a software bug in the FM code. This bug had been introduced during a previous software upgrade. The bug in itself would not have caused a failure, and indeed had been in place for a considerable time, except that a number of other factors combined to cause the incident.

A previous error on one of the other FM computer systems (not the one being used for refuelling) had caused that computer to be ‘primed’ to call for the release of brakes on the unit 4 bridge. An operator carrying out an unrelated operation on this other computer system triggered the computer to ‘remember’ this previous event, and to call for a release of the brakes on the unit 4 bridge.

One of the protective computers on this other system was out of commission, so it did not trap the call to release the bridge brakes on what should be a non-controlled system. Thus this other computer released the brakes on the bridge which was being used for refuelling.

d. Identified Deficiencies:

The primary cause of the incident was a software bug in the FM code. In addition, the following deficiencies were also identified:

i. One of the computers was out of service. If it had been in service it would have prevented the movement of the bridge.

ii. The operator working on the other computer system, which released the brakes on unit 4 bridge, was unable to determine whether a warning lamp was on or off.

iii. The ability of any computer system to control any bridge should be safeguarded by mechanical interlocks, not just software interlocks.

iv. Once locked onto the reactor face, the bridge brake and motor controls should be electrically isolated to prevent bridge movement.

e. Corrective Action:

As a result of this incident, the operating authority has implemented the following changes:

i. A complete inspection and repair has been carried out on the damaged reactor components.

ii. The software bug has been rectified.

iii. A hazards analysis of the FM software has been undertaken to identify other possible latent problems.

iv. Investigations were started to review the design philosophy of the FM protective


TOOL VALIDATION - MAINTENANCE & OPERATIONAL FACTORS

SESSION 3 - AFTERNOON

TOOL VALIDATION MAINTENANCE AND OPERATIONAL FACTORS - M. Fabien Feron, Direction de la Sûreté des Installations Nucléaires

DEVELOPMENT OF SUPPORT TOOLS FOR SAFETY DIGITAL SYSTEM - Mr. Kazuhiko Tanaka, H. Yatabe, Toshiba Corporation

VALIDATION OF PROGRAMMABLE AUTOMATION SYSTEMS FOR SAFETY CRITICAL APPLICATIONS - Mr. Pentti Haapanen, Technical Research Centre of Finland

DIGITAL NEUTRON FLUX INSTRUMENTATION - Mr. Willi Bucher, Hartmann & Brown

SIZEWELL B REACTOR PROTECTION SYSTEM - SOFTWARE DESIGN GOALS - Mr. William D. Ghrist, Westinghouse Electric Corporation
The reactors of the N4 series (1400 MWe), of which Chooz B1 is one, are characterised by extensive computerisation of the instrumentation and control systems. This is one of the main differences between the N4 series reactors and those of the 1300 MWe series currently in service.

The main change is in the control room which is totally computerised to enable reactor control from workstations (control standing up being replaced by control sitting down).

As part of the Chooz B1 commissioning formalities, the instrumentation and control systems of the N4 series reactors were reviewed by the Nuclear Installation Safety Directorate (DSIN), the French safety authority, with the technical support of the Institute for Nuclear Safety and Protection (IPSN) and the standing committee for reactors.

This review has not yet been completed but has already enabled first fuel load at Chooz B1 to be authorised.

After a brief description of the N4 instrumentation and control systems, this document briefly covers the different aspects of its review by the French safety authority, particularly as regards validation tools, maintenance and operational factors.
1 - THE N4 SERIES INSTRUMENTATION AND CONTROL SYSTEMS

The instrumentation and control architecture of the N4 series reactors is based on four levels (see Figure 1):

- level 0, which comprises all sensors and actuators;

- level 1, which consists of programmable logic controllers linking levels 0 and 2. The role of these controllers is to perform all protection and automatic control functions;

- level 2, which is man machine interface. The principle interface consists of centralised computers (KIC system) and the operator stations (see Figure 2). Each operator station comprises a control keyboard and a tracker ball, and can be used for carrying out all control action. There are a number of screens displaying different types of lists: mimic diagrams, alarm sheets and procedures. These can be selected by the operator. The process is facilitated by touch sensitive screens;

- level 3, which consists of the unit supervision facilities, and provides linkage with the exterior. This level is not yet operational at Chooz B.

More specifically:

- level 1 includes the SCAT system (French acronym for unit auxiliary control system), the CO3 system (core control) and the SPIN system (integrated digital protection system), the CS3 system (an engineered safety feature support systems monitor) and the SCAP system (protective by-pass to atmosphere system);

- level 2 includes the unit operating computer (KIC system), the workstations, the wall mimic display panel and the auxiliary panel.

The SCAT system uses of the shelf PLC technology (Hartmann & Braun Contronic E) used in other sectors of industry. The CO3, CS3 and SCAP systems were specially developed by Merlin Gérin. The KIC system (site operating computer system) was developed by the Sema Group using standard equipment (Digital).
2 - TOOL VALIDATION

Extensive use was made of automatic code generation tools in development of the SPIN system software. The design utility used is known as the SAGA environment. Other tools, such as those used for checking codes and carrying out unit tests for integration and validation were also used.

The SAGA environment essentially consists of a design language making it possible to establish the functional design of applications without predetermining how they will be practically implemented. Basically, the SAGA environment makes it possible to construct complex functions from simpler functions, with interactions and relationships between data being described in graphical form (see Figure 3). The design process begins by establishing the interfaces for the application (the inputs and outputs), and continues with progressive sub-division of the functions and data.

The elementary functions are programmed manually in C, and the "complex" functions are then programmed automatically.

The design methodology of the SAGA environment has been validated by Merlin Gérin (vendor) in pilot projects since 1987. The environment has undergone specific validation tests and confirmation of its characteristics has been obtained by testing the software generated with it.

Furthermore, the safety authority has examined, using its own resources, a part of the code generated by the SAGA environment. A representative sample of the protection system has been the subject of in-depth analysis, both statically (examination of documentation and code) and dynamically. Computer tools, notably the ATLAS environment, have been used by the Institute for Nuclear Safety and Protection, the technical support body of the safety authority. This review has made it possible to obtain an adequate level of confidence in the quality of the SPIN system software, but has also revealed a few shortcomings, particularly as regards systematic compliance with the provisions of IEC 880 in the elementary functions.

The safety authority has also requested a validation dossier on the SAGA environment. Some of the information required has already been received. The safety authority has also requested to be informed of any changes in the SAGA environment.
3 - VALIDATION OF PRE-EXISTING HARDWARE AND SOFTWARE

After the Controbloc P20 system was abandoned in 1990, EDF decided to use Hartmann & Braun's Conronic E product for the SCAT system. This change of product obliged EDF to alter the architecture of the N4 instrumentation and control systems.

The use of the SCAT system for post-accident control in the medium term necessitated 2E qualification of a part of Conronic E. This led to doubling certain cards and using hardwired connections. These arrangements were not applied to the parts of category IPS-NC (non-classified but important for safety), which is less constraining and to which the great majority of the Conronic E equipment belongs.

The safety authority examined the hardware qualification of Conronic E, particularly as regards earthquake resistance.

* * *

As regards the system software developed by Hartmann & Braun, the safety authority requested EDF to retrospectively compare the methods and requirements of development of this software with those laid down in IEC 880. One of the main findings was that the development activities were properly sub-divided and that checks and witness points were present. Concerning this point, the independence of the activities is to be noted: specifications relative to design and design relative to integration and validation tests. Improvements concerning certain points of detail were nevertheless incorporated.

EDF has conducted and analysis of vendor feedback. For reference, Hartmann & Braun have assembled a database containing information on all anomalies and problems encountered with the Conronic E system. In this database, it is possible to:

- describe any problem observed by any user of Conronic E,
- assess and classify the problem (serious faults, high priority faults, low priority faults and no faults),
- to inform other users of the situation.

Finally, EDF completed validation of the vendor by a functional qualification to verify that Conronic E effectively met the N4 instrumentation and control requirements.

* * *

The application software was developed directly by Framatome and EDF. It is programmed using an integrated computer-aided design system, which automatically generates the programmable logic controller code from task-oriented graphical languages. This computer-
aided design system was validated by EDF in a number of analyses and tests. The system enables, in particular, numerous coherence and syntax checks. Furthermore, it generates all the corresponding documentation.

* * *

Overall acceptance testing (system + application) was carried out on a platform with a Contronic setup representative of that of the N4 instrumentation and control systems, mainly to test the performance levels (particularly the process avalanche case), as well as the degraded operating modes and associated signals. Apart from these factory platform tests, on-site behaviour of Contronic E has been and continues to be painstakingly monitored at the request of the safety authority.

Furthermore, EDF has developed a close working relationship with the vendor, mainly to understand the internal functioning of Contronic E. Thus, constraints imposed by EDF have resulted in a certain number of modifications of a commercial version. Similarly, some of the problems initially detected were analysed jointly by EDF and Hartmann & Braun to determine the best solutions.

4 - FAILURE OR CRASH OF THE SYSTEM

4.1 - SPIN system

The SPIN system consists of four redundant processing channels which automatically trip the reactor on the basis of a 2/4 vote. This redundancy ensures immunity to hardware failures and enables periodic tests and maintenance to be carried out.

The SPIN system software is designed to ensure a high degree of quality and a virtual absence of defects. Furthermore, it includes self-test functions making it possible to detect operating faults and warn the operators. However, despite the intensive testing, a common mode failure of this software is postulated. Therefore:

- the safety authority ensures the adequacy of the tests performed by the vendor on the "critical" parts of this software;

- a diversification system is necessary. This diversification (ATWS) is provided by separate equipment (the SCAT system using Contronic E technology). This principle was also adopted for the 1300 MWe reactors, which have been operating since the early eighties, and for which the SPIN system has, to date, never suffered a mission failure.
4.2 - KIC system

The KIC system enables true computer-aided plant operation. The computers collect data which they process for greater legibility. Similarly, the data is presented in a manner facilitating diagnosis of the situation (e.g. processing and hierarchy of alarms). In addition, operating procedures have been computerised so that they can be displayed and run from the operator stations.

EDF has evaluated the dynamic performance of the KIC system, connected to the Contronic E equipment: firstly on the basis of 1300 MWe unit transient replicas, then with the Chooz B1 hot test results. In particular, these tests have made it possible to verify the contingency protection mechanisms implemented for beyond-design-basis phenomena.

The possible failures are taken into account in the design of the computer system and EDF have verified the acceptability of their consequences. The safety authority has nevertheless requested EDF to continue these studies in greater depth, particularly by analysis of the software failure modes and their effects (FMEA).

The consequences of a total crash of the KIC system have been examined: the presence of a conventional auxiliary panel (hardwire connection) ensures freedom from this problem provided that:

- the conditions of switching between the KIC system and the auxiliary panel are clearly established,
- the reactor can be controlled, under all circumstances, from the auxiliary panel.

These two aspects have been the subject of an in-depth review by the safety authority, mainly on the basis of observations made on a full-scale simulator: operating teams have been confronted with a number of situations (normal operation, and incident and accident conditions) and a number of control modes (from the auxiliary panel, from the KIC system, and with the latter fully serviceable or partially degraded). In particular, the safety authority has requested the setting in place of an additional means of monitoring the KIC system ("life signs"). These enable the operators to detect contingent residual malfunctions that the system is not yet capable of treating automatically.

Nevertheless, to ensure the reliability of the KIC system, EDF has corrected the majority of the bugs discovered during platform acceptance testing and during on-site utilisation (before Chooz B1 fuel loading). Where bugs have been left uncorrected, a thorough and formalised justification has been made, both from the software implementation point of view and from the reactor control point of view.

Furthermore, the on-site behaviour of the KIC system has been, at the request of the safety authority, monitored closely (including measurement of CPU load, memory usage etc.) and the safety authority has requested EDF to carry out retrospective analyses of any errors occurring, notably on the basis of the system messages generated (error tracing).
5 - MAINTENANCE

The redundant architecture of the SPIN system makes it possible to carry out periodic tests and maintenance during operation of the unit. However, it is not currently planned to upgrade the software during an operating cycle. By way of comparison, upgrades of the SPIN software of the 1300 MWe reactors have been approved by the safety authority and have normally been implemented at the start-up of a new plant. Whatever the case, EDF must explain the reasons for the introduction of a new version, and submit an impact study of the modifications made.

* *

Conronic E is designed to make on-line maintenance possible. Furthermore, the safety authority has requested EDF to adopt a standpoint concerning contingent upgrades of the commercial version.

* *

The KIC system has duplex redundancy for reactor control functions, which enables maintenance to be carried out on it on line. EDF has identified unit states in which a software upgrade can be incorporated and procedures established for application when required. Whatever the case, the auxiliary panel remains serviceable and can be used to control the unit during such operations. Furthermore, it must be borne in mind that on-site installation of a new version may only take place after the version has been validated (or modifications made) factory platform testing.

* *

The updating of the data configuring the application part of the programmable logic controllers (Conronic E) and the computerised control system (KIC system) have also been the subject of assessment by the safety authority. This assessment is justified by both the large number of data configured and also the possible consequences for operation of the reactor. Indeed, it may be necessary to update the parameters as a function of the depletion of the fuel, the state of the unit or for correction of minor errors etc. The safety authority has asked EDF to establish procedures for ensuring proper progress of these operations, particularly in terms of planning and conditions for implementation, identification of modifications made and functional requalification.
6 - CONCLUSION

The behaviour of a computer-based system obviously depends on its architecture (allowance for redundancy and diversification requirements) as well as its hardware and software components.

The requirements concerning the hardware components (earthquake resistance, resistance to electromagnetic disturbances etc.) are generally clearly defined and comprehensive compliance with them can be subsequently verified by appropriate tests.

Totally exhaustive verification is not generally possible with software. Their development (including the verification validation tests), as well as preparation of the associated documentation, must therefore be as thorough as possible. As far as possible, the operator must be closely involved in this development process to be able to:

- understand the operation of the system at the technical level,
- verify proper allowance for its requirements (compliance with specifications, tests during operation, maintenance, modifications, behaviour in degraded mode etc.),
- verify proper integration of the system into the overall instrumentation and control installation (interfaces).

System operating experience feedback must, of course, if it is available and representative of the utilization planned, be analysed to substantiate the assessment made of the development process.

Suitable documentation must vouch for the above points so as to enable the safety authority to judge the pertinence of the objectives or criteria adopted and verify the acceptability of the results obtained.

Furthermore, application of the defence in-depth principle will necessarily lead the safety authority to examine the arrangements taken by the operator to protect against contingent failure (whether partial or total), of the system and to mitigate the consequences.

Finally, the conditions of changing the version of the system, the action to be taken in the event of degraded operation or loss of the system must be predefined and their operability must be verified.

Then, after analyzing the operator safety case, the safety authority can license the system and accept its use in a nuclear power plant.

The prior judgement must then be substantiated by careful observation of the system during normal operation of the nuclear plant.
N4 SERIES I & C SYSTEMS

Control room (level 2)
- Remote Shutdown Panel
- Auxiliary Panel
- Wall-mounted Mimic Panel
- Operator's Workstations
- Central Data Processing (KIC)

"Conronic E" digital and analog control equipment

Core Control and Command (including Digital Protection System)

Turbine-generator controller

Control of safeguard support system

SENSORS AND ACTUATORS
(OUTPUT 1, OUTPUT 2, OUTPUT 3) = F(INPUT 1, INPUT 2, CST)

(OUTPUT 1, DATA1) = F1(INPUT 1, INPUT 2)

(OUTPUT 2, OUTPUT 3) = F2(DATA 1, CST)

= F2(F1(INPUT 1, INPUT 2), CST)

F1 and F2 are elementary functions
F is a complex function
Development of Support Tools for Safety Digital System

Toshiba Co. Kazuhiko Tanaka
Hitachi Ltd. Hiroshi Yatabe

The digital control technology has been applied and expanded to the nuclear power plants on the step by step, through none-safety systems to safety systems in Japan. Kashiwazaki-Kariwa Nuclear Power Station No.6 and No.7 of Tokyo Electric Power Co., the first ABWR(Advanced BWR) in Japan, have adopted the comprehensive digital control systems including the safety related systems.

For applying this safety digital systems, we make its software to keep high reliability usisng same designing and manufacturing method as none-safety digital systems which have gotten good results on operating the nuclear power prants in Japan. This paper reports the characteristics of software language, and the software manufacturing and testing method using the support tools.

1. The Characteristics of Software Language

We apply POL(Problem Orieted Language) for the safety digital systems, which is symbolic language and has good results on none-safety digital systems.

POL have been used in the middle of 1970's, which history is shown in Fig. 1.

On the other side, the control and instrument systems of the nuclear power plants had been digitalized in the first of 1980's, which first step was for radioactive waste treatment control systems and the single looped control systems.
And the second step of applying the digital systems, from the first of 1990's, was for none-safety systems as normal systems in nuclear island or turbine island.

These none-safety digital systems have used the software logics with POL, which have gotten good results on construction stage and plant operating stage.

The software for the safety digital systems was designed and manufactured based on the same method as the none-safety digital systems.

It is easy to compare the software logics with POL, which sample is shown in Fig.2, into IBD (Interlock Block Diagram), as POL is the symbolic language similar to IBD.

The characteristics of the software logic with POL are the followings.
(1) Composed of a single combination of logics such as AND/OR
(2) Repetitive processing (structure of the priority processing based on the event)
(3) No interruptive processing
(4) Description with some ten's symbols

2. The designing and manufacturing method of software logic

(1) The manufacturing flow of the digital system

The digital control systems are designed and manufactured according to the manufacturing flow shown in Fig.3.

On each step of this flow, the requirements of the basis design are clearly specified on each documents or drawings for the following design.

This manufacturing flow is applied not only none-safety digital
systems, but also safety digital systems. The software manufacturing flow is shown in Fig4, which should be included in the digital system manufacturing flow. The first step of software designing is to clear the requirements for software logic from the system design specifications as IBDs. The software logics, which diagrams are similar to IBD, are designed with POL according to the requirement of IBDs. For this software logic designing, we use CAD systems that can draw software logic diagram and convert software to load digital controllers. And we also use the digital maintenance tool for loading software to digital controllers and comparing software as support tool.

(2) The Support Tools for Digital Systems

CAD systems and the digital maintenance tools are used for designing and manufacturing works of software logics.

① CAD system

CAD system can produce the software logic diagram with POL, and convert the software logic into POL command software as loading to digital controllers. Software designers input POL symbolic commands, memory numbers related with input and output signals, name of signals and sequential number of POL commands according to IBD. The designers can compare the software logic diagram output from CAD system, which format is similar to IBD, into the logic on IBD. CAD system also select and convert programing data from the software logic to load software to digital controllers. This conversion, shown in Fig. 5, is arranging the programing data.
according to the sequential command number by time shared. On the process of this conversion, programing language is also converted another language that digital controller can read under the simple rule, with automatically checking mis-conversion using self-diagnosis.

② Digital Maintenance Tool

The digital maintenance tool has mainly 3 functions as loading software to digital controllers, comparing software and monitoring software in digital controllers. The function of loading software is to load software logic produced in CAD system into digital controllers using FD (Floppy Disk). The other function, comparing software, is to compare software in produced in CAD system with software loaded in digital controllers. The last function is to monitor the software logic and signal conditions in digital controllers.

(3) Testing of Software

The manufactured software had been tested on each stage as manufacturing stage, shop testing and site testing.

① Software manufacturing stage
- Drawing comparing software logic diagrams with related IBDs
- Conversion checking with self-diagnosis in CAD system
- Automatically comparing software loaded in digital controllers with software manufactured in CAD system.
② Shop testing stage (loading software in digital controllers)
  · I/O characteristic test of ON/OFF signals and analog signals
    (Confirming the validity of the I/O functions)
  · Sequence test
    (Confirming the validity of the logic in the controller based on IBD)
  · Combination test
    (Confirming the validity of the logic combined with digital controllers)

③ Site testing stage
  · Electrical test
    (Confirming the software logic, transmission and instrument loop combined with all digital controllers and actuators)
  · Pre-operational test
    (Confirming the system interlocks based on IBD, safety protection function for each trip parameter)
  · Heat up test
    (Confirming the plant interlock works to shut down the plant safety, and evaluating the plant transient response)

3. Maintenance of Software

It is possible to change the software as requirements of system design or plant operation procedure.
The changing of software should be managed to protect mischange under software changing procedure which is shown in the following, and same as software manufacturing.
The software logic is changed in CAD system based on stored data, and compared with IBD changed according to requirement of system designs
or plant operation procedure.
The changed software is loaded to digital controller using the digital maintenance tool.
The maintenance tool compare the software not changed with the software already changed.
The operator can confirm the validity of software deviation by checking the result of maintenance tool comparing.

4. Conclusion

The applying safety digital system is based on the step by step approach started through radioactive waste treatment control system to other none-safety systems which has good results with no trouble or problem caused by software.
We are constructing Kashiwazaki-Kariwa Nuclear Power Station No.6 and No.7 of Tokyo Electric Power Co. which are Advanced BWR, and adopt the comprehensive digital system including safety systems.
Unit No.6 had completely finished shop testing and pre-operational testing under V&V(Verification and Validation) basis to JEAG4609, and now under heat up testing.
We has gotten the good results on these testing with no problem caused by software, and evaluate that the selection of software language and the manufacturing procedure of software are proper.
Fig. 1. History of Software Language

- **FORTRAN II** (IDM 704)
- **FORTRAN IV**
- **ANSI**
- **JIS**
- **FORTRAN 77**
- **ALGOL 58**
- **ALGOL 60**
- **ALGOL 68**
- **PASCAL**
- **COBOL**
- **COBOL 65**
- **COBOL 73**
- **COBOL 76**
- **COBOL 78**
- **ANSI, JIS**
- **NPL**
- **PL/I**
- **RPG**
- **RPG II**
- **RPG III**
- **BASIC**
- **POL**
Interlock Block Diagram

Conventional System (Relay Sequence)

Digital System (Logic Sequence)

Fig2. Sample of Logic Diagrams
Fig. 3 Design and Manufacturing Flow of Safety digital System

System Requirement

System Design Specification
(System Spec.)

H/W, S/W Design Specification
Equipment Design Spec.
Interlock block Diagram

H/W Design
(ECWD)

Components
Procurement

Fabrication

S/W Design
(S/W Diagram)

S/W Coding

Integration of H/W&S/W
(manufacturing completed)

Factory Test Procedure
H/W test
S/W test
Semi-Dynamic Simulation Test

Shipping Procedure
(Packing Check/Delivery Check)
Fig 4. Software manufacturing Procedure

Design requirement for hardware/software

- Hardware Design
- Software Design

- IBD
- Inputting Software Logic
- Comparing/Checking

Digital Controller

Display of Maintenance Tool

Digital Maintenance Tool
Fig 5. Sample of Software Diagram and POL Command
VALIDATION OF PROGRAMMABLE AUTOMATION SYSTEMS
FOR SAFETY CRITICAL APPLICATIONS

Pentti Haapanen*, Jukka Korhonen** & Urho Pulkkinen*
Technical Research Centre of Finland (VTT)

ABSTRACT

System vendors are offering safety critical functions in nuclear power plants realised with programmable automation systems. The designed functionality is configured from existing software modules, which are written before the application design process is started, and whose development process may be hard to trace down afterwards. Common approach in validating software based systems by gathering evidence on the high quality of the design process can not totally be realised and other means to achieve assurance about the system safety must be applied. Dynamic testing of the final product gets in this case even more importance.

Diversity is required in safety critical software based systems due to the nature of the software faults as design faults to cause all redundant channels to fail simultaneously. Some experts say that a purely programmable system can not be accepted alone but an analogue back-up is needed, because it is hard to show that diverse program version are free of common faults.

The paper discusses the problems encountered in validating applications based on programmable automation systems, the mechanisms of common mode failures in diverse programmable automation systems and describes the experiments with a dynamic test harness realized at VTT.

* VTT Automation, P.O.Box 1301, FIN-02044 VTT, Finland. Tel. +358 0 4561, Fax +358 0 456 6475
** VTT Electronics, P.O.Box 1100, FIN-90 571 OULU, Finland. Tel. +358 81 551 2111, Fax +358 81 551 2320
VALIDATION OF PROGRAMMABLE AUTOMATION SYSTEMS FOR SAFETY CRITICAL APPLICATIONS

Pentti Haapanen*, Jukka Korhonen** & Urho Pulkkinen*, Technical Research Centre of Finland (VTT)

1 INTRODUCTION

One viable alternative to realise the safety critical control and protection functions having high requirements on dependability (e.g. the reactor protection system) in nuclear power plants is to implement the desired functions on a programmable industrial automation system. This approach was, for example, a couple of years ago proposed by some plant vendors for the fifth Finnish nuclear power plant, which then, however, was cancelled. In this approach the control and protection functions are configured - usually using the base systems own formal graphical description language - from a limited number of off-the-shelf software modules ("function blocks") from the system module library. The system software (a simple operating system) provides facilities to run the configured application programs on a specialised hardware system. The system is operating strictly cyclically (task running, message transmitting and use of system services) without interrupt handling. The hardware system consists of subrack containing central processor modules, bus interface modules and I/O interface modules housed in electronic equipment cabinets. The implementation process here deviates from an ordinary software design process and also the qualification procedures have to be adapted correspondingly.

The safety assessment for the licensing process of computer-based systems in general can not entirely be based on conventional probabilistic methods because of the difficulties in the quantification of the reliability of the software and also the hardware. The evidence on the high quality of the whole design process - the term "design" encompasses everything from system requirements to realization during both initial production and future modifications - has therefore an important role in the licensing process of any safety critical computer-based system. In case of using an industrial automation system as platform, the design process actually is divided in two distinct phases, in many cases clearly separated from each other both in time and location:

- the base system design process, and
- the application design process,

as illustrated in Fig. 1. The figure also gives the different sources of evidence on the system dependability, on which the granting of the operating licence can be based. The evidence contains information about technical, managerial and quality assurance details of the system and the design processes. The weight given to each single piece of evidence from different sources varies greatly from case to case, but in any case the evidence from the first phase should have an important role in the total licensing process.

The division of the design process in two separate phases can cause significant problems for the licensing process. The base automation system design process has usually been completed well before the actual plant design process is started. Therefore it may be hard to trace down the dependability arguments from

* VTT Automation, P.O.Box 1301, FIN-02044 VTT, Finland. Tel. +358 0 4561, Fax +358 0 4566475
** VTT Electronics, P.O.Box 1100, FIN-90 571 OULU, Finland. Tel. +358 81 551 2111, Fax +358 81 551 2320
the base system design process afterwards, if the documentation of the process has not been comprehensive enough. In such case the effort needed for validation of a specific application for the most demanding tasks may require considerable re-engineering of the base system design, which would be very tedious and expensive.

A carefully executed and documented base system design process can, on the other hand, provide assuring evidence on the automation system dependability — the system itself and the implementation tools and methods — for the implementation of demanding applications. This evidence, once gathered, can serve the needs of many projects thus considerably decreasing the effort and costs of an individual licensing process. Furthermore, the experience from the applications of the base system in other fields of industry can provide very valuable evidence on the soundness of the base automation system, that can be benefited in a nuclear plant licensing process. What remains to be done in the individual application licensing process is to prove the dependability of that specific application.

In the following two central issues — diversity and dynamic testing — in licensing of programmable automation systems for safety critical applications are discussed. Some diversity is usually required for the most safety critical applications to protect against the possible common cause failures of redundant trains. The use of an industrial automation system as a platform for a specific application sets some special restrictions for the ways of using diversity. If the base automation system has been proved to be of high quality during the base system design process and/or through application experience, the dynamic testing of the specific application can have an enhanced role in the licensing process.

2 DIVERSITY

Software faults are design faults introduced to the software during the design and implementation process. Some (unknown) input sequences may trigger a residual software fault — fault still remaining in the software after all verification and validation efforts — to cause an erroneous response of the software-based system. In a multi-channel redundant system having identical software in each channel also the possible residual faults are present in all channels. If an input sequence triggering erroneous response is fed to all channels they may fail simultaneously. The residual software faults can thus be a source of Common Cause Failures (CCF's) of a redundant programmable system.

Other sources of Common Cause Failures are the external events (fires, floods, earthquakes etc.) and dependence factors (common support systems, common rooms etc.). These CCF sources can be avoided by proper functional and physical segregation of the redundant channels and their support systems from each other. To protect the system against CCF's caused by residual faults in the software these measures are ineffective. Therefore some diversity between the individual redundant channels is usually required for safety critical functions. The diversity, however, is still a controversial question and a lot of further research is needed to develop models that make it possible to understand and evaluate its effects on the system reliability.

A general structure of a diverse system is presented in Fig. 2. The system consists of at least two diverse subsystems ("variants"), each of which usually has 2 - 4 identical redundant trains to protect against random component failures, external events and dependence factors.

In general the probability \( P(AB) \) of both variants failing at the same time can be written as:

![Figure 2. Basic structure of a diverse system.](image)
\[ P(AB) = P(B) \cdot P(A \mid B) \]  
\[ P(AB) = P(A) \cdot P(B) \]

where \( P(A) \) and \( P(B) \) are the failure probabilities of variants A and B, respectively, and \( P(A \mid B) \) the conditional probability of A failing when B has failed. In the ideal case of complete independence this comes:

\[ P(AB) = P(A) \cdot P(B) \]

In real cases the assumption of complete independence is seldom justified. This means, for example, that if the failure probability of each of two variants is \( p \), the total failure probability of the diverse system lies somewhere between \( p \) (total dependence, \( P(A \mid B)=1 \)) and \( p^2 \) (total independence, \( P(A \mid B)=P(A) \)). Knight & Leveson (1986) e.g. have shown, that independent programming teams tend to make similar errors and thus their products may fail simultaneously. As first Eckhardt & Lee (1985) and later Littlewood and Miller (1989) have postulated, the possibility of dependent failures in diverse software-based systems may stem from the “difficulty” of the input data. Thus diverse pieces of software when introduced to a “difficult” input data set will fail simultaneously. Especially this goes for random diversity, where two design teams produce their applications from same specifications without knowing from each other. Forced diversity can, however, considerably improve the situation. In this case the design teams systematically avoid the use of similar design solutions under a common design coordinator. The assessment of the influence of the diversity on the system reliability is a difficult task and more research is still needed to solve this question.

There are also some important disadvantages connected to the use of diversity in the realization of control and protection functions. Firstly, it may considerably increase the design and construction costs of the systems. Secondly, perhaps even more importantly, it makes the maintenance of the system more complicated and increases the risk of maintenance errors. This can, in worst cases, even lead to decreased reliability and safety of the total system. The degree of diversity in the control and protection system should therefore be considered very carefully, and realized only to that extent that is necessary to achieve the required confidence on the system dependability.

As illustrated in Fig. 3, a system can have arbitrarily high (unknown) reliability that clearly exceeds the required, task specific reliability. The reliability figure acceptable in the licensing process, however, generally depends on the evidence provided for the system dependability. The more effort is used for qualification of the system, the higher reliability figure can be accepted. In a diverse system each variant requires its own qualification effort, the total effort being the greater the higher is the degree of diversity. On the other hand, the reliability claim (and thus the qualification effort) for each variant can the lower the higher is the degree of diversity. The optimal degree of diversity can in principle be found at the minimum of the total qualification effort (and associated costs). In practice this can only be found by sound engineering judgment since the actual relations between costs and reliabilities are unknown.

The diversity in the system can be realized by applying a proper combination of the three design diversity principles:

- functional diversity,
- software diversity and
- hardware diversity.

Some low degree of diversity can even be achieved by using independent (and different) sensors and asynchronous operation in different trains of a single variant. Due to the small differences in sensor signals
the input sequences ("trajectories") to the redundant trains slightly deviate from each other which may be able to prevent the common residual faults to cause simultaneous failing of all trains.

Although it in principle is possible to apply all previously named diversity principles also in case the functions are implemented on a programmable automation system, the possibilities in practice are restricted. Practical alternatives are the use of functional diversity, use of two different programmable system or the use of an analog back-up system.

2.1 Functional diversity

Some software diversity can be realized inside a programmable automation system by using in different redundant trains different function blocks to realize the same integral function, selection of different execution order of the subtasks, dividing the tasks differently to different processors etc. These possibilities may, however, be rather restricted and real meaningful diversity can be achieved only by defining different functions realizing the desired total safety goal in different trains.

2.2 Two different programmable systems

In the most demanding safety tasks it may be difficult to achieve — and prove that it has been achieved — the required diversity between the variants, but a secondary, diverse system is needed. The secondary system may be either an analog, hardwired system, or another programmable system from different manufacturer. This alternative has e.g. been selected by Westinghouse for the Temelin plant in the Czech Republic. The secondary system could possible be a simplified version implementing only the most safety critical tasks of the primary system, e.g. it could be implemented on a Programmable Logic controller (PLC).

The selection of another programmable system has the advantage of being of the same modern technology as the primary system, which may ease the maintenance of the systems. On the other hand, the difference between the two systems is an implementation of random diversity, and it may be difficult to prove the degree of independence achieved between the systems.

2.3 Analog back-up system

The extreme alternative to protect against common cause failure propensity of software-based systems would be the use of a conventional hardwired analog back-up (or "secondary") system realizing at least the most important safety functions of the programmable system. This approach has been applied e.g. in Sizewell B-plant in UK. There has also been opinions presented until quite recently, that a purely programmable system can not be accepted for the most safety critical applications alone, but an analog back-up is necessary. The background of these lines of thought seems to be the assumption of complete independence of failure mechanisms between the programmable and analog systems. This assumption may, however, not be fully justified, since the principle of "difficult" inputs may even be generalized to concern also analog back-up systems.

The use of analog back-up system forces the plant user to maintain human and technical capabilities for two different technologies, which will increase the costs and also the risks of maintenance errors. The analog systems are rapidly giving way to the programmable systems in all fields of industry and their availability may worsen dramatically in the future. When this trend has completed, it would no more be possible to benefit at the nuclear power plants from the experiences of other industries. The availability of skilled people trained to construct and maintain these systems may also get short in the future.

For these reasons it seems wise to avoid this kind of solutions. In any case an unconditional requirement for analog back-up should not be included in the licensing requirements but the selection should be left to the plant owners and architect engineers to be made on the basis of the balance between the pros and cons of different alternatives.
3 DYNAMIC TESTING

The problem with safety critical systems is to show that the stringent requirements have really been met. This problem exists not only with high dependability systems, but also with systems having much more moderate reliability requirements, for instance when the required probability of failure is less than $10^{-3}$ per demand. Demonstration of safety and reliability of the system is usually based on several pieces of evidence, like evaluation of the production process and evaluation of the product itself, i.e. how efficiently the system can detect errors and what kind of mechanisms are there in the system for error recovering. Typically much emphasis is placed on the use of techniques and tools in the software process. It is also argued that the analysis and design methods can have a major impact on the final quality of the product.

However, this kind of evaluation can only provide qualitative information about the system. In spite of even extensive qualitative evidence the actual reliability may turn out quite different from the estimated reliability figures. Therefore safety assessment is generally enhanced with dynamic testing, during which a confidence level towards the system reliability will be established. If the selection of the test cases follows the operational profile of the system, then probabilistic estimates on system reliability can be conducted on the bases of dynamic test results.

3.1 Test harness

The test concept for dynamic testing is sketched in Fig. 4. The generated test sequences are fed from the test data files to the target automation system for producing the system response. The same test sequences are used as inputs for the logical model ("the test oracle") of the target system to produce the expected test results. The system and test oracle responses to the test sequences are then compared to each other. If the responses are not equal, the system and the logical model have to be checked for the difference.

Though the concept seems to suggest that the test cases are fed into the target system and the logical model in parallel, this is not a practical way to realize the testing. The same test sequences have to be driven through the target system and the model, but not necessary in parallel or not even in the same environment. In order to free the on-line part of the test environment from test data transformations and to make the real time execution of the test easier, and also to free the test oracle from the real time requirement, it is
reasonable to divide the test harness into the off-line and the on-line parts as presented in Fig. 5.

The off-line system contains all the test data file preparations, the expected response calculations and the response comparison. The on-line part of the test environment, the actual test harness, contains only the I/O-drivers for feeding the test data to the test object and for recording the object response. The recorded test object responses are moved e.g. on diskettes to the off-line system for comparison to the expected responses.

The prototype test harness is implemented on an ordinary PC-computer equipped with proper I/O-cards for the connection to the test object. The test harness output driver is a simple program loop timed by a clock signal from the operating system clock. The original test data table is arranged off-line to separate input files for each channel. The output driver reads at each time step signal values from these files, makes the necessary interpolation, scaling and possible adding of noise, and loads them to D/A-channels that are connected to the target system. The input driver is continuously observing the state of the outputs of the system. Each time some of the outputs has changed, the identification of the channel, the exact time of the change and the value of the output are stored in the target system response file.

3.2 Logical models

In dynamic testing, the actual response of the target system has to be checked for correctness. This is realized by comparing the actual response to an expected response produced by the logical model. Clearly the requirements of the logical model are stringent as it must meet the requirements of the target system. But even more important is that if it happened to behave in a non-specified manner, it must behave in a different non-specified manner compared to the target system. Otherwise the non-specified behaviour will remain undetected. So what the logical model really needs is as much diversity as possible compared to the target system.

Logical models may exist in various forms: in a very simple case no modeling is perhaps needed at all but the model already exists in tester’s head. In some other cases the model may be a group of algorithms that can be implemented by using some mathematical tool. However, in this case formal modeling method was selected to deal with the complexity of the target systems and to increase diversity.

The logical model is, however, not a complete model of the target system. Some features of the real system can not be studied by means of logical models. The most important of these features are real-time properties. Usually programmable automation systems have real-time requirements, meaning that the system has to respond to a stimulus within a certain time limit. However, as the implementation of the logical model and the target system are diverse, the model can give no information about the real-time characteristics of the target system. In addition, the target system may contain redundancy for higher dependability, i.e. the system includes several redundant units, voters etc. Also these design decisions are ignored in the logical modeling, as they have no significance regarding the logical behaviour of the system.

For creating a model of the target system, a high abstraction level language is needed to describe the behaviour of the system. The language has also to be executable, which presupposes formal semantics. Usually formal languages have two different notations: one for describing the reactive part (i.e. the behaviour) of systems, and one for describing the data transformations of systems. The reactive part is in many cases described using state machines and the lowest level activities (i.e. data transformation part) by means of some high level language like ADA or C.

Several alternative methods can be applied for creating the logical model of a typical target system. For embedded systems perhaps the most generally used are the RT-SA-method and Statecharts. In order to find the most suitable tools for logical modeling, different methods were actually applied to a realistic target system and their efficiency with respect to specified criteria were compared. In the research project carried out at VTT ReaGeniX (based on RT-SA, developed by VTT Electronics) proved to be the best choice for logical modeling. An example of a ReaGeniX data flow diagram is given in Fig. 6.
Figure 6. An example diagram of a ReaGeniX logical model.

An important aspect in creating the model is to select proper documentation for the starting point of modeling. It is not wise to use implementation documents as a reference, as they may already include some shortcomings or misinterpretations. Therefore the model of the target system should be based on the products of early phases of the development process, e.g. the requirements description of the system.

3.3 Test data selection

In general, it is impractical or even impossible to attempt to verify every specified requirement under every possible combination of input data. The amount of work to define, execute and analyze the results grows simply too high. The more common challenge is to select the input data in such a way that an acceptable level of confidence for the necessary quality measurements will be reached. Two basic approaches are used for selecting the contents for the functional test procedures: deterministic or statistical. In the deterministic approach, test data is selected through a systematic analysis of the requirements specification and possibly the software design and implementation. In the statistical approach, test data is randomly selected based on probability distributions defined across the test input domain.

A good source for test case generation would be the actual measurement data from various plant transients in a real operating plant. Plant simulators also form an excellent source for test data. In simulators one can generate such severe accident situations, for which the actual plant data is not available. A third natural source of test data are different transient analyses prepared for the plant safety analyses. Thus test cases should include for instance random sensor failures to some input signals. Also some oscillation may be possible for both types of signals. Experience from real life shows that sensor failures are typical situations where inherent faults actualize as an erroneous functioning of the whole channel or subsystem.

In some cases (Sizewell B, for instance) the generation of the test cases has been started from a rather limited set of basic cases (e.g. 10 to 20) drawn from different sources available, and then multiplied to a larger set by proper randomizing techniques in order the reach statistical significance. Randomizing is generally implemented by adding noise to the signals and pointing random sensor failures to various signals.
In any case, following "operational profile" in test data sampling offers two obvious advantages, which both are certainly useful from the point of view of dynamic testing:

1. There are efficient methods of selecting random points algorithmically, by computing pseudo-random numbers; thus a vast number of tests can be easily defined.

2. Statistical independence among test runs allows statistical prediction of significance in the observed results.

In spite of the strong potential benefits, statistical testing may also suffer from weaknesses encountered in practice. First, it may be difficult to calculate test inputs for random testing, often due to complicated input values. Even worse, the system or the environment may be a novel one, resulting to difficulties in determining the operational profile. In some cases, the type of the target software may cause problems.

Usually it is fairly easy to apply the principles of statistical testing to batch and interactive programs. Batch programs handle one input at a time and their output depends solely on that input. Previous input has no significance to the next output, i.e. the sequence of successive inputs is irrelevant. Therefore statistical testing is easy - tests have nothing to do with each other. Interactive programs are different from batch programs in the sense that program state is affected by previous inputs. Depending on the program state, the response to an input may vary. Thus testing of interactive programs must take input sequences into account. Real-time control programs operate continuously, reading and receiving input data and producing output data. In addition real-time programs usually require multiple input streams, where not only their value and sequencing, but also their behaviour in time are significant.

3.4 Statistical analysis of test data

The classical approach, proposed for instance by Parnas et.al. (1990), employs a simple method for interpreting quantitatively the results of dynamic testing. The approach states that in most safety-critical applications it is not necessary to know the actual probability of failure; it is enough to show that the failure probability is below a specified upper limit. Before the testing is started, the probability for the correctness of the result (i.e. confidence level) should be set according to the safety significance of the target system. A series of test cases can be modelled as Bernoulli trials and thus the probability that \( N \) independent random test cases are successful (given the failure probability is \( p \)) is:

\[
P(\text{"} N \text{ tests successful\}|p) = (1 - p)^N
\]

(3)

From (3) the upper confidence bound for \( p \) at confidence level \( \gamma \) can be determined:

\[
p_u = 1 - (1 - \gamma)^{1/N}
\]

(4)

The required number of successive succesfull test cases \( N \), when acceptable failure probability is less than \( p_u \) with confidence \( \gamma \), is then:

\[
N = \frac{\log(1 - \gamma)}{\log(1 - p_u)}
\]

(5)
Fig. 8 shows the required number of test cases. It is easy to see that as the required reliability grows high, the amount of testing increases dramatically.

Figure 8. Required number of successfully passed tests to prove the required reliability with desired confidence.

According to this approach, detection of an error presumes that testing has to be started from the beginning after completing the necessary modifications. The drawback of not being able to utilise earlier testing experiences is avoided in the approach proposed by Korhonen & Pulkkinen (1996). The method suggests that if a failure is detected, the number of test cases in the next dynamic testing should increase, i.e. reliability requirements should be more stringent.

The proposed procedure is the following:

1. At the start of testing, the number, \( n_1 \), of test cases that must be executed failure-free for the testing to succeed, is computed.

2. The system is put on test and either successfully executes the \( n_1 \) test cases, in which case the testing stops and the system is declared to have achieved its \( pfd \) (probability of failure upon demand) requirement, or a failure is observed on test case \( s_1 (<n_1) \), in which case the testing is stopped.

3. In the light of the evidence of one failure in \( s_1 \) test cases, we compute the number, \( n_2 \), of further test cases that must be executed failure-free for the testing to succeed and stop.

4. The system is put on test again and if it successfully executes the \( n_2 \) test cases, then the testing stops and the system is declared to have achieved its \( pfd \) requirement, or if a failure is observed on test case \( s_1 + s_2 (s_2 < n_2) \), then the testing is stopped and we have to return to item 3.

This basic procedure can be repeated every time a failure has been detected.

Another possibility to take into account failures during tests is to assume, that the number of failures is binomially distributed random variable, and to determine the classical upper confidence bound for the failure probability given the number of tests and failures. The hypothesis that the failure probability is
below a given limit with given confidence can be solved on the basis of the upper confidence bound. Both the cases with or without failures during testing can be analysed also from a Bayesian viewpoint. To do this, the posterior distribution of the failure rate, given the evidence obtained from tests, must be determined. If the posterior probability that the failure rate is below a given limit is larger than given confidence level, then the system is accepted. It is worth noticing that if the prior distribution in Bayesian method is uniform or noninformative, the results are very close to the classical approach. Same is true also for informative priors, if the failure probabilities to be demonstrated are very small, i.e., the prior evidence on the failure probability must be very strong, if the reliability requirement is strong.

The Bayesian approach can be used also from another, predictive viewpoint. The acceptance criteria for the above methods are similar for both classical and Bayesian approach, and they are based on confidence bounds. The predictive Bayesian reliability criteria are probability statements on the number of failures during additional tests. Further, the confidence levels of these criteria do not correspond directly to those of above mentioned approaches. In practice their interpretation is rather difficult.

Number of additional tests required after observing failures may be large. It depends also on the statistical acceptance method. The differences between the basic Bayesian and classical approaches are not large, if they are based on same assumptions.

![Bayesian 90% probability bounds](image)

*Figure 9. Upper 90% confidence bound as a function of the number of successful test cases with different beta-prior distributions.*

If failures are detected during testing, then the procedure of estimating new, failure free number of test cases can be repeated over and over again. In practice, several repetitions of the procedure are certainly rare. Continuing of testing is waste of time and resources in such a case that in spite of high safety and reliability requirements of the target system many errors have already been detected. Other kind of assessment methods should then be applied to uncover the real reason to failures.

4 CONCLUSIONS

The use of a highly qualified industrial automation system as a platform for safety critical applications can offer many advantages from the licensing process point of view. This, however, preconceives that the base
automation system design process has been properly carried through and documented. Systematic collection and documentation of operating experiences from the applications of the base system in other fields of industry can provide valuable evidence on the high quality of the building blocks of the base system and also on the ability of the implementation tools to produce high quality applications. The formal configuration tool, easily comprehensible both for process and automation engineers, can considerably diminish the danger of specification errors, which are a serious source of common cause errors.

The automation system approach sets certain limitations for the use of the diversity principles in the realization of safety critical applications. If the limited means to apply diversity in a single system are not enough to fulfill the dependability requirements of the specific application, the following alternatives remain:

- use of functional diversity in the system,
- use of a secondary (perhaps simplified) different programmable automation system (e.g. based on a programmable logic controller, PLC), or
- use of a secondary, hardwired analogue system.

The last alternative can in time prove to be costly e.g. for maintenance reasons. Its benefits for the increased reliability of the total system can also be set in doubt for two reasons. Firstly the independent behaviour from the programmable variant is not guaranteed and a common cause failure may be possible even in this case. Secondly the possible maintenance problems may jeopardize the reliability of the analog system and the total dependability of the whole system strived after may in time be left out of reach.

Dynamic testing is a generally accepted approach for proofing in practice that the target system behaves properly and meets its reliability requirements. This presupposes that the reliability requirements are not too high. Even fairly modest probabilistic reliability requirements will need thousands of test cases to reach acceptably high confidence level. Use of statistical inference from the dynamic test results requires that the test cases are generated to match the operational profile.

Several aspects need to be taken care of in dynamic testing. The main issues are an efficient environment where to run the test cases, automated generation of tests according to the operational profile, high quality logical model of the target system, and an intelligent comparator for finding out false target system responses.

It can be claimed that dynamic testing is more fitted for programmable automation systems than, for example, pure real-time controlling systems. Programmable automation systems resemble batch systems in the sense that execution of tasks is cyclic and their order is determined. Interrupts that increase system dynamics and at the same time internal nondeterminism, are seldom used. Thus dynamic black-box testing of the application may produce valuable evidence of the system application and by following the principles of statistical testing the quantification of the system reliability becomes possible.

REFERENCES


Digital Neutron Flux Instrumentation

The plant wide application of computer based safety systems is still in discussion. On the level of signal transmitters, microprocessor based equipment is already working within the reactor safety system of some german power plants, e.g. for neutron flux monitoring. The practical experiences on operational behaviour and reliability are very good.

Presented by: Willi Bucher
Hartmann & Braun, Werk München
Landsberger Str. 328, D-80687 München
Tel.: *49-89/58005-100, Fax -169

For: OECD/NEA
INTERNATIONAL WORKSHOP
MUNICH, 5 to 7 MARCH 1996
Digital Neutron Flux Channels

Features

Multiprocessor system with a very small, separated safety related core
Re-calculation with a separate processor
Very high degree of failure detection
Type tested acc. to KTA 3501/3505, DIN V VDE 0801, IEC 880
Remote check signals, electrical & numeric

lean signal processing
diversity
safe & reliable
qualified
testable
# Digital Neutron Flux Channels

<table>
<thead>
<tr>
<th>Type</th>
<th>Application</th>
</tr>
</thead>
<tbody>
<tr>
<td>DAK 250</td>
<td>pulse and intermediate range monitoring</td>
</tr>
<tr>
<td>DGK 250</td>
<td>power range monitoring (with n-ion chambers)</td>
</tr>
<tr>
<td>DVK 250</td>
<td>flux deviation monitoring with SPN-detectors</td>
</tr>
<tr>
<td>DWK 250</td>
<td>wide range monitoring</td>
</tr>
<tr>
<td>DSK 250</td>
<td>local power range monitoring (in core)</td>
</tr>
<tr>
<td>DMK 250</td>
<td>average power range and oscillation monitoring</td>
</tr>
<tr>
<td>NC 102 N</td>
<td>PC software for service and test</td>
</tr>
</tbody>
</table>
Architecture of the Neutron Flux Channels TK 250
Signal Pathes in Neutron Flux Channels TK 250 for PWR
Sizewell B Reactor Protection System - Software Design Goals

William D. Ghrist III
Process Control Division
Westinghouse Electric Corporation
200 Beta Drive
Pittsburgh, Pennsylvania 15238 USA

Abstract

The reactor protection and controls for the successfully operating Sizewell B nuclear plant were developed with particular attention to separating the application specific software functions from the reusable common functions that provide applications support (e.g., lag filter functions, voting logic) and computer system support functions (e.g., processing of analog input signals, inter-processor communications). This has important significance for the reviewability of the system. The protection and control functions themselves can be specified and implemented in a manner that is clearly derived from the top-level functional requirements. The complexity associated with the computer based system is largely confined to the common functions software modules. Decoupling and data hiding are important to the design and reviewability of these modules. Because they are reusable, the common functions modules benefit from the maximum application of development and verification resources, as well as from extensive operating experience throughout the system, as well as in other similar systems. Ultimately, however, it is important not to rely too strongly upon any single principle or method. A balanced application of many different theoretical and practical considerations is necessary to achieve the goal of a system that reliably performs the functions required of it.

Historical Background

The initial Westinghouse program to develop a software-based integrated reactor protection system began in 1975. This first generation system was based primarily on 8-bit microprocessors but also implemented some protection functions in analog circuitry. A full scale prototype was manufactured along with the development of the software to allow for qualification and validation testing. This first generation system was carried into the licensing process and resulted in a preliminary design approval as part of the RESAR-414 program. Although it was never implemented in a U.S. plant, this system design did become the basis for the first French digital instrumentation and control system as the result of a joint development effort and is in operation in the Paluel units.

The second generation safety system development began in the mid-1980’s and was based on 16-bit microprocessor technology. The program eventually evolved into the Nuclear Electric Sizewell B Primary Protection System (PPS) as the first plant application. The technology that was developed resulted in the Westinghouse “Eagle” product line, which is based entirely on digital technology. The first Eagle prototype integrated protection system was completed in 1987, for use as a hardware and software development test bed. This was followed by a Sizewell B specific prototype in 1989. The final Sizewell B Primary Protection System (PPS) was delivered to site in 1992.

Eagle technology has also been applied for protection system upgrade applications, known as Eagle-21, and for high integrity applications other than protection systems, such as Rod Control, Flux Mapping, and NSSS control. One of the largest Eagle applications is in the Sizewell B Integrated System for Centralised Operations (ISCO). ISCO is a replacement for a set of l&C originally planned to be supplied by another vendor. Westinghouse developed a special architecture based on existing building-block modules and delivered the system in 1992. Within ISCO the Nuclear Steam Supply System (NSSS) controls and safety-related control functions were implemented with Eagle technology. The balance of the plant controls and the plant information system were implemented using the
Westinghouse Distributed Processing Family (WDPF II). WDPF II is the Westinghouse standard digital I&C product line designed for general purpose industrial use.

The Sizewell B plant achieved criticality at the end of January, 1995 and received its rating certificate in September of 1995.

**Software Design Goals**

A number of important primary goals were identified when formulating the basic approach to the protection-system software design. These goals were based upon the recommendations of IEC 880, modern software engineering principles, contract specifications, and upon Westinghouse's extensive experience in the development of software for real-time control and safety systems. Among the goals were the following:

- To design the software architecture and operation to be compatible with verification and to implement it in such a manner that it can be thoroughly tested without great difficulty.

A number of important architectural features support this goal, e.g., modularity and decoupling of functions makes testing at the unit level more effective. The use of multiprocessing rather than multitasking, the avoidance of interrupts, and continuous rather than event-driven communication and processing of data all lead to highly deterministic operation. This makes it much easier to verify that the software will operate correctly under all conditions.

It is also important that all aspects of the design, from requirements to final code, be comprehensible, because it is by inspection that most errors are detected.

- To keep a sharp distinction between applications function software and computer systems support software.

The functional requirements of the applications software are derived directly from the top level functional requirements of the protection system. The functional requirements of the computer systems support software are created to support the system architectural design. They must be compatible with the top level protection function requirements, but are not derived directly from them. We believe that the design, implementation, and verification of software is most reliable when the functions derived from different sources are implemented in different modules.

- To make the applications functions as simple as possible.

The ideal is to have a one-to-one correspondence between elements of functional block diagrams and simple expressions or calls to common functions procedures. (Exceptions are in the areas of software initialization, self-testing, and fail-safe design decisions, since these are elements of the software design more than of the protection function.) We believe that this approach increases the probability that the protection function requirements will be correctly translated. It also improves the reviewability of the implementation, and consequently the ability to verify it. This approach also readily accommodates functional requirements that may change or that are not firmly defined until late in the project development.

- To hide the details of computer system services from the applications functions so that the applications software designer can concentrate on implementing the protection functions.

"Information hiding" simplifies the application software engineer's task. Errors are less probable if every designer on the project is not required to know the details of computer systems services software.

- To develop a set of common functions modules that support a wide range of applications functions that might be required.

These common functions include both computer systems services (analog input processing, interprocessor communications, etc.) and commonly used applications support algorithms (filters, time delays, comparators, etc.). It is typical that system design and implementation of a project must proceed before a complete and final set of functional requirements is known, so it is best to provide support for a wide range of commonly needed functions.
• To decompose the common functions into a number of major functions that will work together, but whose internal designs are as independent as practical.

In particular, the details of data structure design should be isolated as much as possible within individual modules. This allows each function to be developed separately, so that design decisions and design changes in each case have minimal impact upon the design of the other functions. A partial exception to this is the creation of a common approach to software self-testing (e.g., range checking of configuration and calibration data) and exception handling. This is basically, however, a common design approach, rather than an actual coupling of the internal designs of the modules.

• To concentrate as much of the software complexity as possible in the common functions. To compartmentalize the complexity into independent modules. To develop these complex modules early in the project so that the maximum amount of time and resources can be devoted to carefully designing and testing the most complicated parts of the software.

• To implement the common functions as reusable modules that can be done once, done right, and do not have to be recreated and reverified for each instance of their use.

This applies not only to reuse for different systems or plants, but to reuse within a given system. We did not want to design, implement, and test a series of essentially similar but specifically different analog input processors, shared memory handlers, etc., for each subsystem in the PPS. We considered this to be one of the major lessons learned from review of the first generation software development process. Inability to follow this approach was recognized as a weakness of the 414 project. The 414 design was limited by the microprocessor resources available at the time. The greatly improved hardware resources of the second generation system to a great extent were used to achieve this design goal.

In addition to benefiting from having maximum resources applied their development and verification, reusable modules also benefit from extensive operating experience throughout the system as well as in other similar systems.

• A related goal, recognized at the very beginning of the development process, is the principle that a program’s configuration and calibration data should be kept separate from the program code. This permits necessary modifications to be made without affecting the verification status of the program code, greatly reducing the verification burden associated with data changes.

A Caveat

As technology advances, goals must be re-evaluated in light of new capabilities. There is, however, one ultimate goal of the system design, implementation, and verification: to produce a system that reliably performs the functions defined in the system functional requirements. The individual goals are not ends in themselves, nor are any of the other software development principles and methods that may be employed; they are means to the ultimate goal. Many different means must be employed in concert to achieve that end. They are based on both theoretical and practical understanding of the methods that produce reliable systems. Each means is important, but no one of them alone will guarantee that the end goal will be achieved. They must be applied together, balancing the needs of each against the needs of the others, to achieve the end goal.