



ANP-10303NP
Revision 0

**SIVAT: TELEPERM XS™ Simulation Validation Test Tool
Topical Report**

June 2009

AREVA NP Inc.

(c) 2009 AREVA NP Inc.

Non-Proprietary

Copyright © 2009

**AREVA NP Inc.
All Rights Reserved**

The design, engineering, and other information contained in this document have been prepared by or on behalf of AREVA NP, a jointly owned subsidiary of AREVA and Siemens, in connection with customer requests to use the TELEPERM XS system in U.S. power plants. No use of or right to copy any of this information, other than by the NRC and its contractors in support of AREVA NP's request for review, is authorized.

The information provided in this document is a subset of a much larger set of know-how, technology, and intellectual property pertaining to the TELEPERM XS system. Without access and a grant of rights to that larger set of know-how, technology and intellectual property rights, this document is not practically or rightfully usable by others, except by the NRC as set forth in the previous paragraph.

For information address: AREVA NP Inc.
An AREVA and Siemens Company
3315 Old Forest Road
Lynchburg, VA 24506

U.S. Nuclear Regulatory Commission

Disclaimer

Important Notice Concerning the Contents and Application of This Report

Please Read Carefully

This report was developed based on research and development funded and conducted by AREVA NP, and is being submitted by AREVA NP to the U.S. Nuclear Regulatory Commission (NRC) to facilitate future licensing processes that may be pursued by licensees or applicants that are customers of AREVA NP. The information contained in this report may be used by the NRC and, under the terms of applicable agreements with AREVA NP, those customers seeking licenses or license amendments to assist in demonstrating compliance with NRC regulations. The information provided in this report is true and correct to the best of AREVA NP's knowledge, information, and belief.

AREVA NP's warranties and representations concerning the content of this report are set forth in agreements between AREVA NP and individual customers. Except as otherwise expressly provided in such agreements with its customers, neither AREVA NP nor any person acting on behalf of AREVA NP:

- Makes any warranty or representation, expressed or implied, with respect to the accuracy, completeness, or usefulness of the information contained in this report, nor the use of any information, apparatus, method, or process disclosed in this report.
- Assumes any liability with respect to the use of or for damages resulting from the use of any information, apparatus, method, or process disclosed in this report.

ABSTRACT

This Topical Report describes the Simulation Validation Test Tool (called SIVAT) developed by AREVA NP to support the development of project-related TELEPERM XS Application Software. This report describes:

- The TELEPERM XS simulation concept and SIVAT principles of operation and
- The high quality development process used to develop the SIVAT Tool.

The SIVAT software package allows the engineered I&C functionality to be tested by simulation. SIVAT uses the qualified TELEPERM XS SPACE Code Generator for generating simulation-capable code from the engineering data stored in the project database. This report shows that the I&C functionality represented in the Application Software can be effectively tested with the SIVAT Tool.

The objective is to prove that the functional requirements established by the process engineers have been translated into Function Diagrams (FDs) without errors, and that the software automatically generated from these FDs provides the functionality required in terms of input and output response. Process models can also be linked into the simulator to perform closed-loop tests.

Running pre-programmed test scripts ensures that simulation runs are traceable and repeatable. Test results are recorded in log files and plots for further evaluation. Additionally, simulation tests with SIVAT have shown to be an indispensable advantage when systems already in operation in the power plant need to be modified. In this case, simulation results prior to and after modification can be compared to verify that no inadvertent changes have been introduced to the I&C functions.

The use of the SIVAT Tool to support TELEPERM XS Application Software verification and validation has important benefits such as the early detection of Application Software faults that serves to reduce project risks earlier in the development process.

Nature of Changes

Revision	Section(s) or Page(s)	Description and Justification
0	All	Initial issue.

Contents

		<u>Page</u>
1.0	INTRODUCTION	1-1
2.0	DEFINITIONS	2-1
3.0	TELEPERM XS SIMULATION WITH SIVAT	3-1
3.1	Fundamentals of Simulation	3-1
3.2	SIVAT in the Application Software Development Process	3-2
3.3	Objectives for the SIVAT Tool.....	3-3
3.4	TELEPERM XS Simulation Methodology.....	3-5
3.5	TELEPERM XS Simulation Environment.....	3-7
3.6	Limitations of Simulation	3-20
3.7	Simulation in the Test Field with ERBUS TELEPERM XS	3-22
4.0	APPLICABLE REGULATORY GUIDANCE	4-1
4.1	Regulatory Guide 1.173 - Software Life Cycle Processes	4-1
4.2	Regulatory Guide 1.169 - Software Configuration Management,"	4-1
4.3	Regulatory Guide 1.168 - Software Verification and Validation	4-2
4.4	Regulatory Guide 1.171 - Software Unit Testing.....	4-4
4.5	Regulatory Guide 1.170 - Software Test Documentation.....	4-4
4.6	Regulatory Guide 1.172 - Software Requirements Specifications	4-6
4.7	Alignment with IEEE Std 1012-1998 Testing Activities	4-6
4.8	Conformance with IEEE Std 7-4.3.2-2003	4-8
4.9	Consistency with IEEE Std 1008-1987	4-10
4.10	Alignment with IEC 60880 Requirements for Tools	4-11
4.11	Alignment with Branch Technical Position 7-14	4-13
5.0	SIVAT MANAGEMENT PLAN	5-1
5.1	Use of SIVAT within TELEPERM XS Technology	5-1
5.2	Key Interfaces.....	5-2
5.3	Organization	5-5
5.4	Problem Reporting.....	5-8
6.0	SIVAT DEVELOPMENT PLAN.....	6-1
6.1	Use of TELEPERM XS Phase Model for SIVAT Development.....	6-1
6.2	SIVAT Development Documentation	6-4
7.0	SIVAT QUALITY ASSURANCE PLAN	7-1
7.1	Use of TELEPERM XS Quality Assurance Process for SIVAT Development.....	7-1
8.0	SIVAT SOFTWARE INTEGRATION PLAN	8-1
9.0	SIVAT SOFTWARE INSTALLATION PLAN	9-1
10.0	SIVAT SOFTWARE MAINTENANCE PLAN.....	10-1
11.0	SIVAT OPERATIONS PLAN	11-1

11.1	Application Software Testing with SIVAT	11-1
11.2	Limitations of Simulation	11-8
11.3	SIVAT Test Documentation	11-9
11.4	Summary of Application Software Integration Testing with SIVAT.....	11-11
12.0	SIVAT TRAINING PLAN	12-1
12.1	Training Organization and Responsibilities	12-1
12.2	Training Methods	12-1
12.3	Training Resources.....	12-2
12.4	Training Requirements.....	12-2
13.0	SOFTWARE SAFETY PLAN	13-1
13.1	Effect of SIVAT on Target System Code.....	13-1
13.2	Fidelity of SIVAT Simulation	13-1
13.3	Transparency of SIVAT Code Generation	13-3
14.0	SIVAT VERIFICATION AND VALIDATION PLAN	14-1
14.1	SIVAT Tool Verification and Validation Activities	14-1
14.2	Initial SIVAT Tool Validation Activities	14-1
14.3	Operating Experience with SIVAT	14-3
14.4	Independent Review of SIVAT	14-4
15.0	SIVAT CONFIGURATION MANAGEMENT PLAN	15-1
15.1	Use of TELEPERM XS Configuration Management Plan	15-1
15.2	SIVAT Life Cycle.....	15-2
15.3	Use of Software Program Manual Configuration Management Plan.....	15-3
16.0	SIVAT TEST PLAN.....	16-1
16.1	Background Information.....	16-1
16.2	Scope of Testing	16-1
16.3	Test Documentation.....	16-2
17.0	CONCLUSIONS	17-1
18.0	REFERENCES	18-1
18.1	U.S. Regulations	18-1
18.2	U.S. Regulatory Guidance	18-1
18.3	International Standards.....	18-1
18.4	U.S. Industry Standards.....	18-2
18.5	Regulatory Review Precedents.....	18-2
18.6	AREVA NP Documents.....	18-2

List of Tables

Table 4-1 - Alignment with IEEE Std 1012-1998 Testing Activities	4-8
Table 6-1 - TELEPERM XS Software Application Classes.....	6-2
Table 14-1 - TELEPERM XS Projects Verified and Validated with SIVAT	14-3

List of Figures

	<u>Page</u>
Figure 3-1 - TELEPERM XS Simulation Concept.....	3-2
Figure 3-2 - SIVAT in the Application Software Development Process	3-3
Figure 3-3 - TELEPERM XS Simulation Components.....	3-7
Figure 3-4 - SIVAT Components	3-8
Figure 3-5 - SIVAT Simulation Environment.....	3-9
Figure 3-6 - FDE Animation Mode.....	3-10
Figure 3-7 - Data Structure in the Simulator Database.....	3-12
Figure 3-8 - Message Simulation Concept	3-14
Figure 3-9 - Open-Loop / Closed-Loop Simulation.....	3-16
Figure 3-10 - SIVAT Test Process	3-17
Figure 3-11 - SIVAT Test Script Example	3-19
Figure 3-12 – SIVAT Test Result Example.....	3-20
Figure 3-13 - ERBUS TELEPERM XS Concept.....	3-23
Figure 5-1 - Procedure for Designing Hardwired and Digital I&C Systems	5-2
Figure 5-2 - Organizational Structure	5-5
Figure 6-1 - TELEPERM XS Phase Model Software.....	6-3
Figure 15-1 – Configuration Management Process Overview	15-2
Figure 15-2 – SIVAT Life Cycle	15-3

Nomenclature

<u>Acronym</u>	<u>Definition</u>
ASC	Assembling Center
BTP	Branch Technical Position
CAD	Computer Aided Design
CATS-SDE	Code Adaptation Tool for Simulator SDE
CCB	Change Control Board
CD-ROM	Compact Disc – Read Only Memory
CFR	Code of Federal Regulations
CoA	Configuration Administrator
CoM	Configuration Manager
CPU	Central Processing Unit
CR	Change Request
CRC	Cyclic Redundancy Checksum
DBB	Database Binder
DBE	Database Editor
DIN	Deutsches Institut für Normung (German Institute for Standardization)
EN	European Committee for Standardization
ERBUS	TELEPERM XS computer-assisted test system for TELEPERM XS test field application (test field simulator)
EUB	Expert User Board
FAT	Factory Acceptance Test
FB	Function Block
FD	Function Diagram
FDE	Function Diagram Editor
FDG	Function Diagram Group
FDGM	Function Diagram Group Module
GUI	Graphical User Interface
I&C	Instrumentation and Control
I/O	Input/Output
IC	Initial Condition
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronic Engineers
ISO	International Organization for Standardization
ITZ	Integration and Test Center
NRC	Nuclear Regulatory Commission

<u>Acronym</u>	<u>Definition</u>
NUPIC	Nuclear Procurement Issues Committee
QA	Quality Assurance
RTE	Runtime Environment
SCM	Software Configuration Management
SCU	Simulator Control Unit
SDE	Simulation Development Environment
SER	Safety Evaluation Report
SimDB	Simulator Database
SimEx	Simulator Executive
SIVAT	Simulation Validation Test Tool
SME	Subject Matter Expert
SMS	Service Monitor Server
SPACE	Specification and Coding Environment
SU	Service Unit
Tcl/Tk	Tool Command Language and associate toolkit
TXS	TELEPERM XS

1.0 INTRODUCTION

This Topical Report describes the Simulation Validation Test Tool (called SIVAT) developed by AREVA NP to support the development of project-related TELEPERM XS Application Software. This report describes:

- The concept of TELEPERM XS simulation and the principles of operation for the SIVAT Tool and
- The high quality development process used to develop the SIVAT Tool.

The SIVAT software package allows the engineered instrumentation and control (I&C) functionality to be tested by simulation. SIVAT uses the qualified TELEPERM XS Specification and Coding Environment (SPACE) Tool Code Generator for generating simulation-capable code from the engineering data stored in the project database.

The objective is to prove that the functional requirements established by the process engineers have been translated into FDs without errors, and that the software automatically generated from these FDs provides the functionality required in terms of input and output response. Process models can also be linked into the simulator to perform closed-loop tests.

Running pre-programmed test scripts ensures that simulation runs are traceable and repeatable. Test results are recorded in log files and plots for further evaluation. Simulation tests with SIVAT have shown to be an indispensable advantage when systems already in operation in the power plant need to be modified. In this case, simulation results prior to and after modification can be compared to verify that no inadvertent changes have been introduced to the I&C functions.

This topical report describes the concept of the TELEPERM XS simulation and the principle of operation of the SIVAT. This report shows that the I&C functionality represented in the Application Software can be effectively tested with the SIVAT Tool. The use of a NRC-approved simulation validation tool has been described in AREVA

NP document ANP-10272, Revision 1, Software Program Manual for TELEPERM XS™ Safety Systems Topical Report (Reference 29), which is referred to as the TELEPERM XS Software Program Manual.

The use of the SIVAT Tool to support TELEPERM XS Application Software validation has important benefits. The early detection of Application Software faults through validation testing with SIVAT serves to reduce project risks earlier in the development process.

AREVA NP requests that the NRC issue a Safety Evaluation Report that approves ANP-10303NP, Revision 0, SIVAT: TELEPERM XS™ Simulation Validation Test Tool Topical Report. AREVA NP intends to use the SIVAT Tool to support validation testing of TELEPERM XS Application Software developed in accordance with the TELEPERM XS Software Program Manual.

2.0 DEFINITIONS

ADD File

Text file containing commands for entering models and variables into the simulator database; serves as input file for the database tool of the simulator control system.

AREVA NP GmbH

Designation used in this report to refer to the AREVA NP organization responsibility for the TELEPERM XS System development. This organization is based in Erlangen, Germany.

AREVA NP (Inc)

Designation used in this report to refer to the AREVA NP organization responsibility for the design of U.S. projects using the TELEPERM XS System. This organization is based in Alpharetta, Georgia.

Application Software [IEEE Std 610.12 (Reference 15)]

Software designed to fulfill specific needs of a user. For TELEPERM XS systems the Application Software reflects the plant specific functionality of the TELEPERM XS I&C system. It is generated and documented by the TELEPERM XS SPACE tool.

Code [IEEE Std 610.12 (Reference 15)]

Computer instructions and data definitions expressed in a programming language or in a form output by an assembler, compiler, or another translator.

cmp_code

TELEPERM XS tool used for verification of the scope of a modification in Application Software code generated after implementing a specification change in the SPACE project database.

Component [IEEE Std 610.12]

One of the parts that make up a system. A component may be hardware or software and may be subdivided into other components.

Configuration [IEEE Std 828 (Reference 16)]

The arrangement of a computer system or component as defined by the number, nature, and interconnections of its constituent parts. In configuration management, the functional and physical characteristics of hardware or software as set forth in technical documentation or achieved in a product.

Configuration Control [IEEE Std 610.12]

An element of configuration management, consisting of the evaluation, coordination, approval or disapproval, and implementation of changes to configuration items after formal establishment of their configuration identification.

Configuration Control Board [IEEE Std 610.12]

A group of people responsible for evaluating and approving or disapproving proposed changes to configuration items, and for ensuring implementation of approved changes.

Configuration Identification [IEEE Std 610.12]

An element of configuration management, consisting of selecting the configuration items for a system and recording their functional and physical characteristics in technical documentation.

The current approved technical documentation for a configuration item as set forth in specifications, drawings, associated lists, and documents referenced therein.

Configuration Item [IEEE Std 610.12]

An aggregation of hardware, software, or both, that is designated for configuration management and treated as a single entity in the configuration management process.

Configuration Management [IEEE Std 610.12]

A discipline applying technical and administrative direction and surveillance to:

- Identify and document the functional and physical characteristics of a configuration item
- Control changes to those characteristics
- Record and report change processing and implementation status
- Verify compliance with specified requirements

Coverage

Method and indicators to assess that the functional features of the software have been comprehensively validated.

cpuload

TELEPERM XS load analysis tool used to analyze the loading on the central processor units.

Cyclic Redundancy Checksum (CRC)

Method applied for identification of data files using industry standard functions to produce a unique checksum. This checksum is used to identify and detect alteration of data during usage, transmission, or storage.

Discrepancies

During the software development life cycle, any difference or perceived difference discovered by various organizations in the later documents or code with the earlier requirements specified in other design documents. These discrepancies are initially documented on the Open Item list and are evaluated for further action.

H1

TELEPERM XS Ethernet network used for communication with TELEPERM XS Gateway and Service Unit (SU).

Interface [IEEE Std 610.12]

1. A shared boundary across which information is passed. This boundary includes design interfaces between design organizations (as interpreted by Regulatory Guide 1.169).
2. A hardware or software component that connects two or more other components for the purpose of passing information from one to the other.
3. To connect two or more components for the purpose of passing information from one to the other.
4. To serve as a connecting or connected component as in 2 above.

Interface Control [IEEE Std 610.12]

In configuration management, the process of:

- Identifying functional and physical characteristics relevant to the interfacing of two or more configuration items provided by one or more organizations
- Ensuring that proposed changes to these characteristics are evaluated and approved prior to implementation

K32

TELEPERM XS backplane bus used for communication inside TELEPERM XS computer units.

L2

TELEPERM XS PROFIBUS network used for communication between TELEPERM XS computer units.

Level 3

Functional requirements definition for the safety I&C technology to be implemented in TELEPERM XS. These requirements are defined in the Software Requirements Specification.

Level 4

Instrumentation and control requirements definition for TELEPERM XS. These requirements are defined in the Software Design Description.

Malfunction

Malfunction that is evoked in a simulated model.

MIC File

Machine language loadable code file.

netload

TELEPERM XS load analysis tool used to analyze the loading on the network connections.

Open Item

Any item which constitutes an error or anomaly from the required status or condition of a properly completed project. Each Open Item is given an identifier that is unique to the project and unit, as well as a record in a database. The entry contains information to track the life cycle of the item from initiation to final resolution.

rediff

TELEPERM XS tool used to detect differences in the functionality of Application Software in the redundant divisions of an I&C system. The tool performs an analysis of logics and parameter data specified for redundant system trains and identifies differences in functionality. The differences must be evaluated by an engineer to determine whether the differences are planned (engineered differences) or unplanned (errors).

reflist

A software program that creates CRC sums recursively for the subdirectories and files within a directory and outputs them in a list, including the date of the last change for the file. This method is used for identification of the TELEPERM XS system software, for project specific additions, for the Application Software implemented on an engineering platform (engineering workstation), and for software downloaded into the I&C system.

Regression Testing [IEEE Std 610.12]

Selective retesting of a system or component to verify that modifications have not caused unintended effects and that the system or component still complies with its specified requirements.

scanmic

'scanmic' is a TELEPERM XS software authentication tool. It analyzes the software configuration of loadable code (called MIC files). 'scanmic' is used to read the version strings of the Application Software components contained in a loadable MIC file from the MIC file itself, and calculate the CRC checksum for each software segment in the MIC file as well as the CRC checksum for the entire MIC file.

This information can be output to a list which serves to document the generated software version. Differences in the software configuration between the old version and the new version can be determined from these lists and then verified.

Software [IEEE Std 610.12]

Computer programs, procedures, and in some cases, associated documentation and data pertaining to the operation of a computer system.

Software Design Description [IEEE Std 610.12]

A representation of software created to facilitate analysis, planning, implementation, and decision making. The software design description is used as a medium for communicating software design information, and may be thought of as a blueprint model of the system.

Software Hazard [Based on IEEE Std 1228 (Reference 24)]

A software design error that could lead to an unintended operation or failure to operate when required.

Software Life Cycle [IEEE Std 610.12]

The period of time that begins when a software product is conceived and ends when the software is no longer available for use.

SPACE

The SPACE engineering system comprises the tools used for the engineering and maintenance of the TELEPERM XS I&C Application Software. In this context, engineering refers to the overall process of creating and testing the Application Software:

- Specification of the I&C functions and hardware topology
- Automatic code generation
- Software authentication, using *reflist* and *scanmic*
- Software loading
- Load analysis tool
- Database administration

System Software [IEEE Std 610.12]

Software designed for a specific computer system or family of computer systems to facilitate the operation and maintenance of the computer system and associated programs such as operating systems, compilers, and utilities.

Test Plan [IEEE Std 610.12]

A document describing the scope, approach, resources, and schedule of intended test activities. It identifies test items, the features to be tested, the testing tasks, who will do each task, and any risks requiring contingency planning.

Unit [IEEE Std 610.12]

1. A separately testable element specified in the design of a computer software component.
2. A logically separable part of a computer program.
3. A software component that is not subdivided into other components.

Validation [IEEE Std 610.12]

The process of evaluating a system or component during or at the end of the development process to determine whether it satisfies specified requirements. *Contrast with:* verification.

Verification [IEEE Std 610.12]

The process of evaluating a system or component to determine whether the products of a given development phase satisfy the conditions imposed at the start of that phase. *Contrast with:* validation.

Version [IEEE Std 610.12]

An initial release or re-release of a computer software configuration item that is associated with a complete compilation or recompilation of the computer software configuration item.

Verification and Validation [IEEE Std 610.12]

The process of determining whether the requirements for a system or component are complete and correct, the products of each development phase fulfill the requirements or conditions imposed by the previous phase, and the final system or component complies with specified requirements.

3.0 TELEPERM XS SIMULATION WITH SIVAT

This section describes the concept of TELEPERM XS simulation and the principle of operation of the SIVAT Tool. The section describes how the I&C functionality represented in TELEPERM XS Application Software can be completely verified with SIVAT.

3.1 Fundamentals of Simulation

The basic principle of TELEPERM XS simulation is relatively simple. One or more models are used to define the process or system to be simulated as realistically as possible. The Simulator Control System is used to implement the following tasks:

The TELEPERM XS simulation concept is shown in Figure 3-1.

The accuracy of the simulation results depends on how well the process or system is represented by the model or models. SIVAT models the TELEPERM XS system running project-specific Application Software. A model with regard to the simulation is a software function which was generally written in a higher-level programming language (C or FORTRAN) or generated using special CAD tools. The simulator control system can be compared to a very user-friendly, task-specific debugger that permits successive execution of the program code while simultaneously enabling the visualization and modification of program variables.

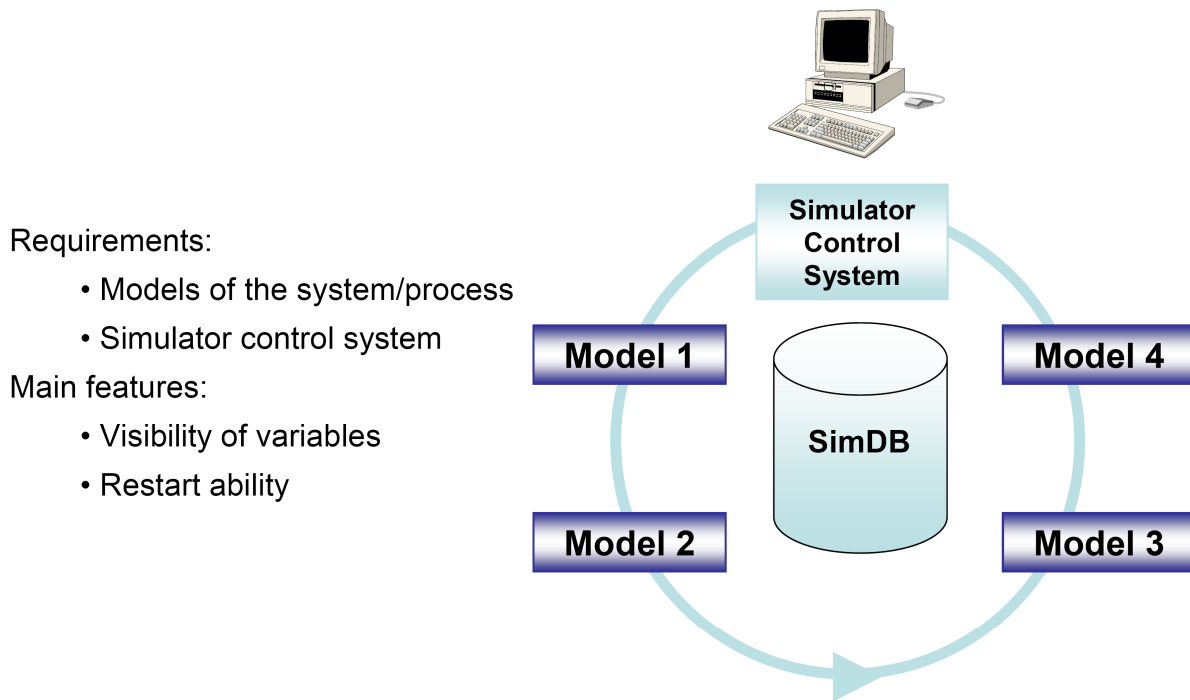


Figure 3-1 - TELEPERM XS Simulation Concept

3.2 SIVAT in the Application Software Development Process

The TELEPERM XS Application Software development process (shown in Figure 3-2) begins with the definition of the Software Requirements Specification (Level 3) for the I&C tasks which have to be implemented. This task definition is converted into the Software Design Description Function Diagrams (Level 4). The Software Design Description forms the basis for specifying the TELEPERM XS instrumentation and control system using the SPACE Function Diagram Editor (FDE). All data regarding the specified I&C functionality is stored in a project database. Thus the complete data set is available from one single source (single source principle). This project database is controlled by the Application Software Configuration Management Plan. The qualified SPACE Code Generator is used for automatic code generation. These tools generate the Application Software C Code, which is subsequently compiled and uploaded to the TELEPERM XS safety processors. The SIVAT Tool is used to support validation testing

of the Application Software in a simulation environment. The Application Software development process is fully described in TELEPERM XS Software Program Manual.

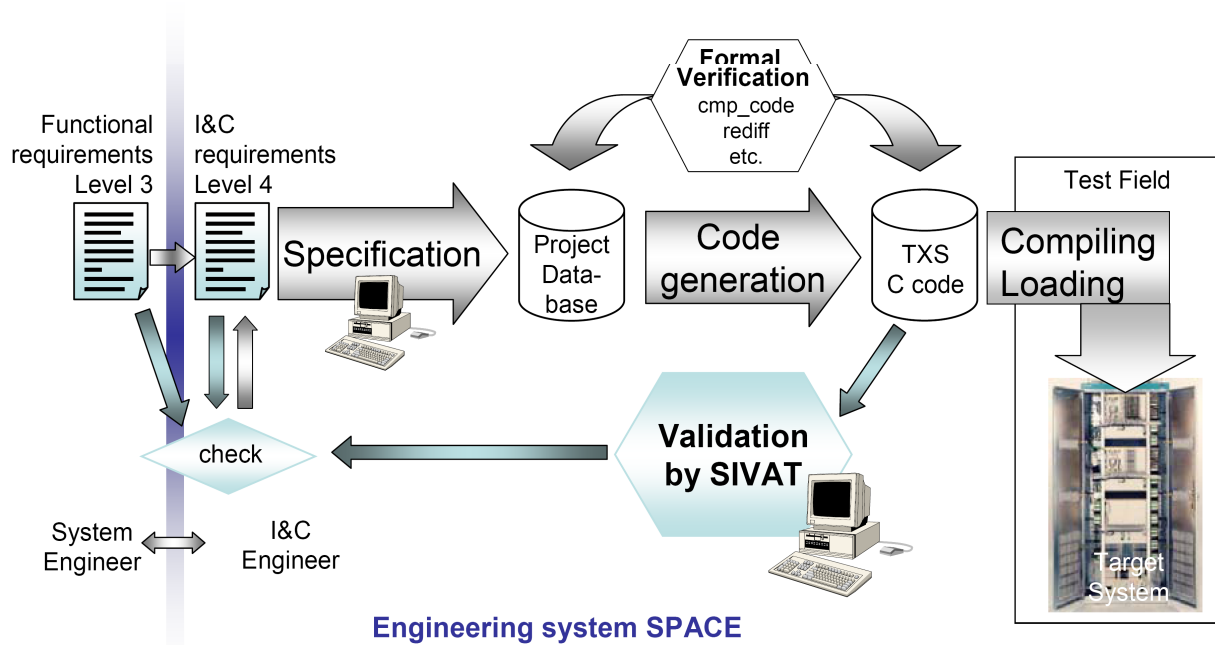
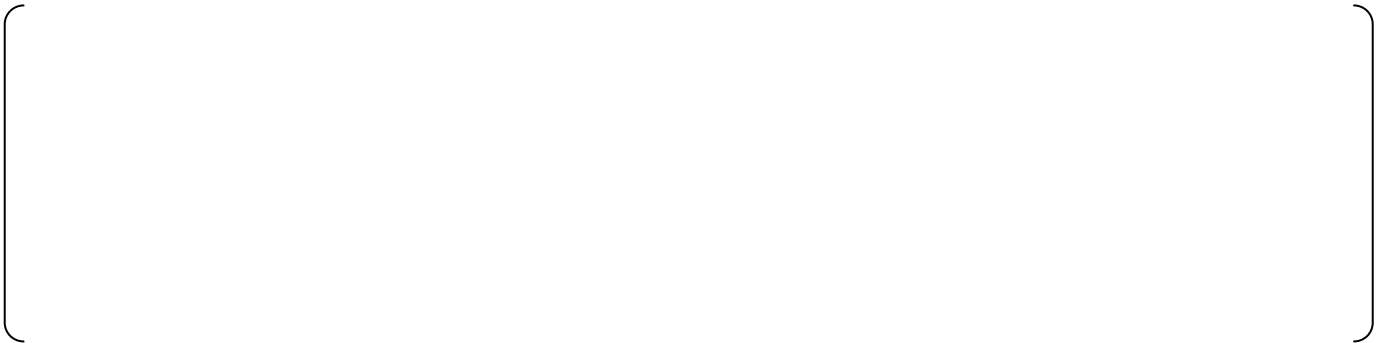


Figure 3-2 - SIVAT in the Application Software Development Process

3.3 Objectives for the SIVAT Tool

The SIVAT Tool is used to achieve the following objectives:



- Reproduce events in the simulator that occurred in installed TELEPERM XS systems.

To attain these goals, the SIVAT Tool has the following features:

- Utilization of a modern simulator control system (visualization of up to 400,000 signals/variables, restart ability).

- Restart capability using Initial Conditions (ICs).
- Easy integration of other models (e.g. process models).

- No real-time limitation (i.e., the simulation can run as fast as possible or can be run in slower time to support visual monitoring).

- Graphical user interface for easy handling of the automatic generation and user friendly interface with the simulation tool.

- Run on a LINUX workstation or a personal computer.
- Short time for generation of the simulator models.

3.4 TELEPERM XS Simulation Methodology

The I&C functionality of a TELEPERM XS System is achieved by interconnecting qualified Function Blocks (FBs). These FBs are stored for the target system in a precompiled library. The correct interconnection of FBs (assigning FB output and input signals) is implemented in the generated FD and Function Diagram Group (FDG) modules that are generated as C functions by the qualified SPACE Code Generator.

The initialization of the Application Software is done in exactly the same way as in the real I&C system. The initialization is done by setting one global variable that is

processed by the FBs. The original FB code is used for the initialization itself; thus, the following software modules are executing in the simulator:

The TELEPERM XS Simulation Components are shown in Figure 3-3.

A small part of the Runtime Environment (messages between TELEPERM XS CPUs and call of the FDGs), the FDGs, the Function Diagram modules, and the FBs contain the complete I&C functionality of a TELEPERM XS CPU. For a TELEPERM XS CPU model, exactly these components are simulated in SIVAT.



Figure 3-3 - TELEPERM XS Simulation Components

3.5 TELEPERM XS Simulation Environment

3.5.1 SIVAT Components

SIVAT consists of three components:

- The simulator control system Simulation Development Environment (SDE),
- The Code Adaptation Tool for Simulator (CATS-SDE), which generates the TELEPERM XS models and controls the complete automatic generation process of the simulator, and
- A graphical user interface (GUI) specific to the TELEPERM XS simulation.

The SDE simulator control system used is made up of three components:

- The database administration program Database Editor (DBE),
- The database preprocessor Database Binder (DBB), and
- The control system Simulator Executive (SimEx).

The SIVAT components are shown in Figure 3-4.

3.5.2 SIVAT Simulation Modeling

Each TELEPERM XS CPU is represented by a model in the simulation environment, ensuring that all internal signals and variables of a CPU model are stored in the simulator database. This is achieved by generating the corresponding ADD files for the simulator control system with the help of CATS-SDE.



The diagram area is currently blank, enclosed in a large rounded rectangular frame. It is intended to show the SIVAT components as mentioned in the text above.

Figure 3-4 - SIVAT Components

The CPU models are called by the control system according to their configured cycle time. SDE is able to process models with different cycle times. Models with shorter cycle times are called more frequently. Furthermore, CATS-SDE can generate an input and an output model. These models are used as a model for the measuring and actuating periphery of the TELEPERM XS System, by converting physical values into electrical values of analog signals and vice versa. This means that it is possible to simulate analog inputs with physical values. Furthermore, these models link external models (e.g., process models) with the TELEPERM XS models (see also Section 3.5.6).

There are various possibilities to simulate the input signals. Their values can be overwritten and are thus directly set in the SimDB. Or, transients can be specified by applying the ramp functionality. As another option, the values can be read in and assigned cyclically from data files, or the input values for TELEPERM XS can also be the output values of a process model. The SIVAT simulation environment is shown in Figure 3-5.



Figure 3-5 - SIVAT Simulation Environment

Just like all other signal and variable values, the values of the TELEPERM XS output signals are listed in the SimDB and can thus be visualized whenever required. Plot functions which support the writing of current signal or variable values in data files are also available.



Figure 3-6 - FDE Animation Mode



3.5.3 Simulator Database

All required signals and variables of the TELEPERM XS CPU models are stored in the SimDB. This enables the two main simulator features of signals and variables: visibility and overwritability. For this purpose, ADD files are generated by CATS-SDE, which list all variables used by the simulated TELEPERM XS CPUs. These files are then processed by the SDE tool DBE. The result is a database file that contains all TELEPERM XS model signals and variables (up to several hundred thousands).

By means of the ADD files, the respective required data structures are created in the size specified by the code generators. To be able to access the individual signals within these structures (pointers), links with the corresponding signal or variable name and the calculated offset relative to the beginning of the structure are created.

The

simulator database data structure is shown in Figure 3-7.

CATS-SDE ensures that all signals and variables that are calculated cyclically by the TELEPERM XS model are integrated in the database and thus stored at an IC. This guarantees that the simulation can be stopped at any time and an IC can be saved. By loading such an IC, it is possible to continue simulation at any time with the stored state.

Figure 3-7 - Data Structure in the Simulator Database

3.5.4 Simulation of Communication

Communication between the individual TELEPERM XS CPUs is implemented via messages that are generated by the SPACE Code Generator. The SPACE Code Generator specifies the message structure as well as the transmission path. Three paths are available for transmission depending on the network topology:

- The K32 backplane bus if the TELEPERM XS CPUs are arranged in the same subrack,
- The TELEPERM XS Profibus (L2) bus if there is an L2 network connection between the subracks, and
- The TELEPERM XS Ethernet (H1) bus if there is an H1 network connection between the subracks.

However, for simulation with SIVAT the transmission medium is irrelevant, since only the message structures are created. In the TELEPERM XS System, this structure is present in the sending, as well as in the receiving TELEPERM XS CPU. In the simulator it exists only once.

In the TELEPERM XS System, the Runtime Environment transfers the messages by utilizing functions of the MicroNET network system. These components are hardware-specific and are not simulated.

The message simulation concept is shown in

Figure 3-8.




Figure 3-8 - Message Simulation Concept

This form of communications simulation is completely sufficient for testing the specified I&C functionality. To examine the effects of failed communications links, the simulation of malfunctions is described in Section 3.5.5.

3.5.5 TELEPERM XS Malfunctions

To verify the effects of certain malfunctions on the specified I&C function, CATS-SDE generates three types of functions to simulate malfunctions that can be switched on or off via the corresponding flags in the SimDB:



(see

also Section 3.6).

3.5.6 Model Interfacing


The integration of the TELEPERM XS Application Software in SIVAT creates an open-loop simulation (i.e., the TELEPERM XS inputs can be stimulated by setting the corresponding signal values and by specifying transients). The behavior of the TELEPERM XS outputs can be recorded by generating data files and subsequently displayed as a graphic representation. The required response of the I&C to certain input signal modifications can thus be analyzed.

If a realistic feedback loop from the process or from one or several aggregates is required in order to evaluate the I&C behavior, separate models can be linked easily via a specified interface. This enables a partial or complete closed-loop simulation, and realistic events and disturbances can be simulated.

A simple example for an aggregate model is a valve model that is controlled by TELEPERM XS with an OPEN and CLOSE command and feeds the checkback signals and the current valve position back to TELEPERM XS. Such a model can be implemented with just a few lines of C or FORTRAN code and be integrated in SIVAT.

Principally, any number of separate models can be linked to the SIVAT simulator. Preconditions are that the models are available as C or FORTRAN functions, the model variables were stored in a simulator database using the SDE tool DBE, and the object code was compiled by the SDE-Tool DBB and the respective compiler. The object code is then available in a model library.

The open-loop / closed-loop simulation capability is shown in Figure 3-9.



The figure area is currently blank, enclosed in large rounded brackets. This indicates that the content of Figure 3-9 is missing from the page.

Figure 3-9 - Open-Loop / Closed-Loop Simulation

The link between TELEPERM XS CPU models and the process model (or other models) is implemented via the TELEPERM XS input or output model. Loose links to process models are also possible. This procedure is preferable if the overhead is too high for completely integrating a complex process model in SIVAT or if the models are not available with source code. In this case, however, more overhead is generated in the synchronization of the models and the complete simulator cannot be restarted.

3.5.7 SIVAT Test Principles

The basic procedure for implementing SIVAT tests is described below using a simple example. Although there are many possibilities to control the simulation interactively

The typical process for SIVAT tests is shown in Figure 3-10. The test cases are formulated in test scripts based on a test specification in which the test objectives, the basic test conditions, and the structure of the test cases are described.

Figure 3-10 - SIVAT Test Process

The simple example below demonstrates the procedure for testing a limitation monitor (COMP MIN). The set point value is 7.5 kV and the hysteresis is 0.1 kV. Each test case is divided into three parts:

1. **Initialization:** The simulation time is set to an initial value (typically 0.0 seconds) to consistently obtain the same time reference in the data file (plot file). Furthermore, the start conditions for the input signals are set (in this example to 8.0 kV for the input signal). For more complex test objects, the initialization is usually implemented by loading an IC that serves as the base for several test cases (e.g. 100% reactor power, failure-free condition).
2. **Plot Definition:** The signals to be saved in the data file are specified by means of a **plot**-command and added to the plot list. After the plot list has been defined, the data file is generated (command **plot-open**).
3. **Test Execution:** The individual test steps are now executed in succession. In this example, a ramp is executed from 8.0 kV to 7.0 kV in 5 seconds to trigger the limit value. Then a ramp is executed again, this time back to 8.0 kV in 10 seconds, to check the hysteresis. Finally the plot file is closed.

A SIVAT test script example is shown in Figure 3-11.

The result of the test case is now listed in the plot file (in this example voltage_limit.dat) and can, for example, be visualized with the SIVAT plot tool based on **gnuplot**. The result can be displayed on a monitor or printed out.

Figure 3-11 - SIVAT Test Script Example

The result of the test case is now listed in the plot file (in this example voltage_limit.dat) and can, for example, be visualized with the SIVAT plot tool based on **gnuplot**. The result can be displayed on a monitor or printed out.

An extract from a SIVAT plot file is shown in Figure 3-12. For each cycle (typically every 0.05 second), the plot function writes one line with the time stamp and the current values of the signals from the plot list to the plot file. The graphic representation of this plot file is shown next to the extract from the plot file. The results are evaluated manually based on the graphic outputs. Plot files can be consulted in cases of doubt.

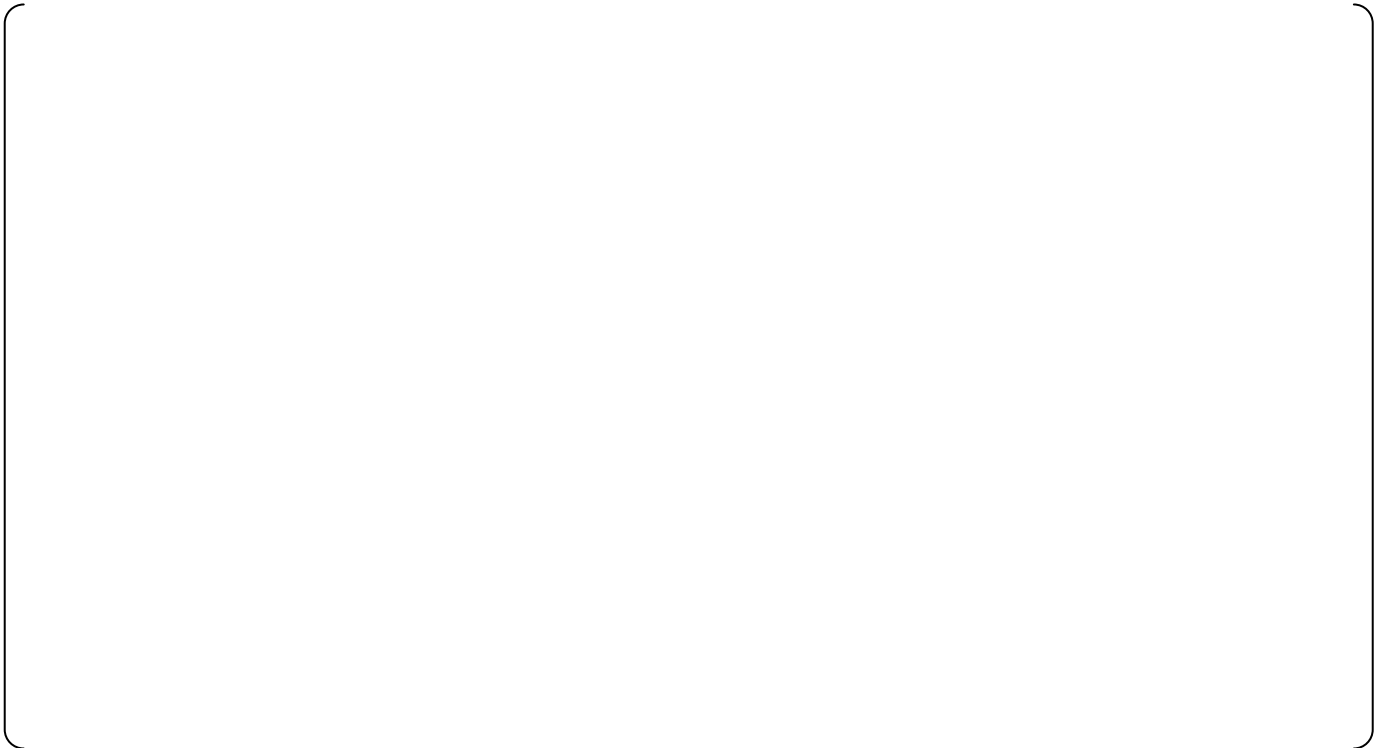


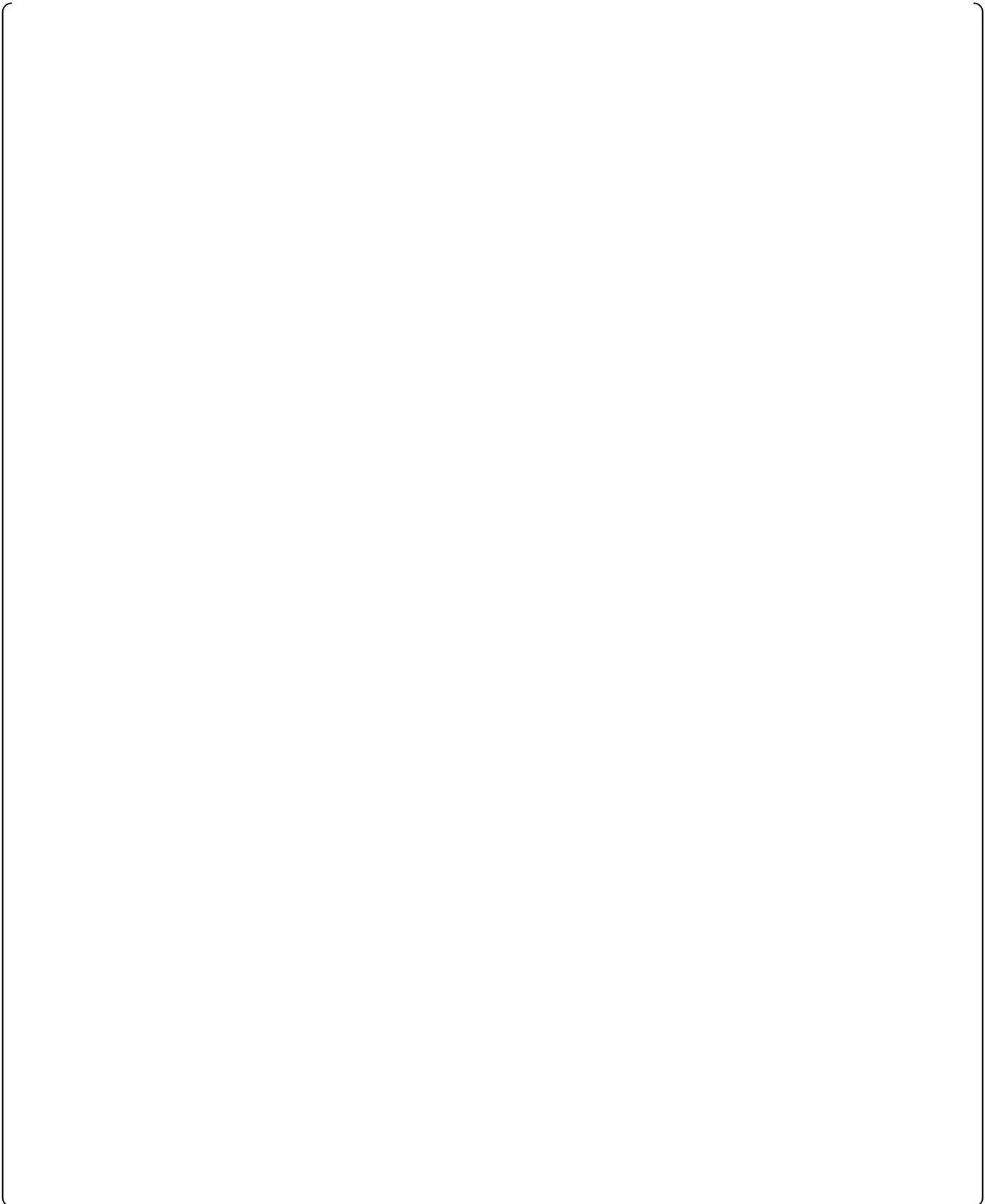
Figure 3-12 – SIVAT Test Result Example

3.6 Limitations of Simulation

As already mentioned in Section 3.1, the accuracy of the simulation depends on how well the models represent the systems. In Sections 3.2 through 3.5, the principles of SIVAT simulation were explained and it was demonstrated that the TELEPERM XS simulation is based on the original code of the Application Software (i.e., the simulation reflects the actual behavior of the specified I&C functions). Nevertheless, even TELEPERM XS simulations with SIVAT have their limitations that distinguish the

The

following system characteristics are not tested by SIVAT:



3.7 Simulation in the Test Field with ERBUS TELEPERM XS

3.7.1 Concept of the Test Field Simulator

The development of the ERBUS TELEPERM XS test field simulator system was based on existing and proven components. The following objectives were defined:

- Universal test system with a modular design that can be flexibly adapted to the respective requirements.
- Control via the network.
- Use of the simulator control system SDE already tried and tested with SIVAT.

The use of ERBUS in simulation testing with SIVAT is shown in Figure 3-13.

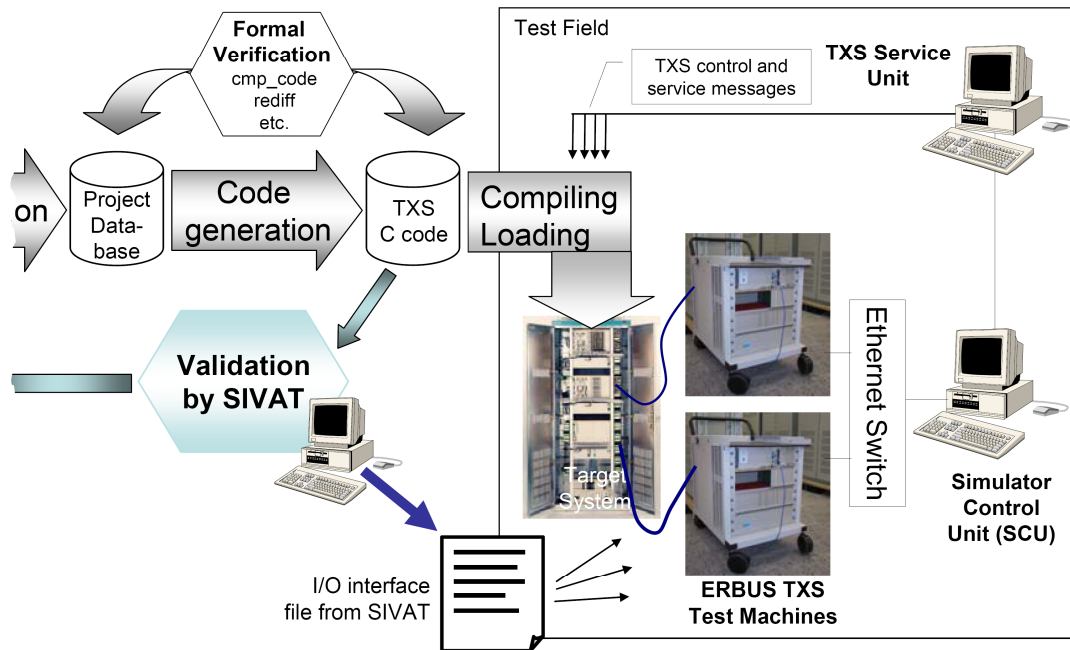


Figure 3-13 - ERBUS TELEPERM XS Concept

Following the manufacture of the cabinet, the TELEPERM XS system is set up in the test field. The objective is to commission the overall system and to test it. To do this, the TELEPERM XS Application Software is loaded onto the CPU modules in the TELEPERM XS cabinets and the TELEPERM XS inputs and outputs are linked with the ERBUS TELEPERM XS test machine outputs and inputs.

All test machines are connected to a central computer, the Simulator Control Unit (SCU). The SCU is the main component of the test field simulator. The same simulator control system that is used for SIVAT also runs on the SCU. Using a list of I/O signals that contains an assignment of the TELEPERM XS signals to the ERBUS TELEPERM XS channels, a simulator is generated. The SimDB contains a map of all TELEPERM XS I/O signals. There is one communications model for each connected test machine and Service Unit in the simulator. These models send or receive the respective assigned signals.

With the simulator in operation, the TELEPERM XS inputs are cyclically stimulated with the values in the SimDB via the ERBUS outputs, and the values at the TELEPERM XS outputs are cyclically entered into the SimDB.

Furthermore, the TELEPERM XS Service Unit (SU) can be linked with the SCU. In this case, the triggering of the TELEPERM XS inputs and outputs can also be implemented at the SU using the SMS.

The main task of the test field simulator is to stimulate and measure all inputs and outputs of a TELEPERM XS system. Depending on the size of the system, this can involve several hundred or even several thousand signals.

3.7.2 Model Interfacing

Due to the utilization of SDE as control system for ERBUS TELEPERM XS, the method of integrating process models could be adopted from SIVAT. The model interface is completely identical (i.e., the process models that were used for the simulation with SIVAT can also be used one-to-one for the test field simulation).

This possibility also permits a closed-loop simulation in the test field with minimum overhead and without time-consuming project-specific proprietary developments or additional hardware.

3.7.3 Limitations of Simulation with ERBUS TELEPERM XS

Unlike SIVAT, the ERBUS TELEPERM XS test field simulator does not model TELEPERM XS I&C. It only implements a hardware-based control of the inputs and outputs of the actual TELEPERM XS system using the communication models. If other models are integrated, they respond in exactly the same way as in the SIVAT simulator due to the same control system and interface.

The only limitation of the simulation with the ERBUS TELEPERM XS test field simulator results from the dynamic response of the overall system. The simulator, as well as the software on the test machines, operates with a fixed cycle time of 50 milliseconds. Like the TELEPERM XS CPUs, they are not synchronized. This results in system

dependent signal propagation delays between setting the value in the SimDB and outputting the corresponding hardware signal. The same applies to reading of hardware signals.

The delay between setting the signal value in the SimDB and triggering the corresponding value with a suitable short-circuited ERBUS output-input is approximately 250 milliseconds. Thus, no response time measurements can be done using this system.

4.0 APPLICABLE REGULATORY GUIDANCE

The applicable regulatory guidance documents are identified and addressed below.

4.1 Regulatory Guide 1.173 - Software Life Cycle Processes

Regulatory Guide 1.173 (Reference 9) endorses IEEE Std 1074-1995 (Reference 23). NRC reviewed the TELEPERM XS software life cycle process as part of the review of the TELEPERM XS Topical Report (Reference 28). NRC approved the TELEPERM XS Topical Report in a safety evaluation report (SER) issued in May 2000 (Reference 25). NRC made the following conclusion in the SER:

2.2.2.2 Software Management Plan

The software management plan for development of a Siemens digital safety system is the same procedure as used for all Siemens safety-critical software development projects. The software management plan is incorporated into Siemens Engineering Procedure FAW-1.1, "Software Life-Cycle Processes." FAW-1.1 specifies the management structure and the processes to be used in the project. This procedure is compatible to IEEE-1074, "Developing Life Cycle Process," and is, therefore, acceptable.

This conclusion is applicable to the SIVAT software, since the software was developed in accordance with the TELEPERM XS software development process described in Section 6.0.

4.2 Regulatory Guide 1.169 - Software Configuration Management,"

Regulatory Guide 1.169 (Reference 5) endorses IEEE Std 828-1990 (Reference 16) and IEEE Std 1042-1987 (Reference 22). NRC reviewed the TELEPERM XS software configuration management process as part of the review of the TELEPERM XS Topical Report. NRC made the following conclusions in the SER:

2.2.2.5 Software Configuration Management Plan

Configuration management activities are controlled by Siemens Engineering Procedure FAW-1.5, "Configuration Management," which outlines the procedures and tools for creating and implementing the configuration management structure and procedures. This procedure is compatible to IEEE-828, "Software Configuration Management Plan," and is, therefore, acceptable.

and

2.2.4 Configuration Management

The staff found that the configuration management procedure FAW-1.5 is compatible to IEEE-1042, "IEEE Guide to Software Configuration Management," and is, therefore, acceptable.

These conclusions are applicable to the SIVAT software, since the software was developed in accordance with the TELEPERM XS software configuration management process described in Section 15.0.

4.3 Regulatory Guide 1.168 - Software Verification and Validation

Regulatory Guide 1.168 (Reference 4) endorses IEEE Std 1012-1998 (Reference 20) and IEEE Std 1028-1997 (Reference 21). NRC reviewed the TELEPERM XS software verification and validation process as part of the review of the TELEPERM XS Topical Report. NRC made the following conclusions in the SER:

2.2.2.14 Software Verification and Validation Plan (SVVP)

The processes for conducting software verification and validation (V&V) activities are described in Siemens Engineering Procedure FAW-1.6, "Verification and Validation Plan." FAW-1.6 specifies the areas of application, the organizational responsibilities, requirements for IV&V

activities, and requirements for documentation. This procedure is compatible to IEEE-1012, "Software Verification and Validation Plans," and is, therefore, acceptable. The requirements for V&V are described in IEC-880-1986, "Software for safety Systems in Nuclear Power Stations," which Siemens has followed throughout the life cycle. IEC-880 is compatible to IEEE-7-4.3.2, "IEEE Standard for Digital Computers in Safety Systems of Nuclear Power Generating Stations," and is, therefore, acceptable.

NRC also documented in SER Sections 2.2.2.8, 2.2.2.10, and 2.2.2.12 that the TELEPERM XS software review process

described in Siemens Engineering Procedure FAW-4.2, "Reviews." This procedure describes the software review process, including responsibilities, review methods, the review processes, and activities to be performed after the review is completed. This procedure is compatible to IEEE-1028, "Software Review and Audit," and is, therefore, acceptable.

And, the TELEPERM XS software verification and validation process was further evaluated:

2.2.3 Development and V&V Organization and Process

The V&V processes are defined in Siemens Engineering Procedure FAW-1.6, "Verification and Validation Plan." This plan specifies all activities performed during the safety system development process. The responsibility for V&V activities is with the person responsible for the system or module development. This procedure is compatible to IEEE-1012, "Software Verification and Validation Plans," and is, therefore, acceptable. Siemens internal V&V processes were performed by members of the same development team, a member of another team within the digital I&C organization, or by employees outside the digital I&C organization. The person performing the internal V&V activity was not the same

person who generated the product to be reviewed. External IV&V activities were performed by TÜV organizations and iSTec.

This conclusion is applicable to the SIVAT software, since the software was developed in accordance with the TELEPERM XS software development process described in Section 6.0.

4.4 Regulatory Guide 1.171 - Software Unit Testing

Regulatory Guide 1.171 (Reference 7) endorses IEEE Std 1008-1987 (Reference 19). NRC reviewed the TELEPERM XS software test process as part of the review of the TELEPERM XS Topical Report. NRC made the following conclusion in the SER:

2.2.3 Development and V&V Organization and Process

Validation activities include testing the application to ensure it performs according to the system requirements. These activities are controlled by Siemens Engineering Procedure FAW-4.1, "Testing." Testing includes specifying the test requirements, performing the tests, and producing the test report. Testing includes module testing, component testing, and system testing in a simulated and real environment. This procedure is compatible to IEEE-1008, "Software Unit Testing," and is, therefore, acceptable.

This conclusion is applicable to the SIVAT software, since the software was developed in accordance with the TELEPERM XS software development process described in Section 6.0 and the test documentation requirements defined for the use of SIVAT in Section 11.3.

4.5 Regulatory Guide 1.170 - Software Test Documentation

Regulatory Guide 1.170 (Reference 6) endorses IEEE Std 829-1983 (Reference 17). NRC reviewed the TELEPERM XS software test documentation process as part of the

review of the TELEPERM XS Topical Report. NRC made the following conclusion in the SER:

2.2.2 Software Documentation

This section summarizes the software documentation associated with the TXS system development. The type tests of the TXS software components were performed in accordance with German standard KTA-Standard 3503. The principles of type testing and the test activities were defined from this standard. These were applied to the following areas: separation in the theoretical and practical tests, institutions to be involved in type tests, roles of these institutions in type tests, and documentation of type tests.

The content of the theoretical and practical tests is defined by the software standard DIN IEC-880. KTA standards also require that the present state-of-the-art be taken into account during the qualification. In addition to KTA-1401, which defines criteria for quality assurance systems, the following software standards were applied and verified:

- ISO-9000-3, "Management for Quality and Requirements of Quality Assurance,"
- IEEE-830, "Software Requirement Specifications,"
- IEEE-828, "Software Configuration Management Plan,"
- IEEE-1012, "Software Verification and Validation Plans,"
- IEEE-829, "Software Test Documentation,"
- IEEE-1008, "Software Unit Testing,"
- IEEE-1028, "Software Reviews and Audits," and

- ANSI/ANS-10.4, "Verification and Validation of Scientific Engineering Programs for the Nuclear Industry."

This conclusion is applicable to the SIVAT software, since the software was developed in accordance with the TELEPERM XS software development process described in Section 6.0 and the test documentation requirements defined for the use of SIVAT in Section 11.3.

4.6 Regulatory Guide 1.172 - Software Requirements Specifications

Regulatory Guide 1.172 (Reference 8) endorses IEEE Std 830-1993 (Reference 18). NRC reviewed the TELEPERM XS software requirements development process as part of the review of the TELEPERM XS Topical Report. NRC made the following conclusion in the SER:

2.2.2.6 Hardware and Software Specification

The procedure for controlling the hardware and software specifications is Siemens Engineering Procedure FAW-3.3, "Organization of the General Specification for SW and HW Components." This procedure governs the organization of the specifications for the digital safety systems created under this set of tools and processes. This procedure is compatible to IEEE-830, "Software Requirement Specifications," and is, therefore, acceptable.

4.7 Alignment with IEEE Std 1012-1998 Testing Activities

IEEE Std 1012-1998 describes four testing activities:

Component Testing: Testing conducted to verify the correct implementation of the design and compliance with program requirements for one software element (e.g., unit, module) or a collection of software elements. (Clause 3.1.3)

Integration Testing: An orderly progression of testing of incremental pieces of the software program in which software elements, hardware elements, or both are: combined and tested until the entire system has been integrated to show compliance with the program design, and capabilities and requirements of the system. (Clause 3.1.10)

System Testing: The activities of testing an integrated hardware and software system to verify and validate whether the system meets its original objectives. (Clause 3.1.26)

Acceptance Testing: Testing conducted in an operational environment to determine whether a system satisfies its acceptance criteria (i.e., initial requirements and current needs of its user) and to enable the customer to determine whether to accept the system. (Clause 3.1.1)

IEEE Std 1012-1998 Figure 2 shows a progression of test activities (i.e., component, integration, system, and acceptance testing) occurring during the development process (i.e., design, implementation, and test activities).

The combination of TELEPERM XS generic qualification testing and project-specific testing addresses all of the testing activities in IEEE Std 1012-1998, as shown in Table 4-1.

AREVA NP intends to use of the SIVAT Tool to support validation testing of TELEPERM XS Application Software as shown by the blue-shaded box shown in Table 4-1.

The decision to use the SIVAT Tool for validation testing requires that the SIVAT Tool development process conformance with clause 5.3.2 of IEEE Std 7-4.3.2-2003 (Reference 14).

Table 4-1 - Alignment with IEEE Std 1012-1998 Testing Activities

IEEE Std 1012-1998 Testing Activity	Generic TELEPERM XS Testing	Project-Specific Testing
Component Testing	X (hardware and software type tests, including Function Blocks)	Not Applicable (based on use of qualified hardware and software modules)
Integration Testing	X	Application Software: SIVAT for integration of Function Block modules Optional X (see Note 1)
		System Components: Pre-Factory Acceptance Test (FAT) prerequisites and procedure dry runs (manufacturing tests)
System Testing	X	X
Acceptance Testing	Not Applicable	(integrated in system testing, including FAT, based on use of qualified system components and development tools)

Legend: X indicates alignment with IEEE Std 1012-1998 testing.

Note 1 – Additional Application Software integration and functional test cases to validate engineering I&C functionality are added to the scope of system validation testing for the case where SIVAT testing is not used for Application Software integration and functional testing to satisfy IEEE Std 1012-1998 validation requirements. Validation testing with SIVAT is performed as an IEEE Std 1012-1998 Implementation Activity task.

4.8 Conformance with IEEE Std 7-4.3.2-2003

IEEE Std 7-4.3.2-2003 contains the following guidance for software tools used to support software development processes and verification and validation processes:

5.3.2 Software tools

Software tools used to support software development processes and verification and validation (V&V) processes shall be controlled under configuration management.

One or both of the following methods shall be used to confirm the software tools are suitable for use:

- a) A test tool validation program shall be developed to provide confidence that the necessary features of the software tool function as required.
- b) The software tool shall be used in a manner such that defects not detected by the software tool will be detected by V&V activities.

Tool operating experience may be used to provide additional confidence in the suitability of a tool, particularly when evaluating the potential for undetected defects.

IEEE Std 7-4.3.2-2003 has been endorsed by NRC in NRC Regulatory Guide 1.152 (Reference 3).

SIVAT conforms to the guidance in clause 5.3.2 of IEEE Std 7-4.3.2-2003. SIVAT was developed and is maintained within the TELEPERM XS configuration management process, described in Sections 4.2 and 15.0.

SIVAT was validated against test field data from the TELEPERM XS I&C modernization projects at the Unterweser and Philippsburg 1 nuclear power plants, as described in Section 14.1.

The limitations of SIVAT are clearly identified and understood, as described in Section 3.6. The system characteristics not tested by SIVAT are either tested during the TELEPERM XS generic qualification process, verified with other TELEPERM XS

analysis tools, or validated during system validation testing, as described in Section 11.2. It must also be understood that validation testing with SIVAT is just one component of the overall verification and validation program used for Application Software development, as described in the TELEPERM XS Software Program Manual.

AREVA NP has operating experience with the use of SIVAT for more than 20 project specific applications, as described in Section 14.3.

In addition, the SIVAT software was developed with a high quality development process as described in Sections 5.0 through 16.0 that were performed in accordance with the AREVA NP quality assurance (QA) program.

4.9 Consistency with IEEE Std 1008-1987

The benefit of Application Software validation testing with SIVAT is the early detection of faults. A balance is drawn between performing Application Software validation testing during the FAT later in the development process and performing Application Software validation testing with SIVAT earlier in the process. IEEE Std 1008-1987 recognizes that:

There are significant economic benefits in the early detection of faults. This implies that test set development should start as soon as practical following availability of the unit requirements documentation because of the resulting requirements verification and validation. It also implies that as much as practical should be tested at the unit level. (Paragraph B2.4)

The early detection of Application Software faults through validation testing with SIVAT serves to reduce project risks earlier in the development process.

The SPACE-generated code should be validated in a testing environment by means of the SIVAT simulation tool. The purpose of the simulation is to test the generated TELEPERM XS Application Software with regard to the way the process engineering tasks and/or the I&C function specification are implemented in the I&C. This is to

identify any Application Software errors as soon as possible, and prove the fulfillment of the requirements.

4.10 Alignment with IEC 60880 Requirements for Tools

IEC 60880 (Reference 12) contains requirements for the development and use of software tools. This standard was used by AREVA NP in the development of the SIVAT Tool. The relevant sections of IEC 60880 related to software tools are discussed below.

IEC 60880 Section 8.2.3.2.2 suggests that software written in application-oriented languages shall be verified to be functionally correct and consistent, for example by manual inspection or by the use of automated tools which allow simulated running of the software in a debug environment. SIVAT is the TELEPERM XS tool that supports the verification and validation of Application Software in a simulation environment.

IEC 60880 Section 8.2.3.2.5 requires that software tools used for verification or validation shall be qualified as required by Clause 14. The elements of Clause 14 are addressed separately below.

IEC 60880 Section 14.1.1 states that software tools are most powerful when they are defined to work co-operatively with each other. It also notes that care should be taken not to require tools to undertake tasks beyond their capability, for example, they cannot replace humans when judgment is involved. It goes on to state that when selecting a tool, the benefits and risk of using a tool must be balanced against the benefits and risk of not using a tool. And finally, it suggest that the important principle is to choose tools that limit the opportunity for making errors and introducing faults, but maximize the opportunity for detecting faults.

SIVAT was developed specifically as part of the TELEPERM XS technology system as described in Section 3.0 of this report. The limitations of SIVAT are clearly identified and understood, as described in Section 3.6. The system characteristics not tested by SIVAT are either tested during the TELEPERM XS generic qualification process, verified with other TELEPERM XS analysis tools, or validated during system validation

testing, as described in Section 11.2. The benefits testing with SIVAT are described in Section 11.0 of this report and are contrasted with the potential risk described in Section 13.0.

IEC 60880 Section 14.2 discusses the selection of tools and specifies that tools shall be selected to support the software engineering process. It states that the limits of applicability of all tools shall be identified and documented. It also states that tools shall have sufficient reliability to ensure that they do not jeopardize the reliability of the end product.

SIVAT has been specifically developed to support the TELEPERM XS Application Software development process, as described in Section 3.2 of this report. The limitations of SIVAT are clearly identified and understood, as described in Section 3.6. SIVAT was developed to have high reliability, as described in Section 4.8.

IEC 60880 Section 14.3.1 addresses the software engineering environment for tools. The SIVAT engineering process is described in Section 6.0 of this report. The results of the SIVAT development process are described Section 3.0.

IEC 60880 Section 14.3.2 addresses tool qualification. The qualification of the SIVAT Tool is described in Section 14.0 of this report.

IEC 60880 Section 14.3.3 addresses tool configuration management. Configuration management for the SIVAT Tool is discussed in Section 15.0 of this report.

IEC 60880 Section 14.3.4 addresses translators and compilers and is not applicable to the SIVAT Tool, since it is not a translator or compiler. The SPACE tool is the TELEPERM XS translator and compiler.

IEC 60880 Section 14.3.5 addresses application data tools and is not applicable to the SIVAT Tool, since it is not an application data tool.

IEC 60880 Section 14.3.6 discussed automation of testing and is not applicable to the SIVAT Tool. The use of the SIVAT Tool to support Application Software verification and

validation activities is described in Section 11.0 of this report. The Independent Verification and Validation Group designs test specifications, test cases, and test procedures to achieve the required coverage. The Independent Verification and Validation Group also analyzes the test results to established acceptance criteria. These aspects of testing with the SIVAT Tool are not automated.

Figure 5-1 in this report aligns with the lifecycle for application orientated software engineering shown in Figure C-1 shown in IEC 60880.

4.11 Alignment with Branch Technical Position 7-14

NRC Branch Technical Position 7-14 (Reference 10) addresses software tools at several points.

B.3.1 Acceptance Criteria for Planning

Acceptance Criteria for Resources Characteristics of Planning Documents

Methods/tools - It is important to remember that if the output of any tool can not be proven to be correct, such as may occur if the tool produces machine language software code, the tool itself should be developed or dedicated as safety-related, with all the attendant requirements.

SIVAT is not used to produce machine language software code. The SPACE Tool Code Generator is used for this purpose and is classified as safety-related.

B.3.1.2.3 Resource Characteristics of the SDP

Methods/tools involves a description of the software development methods, techniques and tools to be used. The approach to be followed for reusing software should be described. The SDP should identify suitable facilities, tools and aids to facilitate the production, management and publication of appropriate and consistent documentation and for the development of the software. It should describe the software development environment, including software design aids, compilers, loaders, and subroutine libraries. The SDP should require that tools be qualified with a degree of rigor and level of detail appropriate to the safety significance of the software which is to be developed using the tools. Methods, techniques and tools that produce results that cannot be verified or that are not compatible with safety requirements should be prohibited, unless analysis shows that the alternative would be less safe.

SIVAT conforms to the guidance in clause 5.3.2 of IEEE Std 7-4.3.2-2003, which ensures that it is qualified with the degree of rigor and level of detail appropriate for a validation tool for safety-related software.

B.3.1.2.4 Review Guidance for the SDP

Under the Resource Characteristics, the methods and tools to be used should be evaluated. Of particular interest is the method by which the output of software tools, such as compilers or assemblers, will be verified to be correct. The criteria from IEEE Std 7-4.3.2-2003 is that software tools should be used in a manner such that defects not detected by the software tool will be detected by V&V activities. If this is not possible, the tool itself should be safety-related.

SIVAT conforms to the guidance in clause 5.3.2 of IEEE Std 7-4.3.2-2003, which ensures that it is qualified with the degree of rigor and level of detail appropriate for a

validation tool. The limitations of SIVAT are clearly identified and understood, as described in Section 3.6 of this report. The system characteristics not tested by SIVAT are either tested during the TELEPERM XS generic qualification process, verified with other TELEPERM XS analysis tools, or validated during system validation testing, as described in Section 11.2. It must also be understood that validation testing with SIVAT is just one component of the overall verification and validation program used for Application Software development, as described in the TELEPERM XS Software Program Manual.

B.3.1.4.3 Resource Characteristics of the SIntP

Methods/tools refers to a description of the methods, techniques and tools that will be used to accomplish the integration function. The SIntP should require that integration tools be qualified with a degree of rigor and level of detail appropriate to the safety significance of the software which is to be created using the tools.

SIVAT is not used to produce or integrate software code. SIVAT conforms to the guidance in clause 5.3.2 of IEEE Std 7-4.3.2-2003, which ensures that it is qualified with the degree of rigor and level of detail appropriate for a validation tool.

B.3.1.5.3 Resource Characteristics of the SInstP

Methods/tools involves a description of the methods, techniques and tools that will be used to accomplish the installation function. The SInstP should require that installation tools be qualified with a degree of rigor and level of detail appropriate to the safety significance of the software which is to be installed using the tools.

SIVAT is not used to install software code. SIVAT conforms to the guidance in clause 5.3.2 of IEEE Std 7-4.3.2-2003, which ensures that it is qualified with the degree of rigor and level of detail appropriate for a validation tool.

B.3.1.10.3 Resource Characteristics for the SVVP

Methods/tools involves a description of the methods, equipment, instrumentation and tools used to carry out each V&V task. Test methods should be specified for unit, integration, validation, installation and regression testing. The SVVP should specify a process for selecting tools. The hardware and software environment within which the V&V tools are to be applied and any necessary controls should be described.

AREVA NP intends to use of the SIVAT Tool to support validation testing of TELEPERM XS Application Software. SIVAT conforms to the guidance in clause 5.3.2 of IEEE Std 7-4.3.2-2003, which ensures that it is qualified with the degree of rigor and level of detail appropriate for a validation tool.

5.0 SIVAT MANAGEMENT PLAN

SIVAT was originally developed during the years 1998-1999 to provide a simulation-based test environment to support the development of project-related TELEPERM XS Application Software.

5.1 Use of SIVAT within TELEPERM XS Technology

The TELEPERM XS Topical Report describes the simulator-based validation process for TELEPERM XS Application Software in Section 2.4.3.3.2. The simulator-based validation tool described in the report is SIVAT. The role of the simulator-based validation tool in the standard AREVA NP engineering process for TELEPERM XS project implementation is shown in Figure 5-1 (TELEPERM XS Topical Report Figure 2.8). The correctness of TELEPERM XS code generation in the course of application projects is covered by validation activities (i.e., software validation testing with SIVAT or during system testing).

TELEPERM XS Application Software is developed using the SPACE tool FDE. This tool is used to develop FDs and Network Diagrams. FDs specify the signal processing requirements for the system. Network Diagrams define the hardware components of the system and their logical interconnections. Software code is automatically generated from the FDs and Network Diagrams by the SPACE code generators. The project-specific TELEPERM XS System is developed from qualified hardware and software modules using the qualified development tools. Logical 'software integration' occurs at this stage.

SIVAT was designed to be used to support validation testing of the Application Software prior to installation into the target hardware. The validation test cases are created on the basis of the functional requirements defined for the Application Software. These validation activities serve to validate the detailed software engineering performed with the SPACE tool.

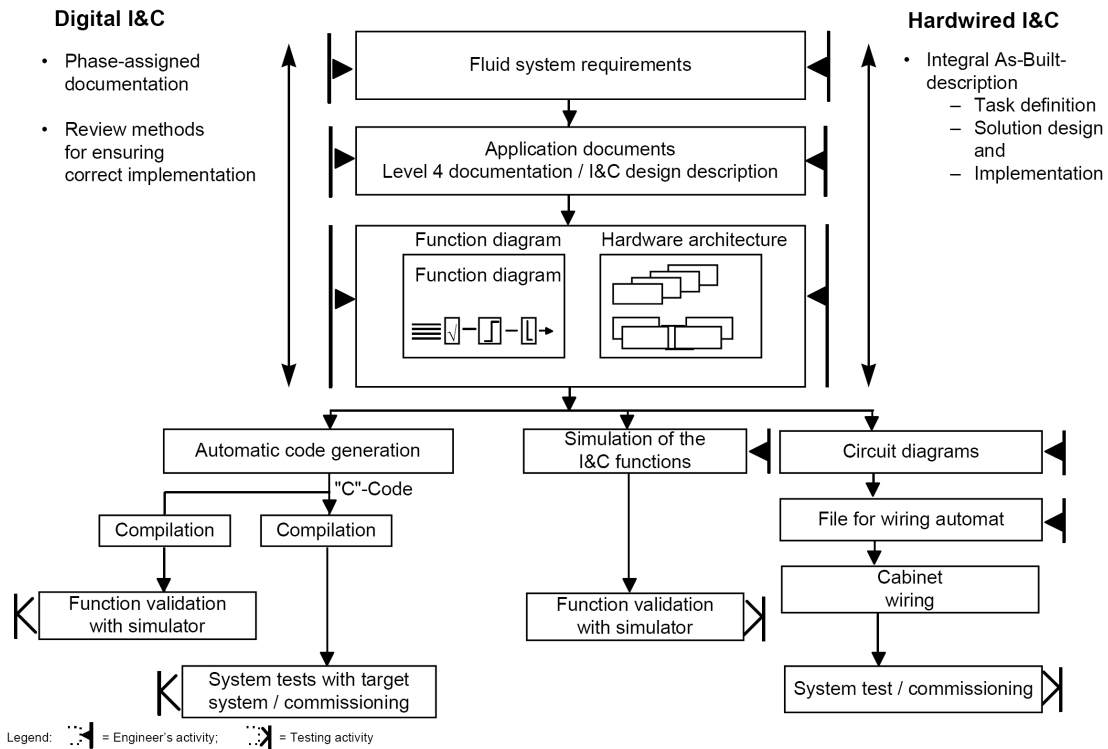


Figure 5-1 - Procedure for Designing Hardwired and Digital I&C Systems

The TELEPERM XS System Software development process is described in Section 3.2 of the TELEPERM XS Topical Report. The associated development process implementing procedures are summarized in Section 5 of the TELEPERM XS Topical Report.

5.2 Key Interfaces

The processes to develop and use the SIVAT Tool have key interfaces with other TELEPERM XS Topical Reports and the AREVA NP QA programs. Section 5.2.1 describes the interface with the TELEPERM XS Topical Report. Section 5.2.2 describes the interface with the TELEPERM XS Software Program Manual. Section 5.2.3 describes the interface with the AREVA NP QA programs.

5.2.1 Interface with the TELEPERM XS Topical Report

The TELEPERM XS Topical Report describes the generic development and qualification process for the TELEPERM XS digital I&C system.

SIVAT was developed under the same QA program and software lifecycle development process and procedures as described in the TELEPERM XS Topical Report. The SIVAT development environment is governed by the AREVA NP GmbH information security program. SIVAT was developed in accordance with the following AREVA NP GmbH procedures:

- Engineering Procedure FAW-TXS-1.1, Phase Model for the Development of Software Components for TELEPERM XS (Reference 31)

SIVAT was developed based on a requirements specification and technical specification document. The results of the SIVAT development process are described in Section 3.0 of this Topical Report.

- Engineering Procedure FAW-TXS-1.5, Configuration Management Plan for the TELEPERM XS System Platform (Reference 32)

Changes to the SIVAT Tool are controlled via FAW-TXS-1.5, which establishes requirements to ensure that changes are controlled, documented, and tested. The configuration management plan is described in Section 15.0 of this report.

- Engineering Procedure FAW-TXS-1.6, Software Verification and Validation Plan (Reference 33)

The validation of the product was performed with tests of a real TELEPERM XS application (data from a test) and the results of a SIVAT simulation of the same application. The verification and validation process for SIVAT is described in Section 14.0 of this report.

NRC reviewed these and other procedures as well as the QA program as part of the review of the TELEPERM XS Topical Report (see References 26 and 27). NRC approved the TELEPERM XS Topical Report in a safety evaluation report issued in May 2000.

Engineering Procedures FAW-TXS-1.1 and FAW-TXS-1.6 have not changed since the TELEPERM XS Topical Report was issued.

Engineering Procedure FAW-TXS-1.5 has evolved since the TELEPERM XS Topical Report was issued. The changes include the addition of a change control board to the configuration management process and the inclusion of additional detail describing configuration management tasks (e.g., more precise configuration identification).

5.2.2 Interface with the TELEPERM XS Software Program Manual

The TELEPERM XS Software Program Manual describes the overall lifecycle development process used for the development of project-specific TELEPERM XS Application Software.

AREVA NP intends to use the SIVAT Tool to support validation testing of TELEPERM XS Application Software developed in accordance with the TELEPERM XS Software Program Manual. Section 14.0 describes the overall software verification and validation plan for TELEPERM XS projects, which includes the use of NRC-approved simulation test tools such as SIVAT for validation testing. Section 15.0 describes the software configuration management plan that would be used to control the SIVAT software obtained from AREVA NP GmbH for use on for TELEPERM XS projects. Section 16.0 describes the software test plan for TELEPERM XS projects, which includes the use of NRC-approved simulation test tools such as SIVAT.

5.2.3 Interface with AREVA NP Quality Assurance Programs

All design work, products, and services provided for a TELEPERM XS project in the U.S. are performed to the requirements of the AREVA NP Quality Management Manual

(Reference 30), which implements the requirements of 10 CFR Part 50 Appendix B (Reference 2).

AREVA NP’s implementation of the Quality Management Manual is periodically audited by the Nuclear Procurement Issues Committee (NUPIC). The NUPIC program evaluates suppliers furnishing safety-related components and services and commercial grade items to nuclear utilities. In addition, NRC periodically conducts inspections of AREVA NP (including AREVA NP GmbH) as part of the supplier inspection program.

AREVA NP purchases TELEPERM XS System Software (including the SIVAT Tool) that is developed by AREVA NP GmbH under its QA program. AREVA NP GmbH is an approved supplier per AREVA NP’s approved supplier list.

5.3 Organization

The TELEPERM XS software development organization (AREVA NP GmbH) is organized in accordance with responsibility and authorities for the generic platform software lifecycle activities. The organizational structure is shown in Figure 5-2.

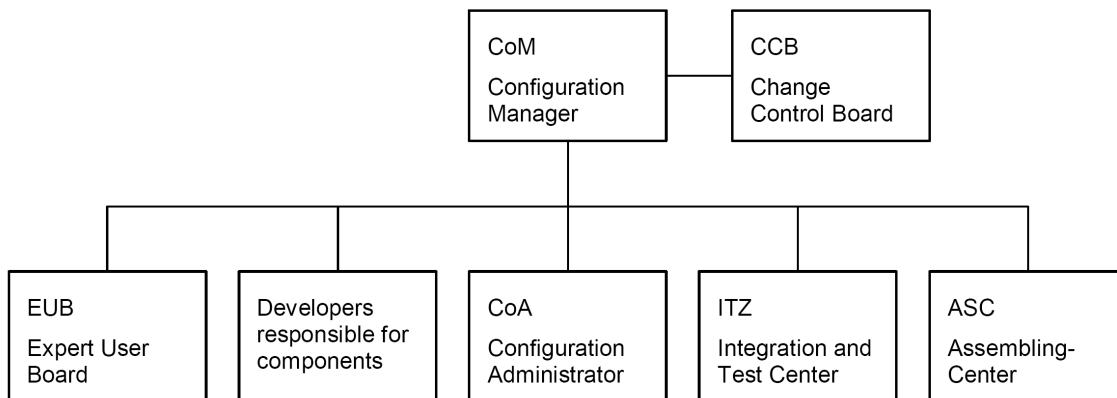


Figure 5-2 - Organizational Structure

5.3.1 Roles and Responsibilities

The following roles and responsibilities are defined for the TELEPERM XS platform software and software tool development activities.

- CCB: Change Control Board

The CCB is a body comprising personnel from management, marketing, project development (TELEPERM XS users), and system support that controls important changes and the further development of the TELEPERM XS system platform from a strategic and business viewpoint as well as from the user's viewpoint.

The CoM is appointed by the CCB and reports back to them at regular intervals.

- CoM: Configuration Manager

The CoM defines the product structure, determines the configuration planning, initiates and monitors all activities in the CM and decides on

- Content of a change plan
- Scope, implementation and approval of change requests (CRs)
- Release of components and packages
- Content and release of Product Information documents

The CoM is supported by the CoA.

- CoA: Configuration Administrator

The CoA is responsible for the activities of the change and release procedures and for continuous recording of the configuration documents, which also includes the generation of Product Information documents.

The CoA is appointed by the CoM.

- EUB: Expert User Board

The EUB is a body of experienced TELEPERM XS users, TELEPERM XS developers, the CoA and the CoM (CoM only when required). The composition of this body varies in accordance with the problem under discussion. The CoM is responsible for selection of the members. The CoA is responsible for its organization, the preparation of statements and reporting of results.

The tasks of the body are:

- To assess CRs that affect important aspects of the concept of the TELEPERM XS system platform
 - Development of strategies for individual components of the TELEPERM XS system platform and specification of the individual steps for following a strategy
 - Development of a long-term product strategy and determining the individual steps in its implementation
- Developers Responsible for Components

These persons are responsible for maintenance and further development of those configuration items (software or hardware components) within their responsibility, as well as for documenting implemented changes in the development documentation and in the change and release procedures. The area of responsibility also includes compliance with the development specifications and application of the relevant QA procedures including the CM activities.

- ITZ: Integration and Test Center

The ITZ integrates new configurations into packages and checks the mutual interface compatibility of configuration items and packages, as well as the proper functioning of commonly used software and hardware configurations.

- ASC: Assembling Center

The ASC supplies packages in traceable form to external and internal customers and performs software installation on customer's computers in accordance with the installation guidelines.

The activities of the software development organization are periodically audited by the Quality Assurance organization.

5.4 Problem Reporting

This section defines the responsibilities and requirements for identifying, processing, and resolving problems and discrepancies regarding the use of the TELEPERM XS SIVAT Tool for validation testing of Application Software developed by AREVA NP for use in safety-related I&C applications deployed in the U.S. The problem reporting process handles hardware and software component problems, nonconformances, verification and validation and testing anomalies, reporting of defects and noncompliance in accordance with 10 CFR Part 21 (Reference 1), as well as customer suggestions and potential product improvements.

Employees working on TELEPERM XS projects or using the TELEPERM XS software are responsible for following the methods and principles described in this section. Each employee who identifies a discrepancy, potential for improvement, a nonconformance, or a potential safety concern in connection with the SIVAT Tool must ensure that this deficiency or problem is clearly identified and reported, such as by recording error messages, producing screen shot copies, or creating a memory dump.

5.4.1 Corrective Action Program

The AREVA NP Corrective Action Program (Reference 34) establishes the process for promptly identifying and correcting conditions adverse to safety and quality in addition to providing the means for customer notification of these conditions. The Corrective Action Program also establishes the means for the identification and resolution of near miss problems, customer identified problems, and complaints. The condition report process implements a graded approach to managing adverse conditions. Condition report process actions are based on the significance, that is, Levels 1, 2, 3, or 4, associated with the adverse condition. An evaluation is performed and documented in the Corrective Action Program to determine if previous similar projects and customers are affected. Actions will be assigned to AREVA NP GmbH for evaluation and resolution for problems identified with the SIVAT Tool.

Items from the Open Items list (discussed in Section 5.4.2) are reviewed for conditions adverse to quality and safety, which are entered into the Corrective Action Program. Additionally, problems identified after delivery (see Section 5.4.4) are entered into the Corrective Action Program. The NRC reporting requirements of 10 CFR Part 21 are then evaluated. If required, a report is made in accordance with the AREVA NP procedures and affected customers are notified.

5.4.2 Open Items Process

Identified issues or Open Items are documented, and the organization responsible for the design evaluates and resolves them. Open Items are collected in a project-specific database as they are identified. Open Items associated with the SIVAT Tool are forwarded to AREVA NP GmbH for evaluation and resolution. Open Items that involve conditions adverse to quality and safety are entered into the Corrective Action Program.

For each Open Item, a brief description and a reference that describes the origin and the reason for the Open Item is documented in the database.

Although the Open Items database is the tool by which the Open Items are processed and managed, it does not satisfy the record keeping requirements of Appendix B of

10 CFR Part 50. Therefore, individual Open Item forms for project Open Items are stored in the AREVA NP Records Management System at the end of the project as a part of final documentation.

5.4.3 Discrepancies Identified by Testing

Discrepancies identified during testing are first recorded in a test discrepancy log and evaluated with the Software Design Group to determine if the problem resolution lies in revising the test plan or procedures or if the discrepancy is a software problem that may result in a modification. Problems identified with the SIVAT Tool are forwarded to AREVA NP GmbH for evaluation and resolution.

5.4.4 Discrepancies Identified after Release to the Customer

Discrepancies identified with the SIVAT tool after the release to the customer are to be handled in accordance with the Quality Management Manual and the Corrective Action Program. The NRC reporting requirements of 10 CFR Part 21 are evaluated. If required, a report is made in accordance with AREVA NP procedures and affected customers are notified.

6.0 SIVAT DEVELOPMENT PLAN

The Software Development Plan describes the life cycle activities for TELEPERM XS SIVAT Tool software development.

6.1 Use of TELEPERM XS Phase Model for SIVAT Development

The TELEPERM XS Topical Report describes the generic development and qualification process for the TELEPERM XS digital I&C system.

SIVAT was developed under the same QA program and software lifecycle development process as described in the TELEPERM XS Topical Report. SIVAT was developed in accordance with AREVA NP GmbH Engineering Procedure FAW-TXS-1.1, Phase Model for the Development of Software Components for TELEPERM XS.

SIVAT was developed based on a requirements specification and technical specification document. The results of the SIVAT development process are described in Section 3.0 of this Topical Report.

NRC reviewed this procedure and the QA program as part of the review of the TELEPERM XS Topical Report. NRC approved the TELEPERM XS Topical Report in a safety evaluation report issued in May 2000.

6.1.1 TELEPERM XS Software Application Classes

The TELEPERM XS software components are assigned to application classes, which serve to standardize various requirements on the development process and the verification and validation. The classes used in TELEPERM XS are shown in Table 6-1.

Table 6-1 - TELEPERM XS Software Application Classes

TELEPERM XS Software Application Classes:	
A = Safety Function	For safety functions, including Category A of IEC 61226 (Reference 11) (e.g., online software MICROS and Runtime Environment)
B = Safety Related Function	For signaling functions, other graded safety classes
C = Code Generation for Safety Function	Tools for generating software of classes A and B e.g. code generator for Function Diagram Groups (FDGCG) Runtime Environment (RTECG)
D = Engineering and Service	Tools and software solutions without safety functions (e.g. FBs not in Class A (FB add-on), gateway, simulator (SIVAT), service tools, internal tools, etc.)

The SIVAT software has been classified as Class D.

6.1.2 Phase Model for the Software Lifecycle

The phase model structures the process for manufacturing and maintaining software in a sequence of connected tasks and activities which when performed successively, results in completion and verification and validation that the software is fit for its purpose. Development of every TELEPERM XS software component, including the SIVAT Tool, is performed according to the requirements of the standard phase model.

The TELEPERM XS phase model is shown in Figure 6-1.

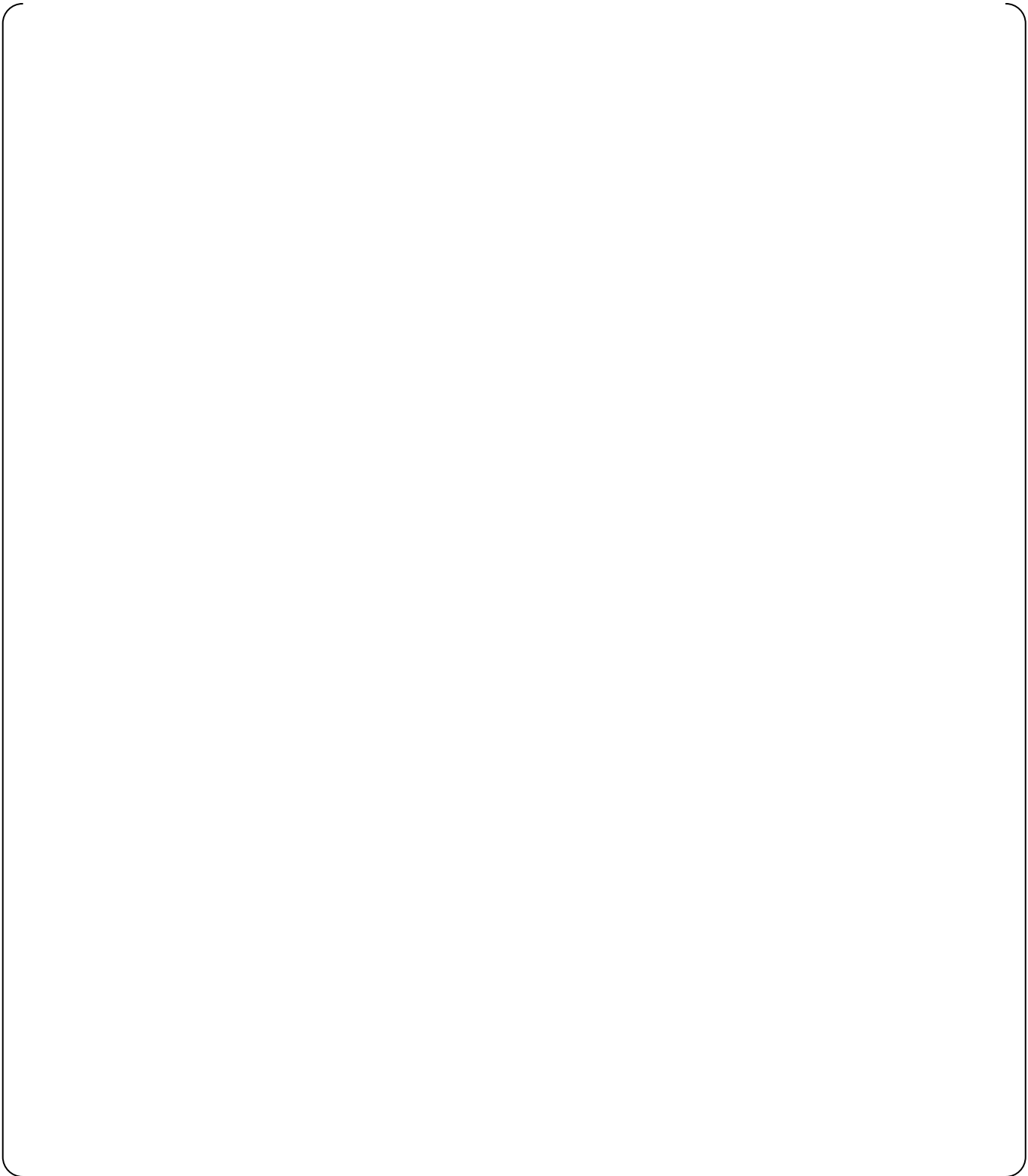


Figure 6-1 - TELEPERM XS Phase Model Software

There are two alternative phase paths:

- "Original phase path": Support and modifications for all TELEPERM XS software components that were available in 2005
- Phase path "2006": Software or firmware newly developed from 2006 onwards.

Procedures FAW-TXS-1.1, FAW-TXS-1.5, FAW-TXS-2.2, FAW-TXS-4.1, and FAW-TXS-4.2 always apply. As such they are applicable to the development of the SIVAT software.

Procedures FAW-TXS-1.6, FAW-TXS-2.1, FAW-TXS-3.3 to FAW-TXS-3.6 apply to modifications to all software components of TELEPERM XS Software Application Classes A to C that existed at the end of 2005. As such they are not applicable to the development of the SIVAT software, since it is Class D software.

Software developments for Application Classes A and C must be submitted to an external assessor for software qualification during development. Creation and internal / external checking of the development results is performed as specified by FAW-TXS-1.6. For software development for Application Class B, it must be determined on a case-by-case basis whether external assessment is necessary. SIVAT software is classified as Class D software, based on the requirements in FAW-TXS-1.1; consequently, no software type-testing, including third party assessment was required.

6.2 SIVAT Development Documentation

The following documents were created for the initial development of SIVAT:

- KWU NLL4/98/042, Rahmenlastenheft TELEPERM XS-Simulator (Frame Requirement Specification)
- KWU NLL4/98/068, Lastenheft SIMM (Requirement Specification)

- KWU NLLZ ST/99/023b, TELEPERM XS Pflichtenheft, Version 01.21: Generator CATS-SDE für die TELEPERM XS-Simulationsumgebung (Functional Specification)
- KWU NLL4/1998/180a, Vergleich der Ergebnisse des Prüffeldes (n-cpu) und der Offline (1-cpu) Simulationsumgebung (Test Results NPPs Unterweser and Emsland)
- KWU NLL4/2000/032, Auswertung der SIVAT Tests der LT-Funktionen (Test Results NPP Philippsburg)

7.0 SIVAT QUALITY ASSURANCE PLAN

The Software Quality Assurance Plan describes the necessary processes that ensure that the software attains a level of quality commensurate with its importance to safety.

7.1 Use of TELEPERM XS Quality Assurance Process for SIVAT Development

SIVAT was developed under the same QA program and software lifecycle development process as described in the TELEPERM XS Topical Report. NRC reviewed this procedure and the QA program as part of the review of the TELEPERM XS Topical Report. NRC approved the TELEPERM XS Topical Report in a safety evaluation report issued in May 2000. NRC made the following conclusions in the SER:

2.2.2 Software Documentation

This section summarizes the software documentation associated with the TXS system development. The type tests of the TXS software components were performed in accordance with German standard KTA-Standard 3503. The principles of type testing and the test activities were defined from this standard. These were applied to the following areas: separation in the theoretical and practical tests, institutions to be involved in type tests, roles of these institutions in type tests, and documentation of type tests.

The content of the theoretical and practical tests is defined by the software standard DIN IEC-880.

KTA standards also require that the present state-of-the-art be taken into account during the qualification. In addition to KTA-1401, which defines criteria for quality assurance systems, the following software standards were applied and verified:

-
- ISO-9000-3, "Management for Quality and Requirements of Quality Assurance,"
 - IEEE-830, "Software Requirement Specifications,"
 - IEEE-828, "Software Configuration Management Plan,"
 - IEEE-1012, "Software Verification and Validation Plans,"
 - IEEE-829, "Software Test Documentation,"
 - IEEE-1008, "Software Unit Testing,"
 - IEEE-1028, "Software Reviews and Audits," and
 - ANSI/ANS-10.4, "Verification and Validation of Scientific/Engineering Programs for the Nuclear Industry."

Among the standards referenced in the Standard Review Plan and the Branch Technical Positions, IEEE-7-4.3.2 gives specific requirements concerning software development. Most of these requirements are given by reference to the standards ASME NQA-10.4, IEEE-730, IEEE-828, IEEE-1012, and IEC-880. The requirements of ASME NQA-10.4 are covered by KTA -401, and the requirements of IEEE-730 are covered by ISO-9000-3. All other standards were directly applied in the development and evaluated in the type tests.

The key elements of the SIVAT software quality assurance process include the software management process described in Section 5.0, the software development process described in Section 6.0, the verification and validation process described in Section 14.0, the configuration management process described in Section 15.0, and the test plan described in Section 16.0.

8.0 SIVAT SOFTWARE INTEGRATION PLAN

The purpose of a Software Integration Plan is to provide a general description of the software integration process, the hardware/software integration process, and the goals of those processes.

The issue of SIVAT Tool integration with the corresponding SPACE tool issue is central to the SIVAT life cycle activities described in Sections 15.2 and 16.2.

The release documentation for each version of the SIVAT software provides information on the required prerequisite TELEPERM XS Software (e.g., SPACE tool, database management systems, and LINUX operating systems) and supported versions to support installation, as noted in Section 9.0.

9.0 SIVAT SOFTWARE INSTALLATION PLAN

The SIVAT Users Manual provides general instructions for loading the SIVAT software.

The SIVAT software is delivered on a CD-ROM. The installation process is controlled by an installation script and can only be performed by a system administrator.

The release documentation for each version of the SIVAT software provides a listing of the software files, including file size and CRC checksum. The release documentation also provides information on the required prerequisite TELEPERM XS Software (e.g., SPACE tool, database management systems, and LINUX operating systems) and supported versions.

10.0 SIVAT SOFTWARE MAINTENANCE PLAN

The purpose of a Software Maintenance Plan is to provide a general description of the software maintenance process and the goals of that process. In particular, the Software Maintenance Plan should list the general functions that the software maintenance organization will be expected to perform, and provide general information on obtaining field trouble reports. Maintenance should be limited to the process of modifying a software design output to repair nonconforming items or to implement pre-planned actions necessary to maintain performance. Modifications to improve performance or other attributes, or to adapt the design outputs to a modified environment, should be considered design changes.

The Software Maintenance Plan is not applicable to the SIVAT software. No maintenance is performed on the SIVAT software. There is no SIVAT system to be maintained by a maintenance organization. Changes to the SIVAT software are controlled by the TELEPERM XS software development organization AREVA NP GmbH. The configuration management process, including change control, is described in Section 15.0.

11.0 SIVAT OPERATIONS PLAN

The SIVAT Operations Plan provides a general description of the operation of SIVAT. The use of SIVAT to perform TELEPERM XS Application Software testing of I&C functionality and validation of system requirements is also described. The capability to simulate various TELEPERM XS malfunctions is described. In addition, the limitations of simulation are discussed.

SIVAT is used to support validation testing of TELEPERM XS Application Software developed in accordance with the TELEPERM XS Software Program Manual.

11.1 Application Software Testing with SIVAT

SIVAT can be used to perform Application Software integration and functional testing. Application software validation through SIVAT testing is one of the layers of validation testing that is used to ensure Application Software quality. It can also simulate certain

response to these faults is as intended. SIVAT enables the independent verification and validation engineer to compare the validation results to the software requirements specification. The Independent Verification and Validation Group uses the software requirements traceability matrix to ensure that software requirements have been tested.

The benefit of Application Software validation testing with SIVAT is the early detection of faults. A balance is drawn between performing Application Software validation testing during the FAT later in the development process and performing Application Software validation testing with SIVAT earlier in the process. IEEE Std 1008-1987 recognizes that:

There are significant economic benefits in the early detection of faults. This implies that test set development should start as soon as practical following availability of the unit requirements documentation because of

the resulting requirements verification and validation. It also implies that as much as practical should be tested at the unit level. (Paragraph B2.4)

The early detection of Application Software faults through validation testing with the SIVAT Tool serves to reduce project risks earlier in the development process.

The SPACE-generated code should be validated in a simulation test environment by means of the SIVAT Tool. The purpose of the simulation is to test the generated TELEPERM XS program modules with regard to the way the process engineering tasks and/or the I&C function specification are implemented in the I&C. This testing is performed to identify any Application Software errors as soon as possible and prove the fulfillment of the requirements.

The Independent Verification and Validation Group personnel should be independent from the Software Design Group according to NRC Regulatory Guide 1.171:

Criterion III, "Design Control," imposes an independence requirement for the verification and checking of the adequacy of the design, requiring that those persons who verify and check be different from those who accomplish the design. Therefore, independence is an additional requirement for software unit testing. Either those persons who establish the requirements-based elements for a software unit test must be different from those who designed or coded the software, or there must be independent review of the establishment of the requirements-based elements. The guidance in section A7 of Appendix A to IEEE Std 1008-1987 provides acceptable ways to meet this requirement for software unit testing. These independent persons must be sufficiently competent in software engineering to ensure that software unit testing is adequately implemented.

Application Software integration and functional testing with SIVAT can be used to validate engineering I&C functionality and satisfy IEEE Std 1012-1998 validation

requirements for Application Software integration testing. This Application Software integration testing using SIVAT is performed under the direction of the Independent Verification and Validation Group. The test plans, specifications, procedures, and reports are prepared in accordance with the Application Software verification and validation plan and 10 CFR Part 50 Appendix B requirements.

The functions of the TELEPERM XS Application Software can be tested module by module for each TELEPERM XS processor and can also be tested as an integrated software system. SIVAT can also be used in conjunction with system models to test the TELEPERM XS Application Software in a closed-loop test.

SIVAT has the capability to support both white box and black box testing of the TELEPERM XS Application Software. The following tests should be carried out using SIVAT:

- Validation of the I&C functions against the software requirements
- Testing of the specified I&C functionality as defined in the Software Design Description
- Simulated system behavior and failure response

The simulation in SIVAT is based on the code of the FDs and FDGs generated by means of the SPACE Code Generators. The tests cover the correct choice, integration, and parameterization of function modules. The reaction of the Application software to failures of I/O modules, TELEPERM XS processors, or data transfers can also be tested.

The goal of the project-specific simulation testing with SIVAT is to verify the correct implementation of all the functions and requirements specified in the Software Requirements Specification. Additionally, simulation testing with SIVAT will verify that Application Software functionality, specified in the Software Design Description, is

tested to validate that the software elements correctly implement software requirements. As a minimum the criteria for this determination are:

- Compliance with functional requirements.
- Performance at boundaries, interfaces, and under stress and error conditions.

The proper functionality of the project-specific Application Software is tested to validate the following standard TELEPERM XS characteristics:

- Test results must be verified from start of test until the completion of the test in order to ensure that no unexpected intermediate results are present.
- Signals must be handled in a manner to ensure spurious alarms are not generated by the software.
- Correct setting of FB parameters must be checked against software requirements.

11.1.1 Test of the Required I&C Functionality

The aim of the Functional tests is to validate the correspondence of the FDs developed with SPACE against the I&C requirements (Software Requirements Specification and Software Design Description). Inputs, outputs, and/or internal signals are simulated in the SIVAT test environment (e.g., by using scripts, signal generators, etc.). The signals and the system responses are recorded and compared to those stated in the test specification. The result of the comparison is documented.

The correct processing of FDs, the parameter settings of function blocks, the signal exchange between redundant channels, the correct generation of output, status, and alarm signals are all tested.

The test cases are adjusted to the modular structure of the FDs. Separate test cases are developed for each I&C function. The test cases can be subdivided for ease of testing. Each I&C function is validated. In the scope of function tests, all input and

output signals as well as the signal paths (logic connections in the FD) are covered by tests. Functions which are not validated in the scope of this phase of testing are identified for later validation in the test field.

For integration testing of the TELEPERM XS Application Software, the tests are preferably carried out in steps by starting with the inspection of small units, incrementing up to the inspection of larger units, such as:

- Tests of partial functions (e.g., of a sub-module)
- Test of the overall module
- Test of complete I&C functions

The tests to be carried out using SIVAT are designed such that they can be repeated to support validation of future revisions to the TELEPERM XS Application Software.

11.1.2 System Level Simulation Tests

The aim of this phase is the validation of the I&C functionality at the simulation system level. The test cases are derived from the design basis event protection requirements and test the system response as a whole. Interactions between I&C functions and redundancies are included in the test. Tests of control functions whose dynamic behavior is not significantly affected by the feedback of the system can be carried out as open loop tests. If significant dynamic feedback from the target system needs to be considered, as in case of standard functions, closed loop tests are used. If closed loop tests are required the availability of dynamic plant models is preferred; otherwise, simplified models can be used.

11.1.3 TELEPERM XS Malfunctions Simulation

The software is tested with regard to the requirement specification for postulated errors. The cases to be checked are analyzed and the checks defined according to requirements. The test program is defined together with the customer.

In the plant-specific tests, the effects of failures to the I&C functions are tested. The test cases include computer and communication failures. The effects of the tested errors are analyzed. Alarm and status signal processing are also tested. Tests cases can be developed to check the system behavior involving cross channel communications and function interaction of signals exchanged between redundancies and any failures associated with these signal exchanges. Functional tests under error conditions are performed.

Communication between the individual TELEPERM XS CPUs is implemented via messages that are generated by the SPACE Code Generator. The SPACE Code Generator specifies the message structure as well as the transmission path. Three paths are available for transmission depending on the network topology:

- The K32 backplane bus if the TELEPERM XS CPUs are mounted in the same subrack,
- The L2 bus if there is an L2 network connection between the subracks, and
- The H1 bus if there is an H1 network connection between the subracks.

communications simulation is completely sufficient for testing the specified I&C functionality. The effects of failed communications links on the I&C functionality are tested.

To verify the effects of certain malfunctions on the specified I&C function, SIVAT generates three types of faults to simulate malfunctions that can be inserted or removed:

described in Section 11.2.

is therefore not possible, as

11.2 Limitations of Simulation

TELEPERM XS simulation by SIVAT is based on the original code of the Application Software (i.e., the simulation reflects the actual behavior of the specified I&C functions). Nevertheless, even TELEPERM XS simulations with SIVAT have their limitations that distinguish the simulation environment from the actual TELEPERM XS system. Some of these limitations can only be overcome at great expense, others are resolved by formally verifying the I&C specification (analysis tools) independently of the simulation. The following system characteristics are not tested by SIVAT:

11.3 SIVAT Test Documentation

Test Specifications are prepared for each Test Case. Test Specifications incorporate the Test-Design Specification and Test-Case Specification into a single document and conform to IEEE Std 829-1983 and IEEE Std 1008-1987.

Test Procedures are prepared for each Test Case. The Test Procedures contain test scripts that implement the test cases defined in the Test Specifications. The Test Procedure verifies that the correct versions of the project-specific TELEPERM XS application software, SIVAT, and test scripts are used for the testing. The Test Procedures conform to IEEE Std 829-1983 and IEEE Std 1008-1987.

The Independent Verification and Validation Group verifies the SIVAT Test results by reviewing the Test Log, Test Incident Report, and Test Summary Report to ensure that they demonstrate that the system satisfies the criteria of the SIVAT Test Plan and Test Procedures. The Independent Verification and Validation Group verifies the correct versions of software were used in the SIVAT Test.

The Test Case is considered failed if the test script has a syntax error that prevents the script from running or if the test script or the Test Specification is found to be in error (i.e., the results of the test do not match the predicted results described in the Test Specification).

Any errors encountered while performing the test will be documented in the Test Log and Test Incident Report.

The suspension criteria and resumption requirements used for software validation testing are:

- If a discrepancy is found during test execution, the discrepancy is documented in the Test Log and the Test Incident Report and, if warranted, the testing resumes.
- A disposition of the discrepancies logged will determine if the discrepancy affects the Test Specification, Test Procedures, Software Requirements Specification, or the project-specific Application Software.
- If a discrepancy is found while comparing the plot data to the expected results, the discrepancy is recorded, evaluated, and resolved. The discrepancy is recorded in the Test Incident Report and the review of test results continues.
- When a discrepancy is detected that affects the affected design documents or the project-specific Application Software, an Open Item is created and the discrepancy is resolved.
- During review of the test results, all discrepancies are recorded in the Test Incident Report.
- Test reruns may start after required changes to the affected design documents and project-specific Application Software have been implemented and the Test Specifications and Test Procedures have been updated to the new design.
- Test reruns are performed on all sections of the Test Specification determined necessary and recorded in the Test Incident Report.

The pass/fail criteria used for system/software validation testing are:

- A Test Item is considered successfully passed when the results of the test match the expected results described in the Test Specification with no unexpected intermediate results.
- A test Item containing unexpected results may be considered to be successfully passed if the evaluation of the unexpected result concludes that the TELEPERM XS Application Software is functioning correctly. Disposition of the item is documented and preserved in the Test Incident Report. Under these conditions, a retest of the item will not be necessary.
- A Test Item is considered failed if the test script has a syntax error that prevents the script from running or if the test script or the Test Specification is found to be in error (i.e., the results of the test do not match the predicted results described in the Test Specification).

The Open Items process, as described in Section 5.4.2, is used to document any discrepancies identified during software validation testing. The project verification and validation report lists any verification and validation discrepancies or problems discovered during the software validation tests, and associated anomaly evaluations.

11.4 Summary of Application Software Integration Testing with SIVAT

Testing with SIVAT is optional but it is the preferred approach to TELEPERM XS Application Software integration testing. This approach is preferred because it leads to early detection and correction of Application Software faults, which serves to reduce project risks earlier in the development process.

Testing with SIVAT can serve as module or unit testing (i.e., FD or FDG testing). It can also serve as integration testing of the TELEPERM XS Application Software (i.e., testing of the Application Software for all TELEPERM XS modules working together) within the limitations of simulation. Additional testing is performed as part of the manufacturing tests to address the limitations of simulation testing.

The SIVAT test cases are designed such that they can be repeated to support validation of future revisions to the TELEPERM XS Application Software.

12.0 SIVAT TRAINING PLAN

The Training Plan describes the method of ensuring that training needs for the use of the SIVAT Tool for TELEPERM XS Application Software testing are achieved. The Training Plan provides a general description of the training organization and the basic training responsibilities. It also describes the basic training methods and primary training resources. The specific training requirements for use of the SIVAT Tool are defined.

12.1 Training Organization and Responsibilities

The AREVA NP (Inc) Training Group is responsible for conducting employee training on the use of the SIVAT Tool. The AREVA NP (Inc) Training Group reports to the AREVA NP (Inc) department manager. The AREVA NP (Inc) Manager has the overall ownership of training within the AREVA NP (Inc) department. The AREVA NP (Inc) Training supervisor is responsible for oversight and implementation of the AREVA NP (Inc) training program.

The training process is modular in nature and supports delivery of specific training for the needs of the various groups within the AREVA NP (Inc) department. The training is provided in accordance with AREVA NP administrative requirements

12.2 Training Methods

Training is normally implemented using one or more of the following forms of delivery:

- Instructor led - Instructors are either formally trained in instructional techniques, or SMEs under the guidance of the training supervisor. Instructor led training is presented in a classroom format using slides, overheads, or other media along with a student handout.
- Hands on - Hands on training is provided using hardware, software, and tools/equipment similar to those used on the job. Hands-on training is structured.

- Self-study - Self study is performed by the trainees, using formal materials or company/project documents as a guide.
- On-the-Job - On-the-Job training (also called mentoring) is performed under the cognizance of a qualified person. The qualified person maintains adequate oversight of the trainee to ensure the correct performance of the task.

Personnel mastery of the course materials is evaluated as required.

- If required by the curriculum, students are evaluated to determine mastery of the topics.
- Successful completion of hands-on tasks may be used to demonstrate mastery.
- Oral questioning techniques may be used to demonstrate mastery.

The method of evaluation is selected according to the following criteria:

- The difficulty of the task/job requirement
- The frequency of the task/job requirement
- The criticality of the task to nuclear safety

Training related records are submitted to the AREVA NP (Inc) Training supervisor and maintained electronically.

12.3 Training Resources

The SIVAT Tool and the associated SIVAT-TXS Simulation Based Validation Tool User Manual are the primary training resources.

12.4 Training Requirements

All Design Group personnel shall be qualified on the use of the SIVAT Tool prior to performing debugging of the Application Software with the tool.

All Independent Verification and Validation Group personnel shall be qualified on the use of the SIVAT Tool prior to performing validation testing of the Application Software with the tool.

13.0 SOFTWARE SAFETY PLAN

The SIVAT Tool utilizes the C Code generated by the SPACE Code Generators used to generate the code for the target system. The code is modified to run as a model in the SIVAT environment, but the code functionality is unaffected. The simulation process is described in more detail in Section 3.0. The C Code is compiled using two widely used compilers: one for the target processors (Intel iC 86) and one for the simulation environment (GNU Compiler Collection - GCC).

13.1 Effect of SIVAT on Target System Code

The SIVAT Tool does not produce code that is run on the TELEPERM XS safety processors. In fact, the code produced for simulation could not run on the safety processors due to differences in compilation and the alterations to memory mapping to support messaging simulation. The SIVAT Tool does not modify the Application Software code that is loaded on the TELEPERM XS safety processors. SIVAT has no effect on the Application Software and cannot create a safety hazard affecting safety functions.

13.2 Fidelity of SIVAT Simulation

SIVAT generates a separate CPU model in C Code for each processing module from the project-specific SPACE database. The Application Software code for FDs and FDGs is included in the respective CPU models. The CPU models also include a partial emulation of the TELEPERM XS Runtime Environment that satisfies the requirements of the simulation control system. The cycle time of the runtime environment is based on the design determined cycle time.

[]

The source codes of most FBs do not change for the simulator, as the code is designed with the requirement that the FBs can be used in the simulator. There are some exceptions to this rule for two hardware/software interfaces:

The interface of these FBs is the same in the simulator as in the actual I&C, but the behavior of these FBs in the simulator differs from the behavior in the actual TELEPERM XS I&C system hardware.

The limitations of SIVAT simulation are well known, as described in Sections 3.6. Other means of Application Software verification or validation are described in Section 11.2 to address these limitations. As such, the safety hazards not detected because of the limitations of SIVAT simulation can be identified by other verification or validation activities.

13.3 Transparency of SIVAT Code Generation

The C Code created by the SPACE Code Generator and the C Code modified for simulation can be readily compared using standard code difference identification tools. The differences can be checked at any time since the code files are archived in the software libraries. This capability allows internal auditors, customer representatives, external assessors, and regulatory authorities the capability to provide any degree of oversight. This capability also supports investigation of anomalous behavior observed during SIVAT use. This capability has been demonstrated to NRC on several occasions (meetings and audits related to TELEPERM XS projects). As such, suspected safety hazards can be readily investigated and corrected if substantiated.

14.0 SIVAT VERIFICATION AND VALIDATION PLAN

The SIVAT Verification and Validation Plan describes the methods used to ensure correctness of the SIVAT Tool software.

14.1 SIVAT Tool Verification and Validation Activities

The verification activities for the SIVAT software are performed in accordance with FAW-TXS-1.6. All software development lifecycle documents that are created for SIVAT are reviewed. The review process is conducted and documented according to FAW TXS-2.2 using review checklists. The verification process of the phase results is documented in the review protocol and the checklist for each phase of a software component.

All changes that are introduced to the SIVAT source codes are reviewed in accordance with the requirements of FAW TXS-1.5, as described in Section 15.0. The majority of the CRs address the controlling interface (e.g. changes to the graphical user interface or commands) of SIVAT. The simulated behavior of the TELEPERM XS Application Software depends directly on the SPACE Tool Code Generators, which are controlled in accordance with the lifecycle process described in the TELEPERM XS Topical Report.

All changes made to the SIVAT software are validated with an Integration and CR test performed by qualified personnel on dedicated test machines. The Integration and CR test is performed for each SIVAT release, as described in Sections 15.0 and 16.0. The testing includes tests for CRs that were introduced in the software release.

The validation results are stored together with the test report inside the document management system. The SIVAT source code is stored in the software configuration management system.

14.2 Initial SIVAT Tool Validation Activities

The I&C system at the Unterweser nuclear power plant was retrofitted with TELEPERM XS in 1996. At that time, all tests in the test field were still implemented with the real

TELEPERM XS system and a linked process model. The test cases were also verified with a UNISYS test arrangement and a Konvoi system simulator. UNISYS was the simulator control system that was used for TELEPERM XS simulations prior to SIVAT. The results of this simulation were compared with the test field results and concordance was verified.

No test field was available for upgrading the TELEPERM XS I&C system at the Unterweser plant in 2000, since the TELEPERM XS system was installed in the plant. Planned changes could only be tested through validation with SIVAT, which has been available as a TELEPERM XS V&V tool since 1998. For this purpose, first the test cases from the old simulation environment (UNISYS) and the test field were recalculated with SIVAT. Since the results matched, the verification of the modified I&C functionality was also implemented with SIVAT.

In addition, a closed-loop system test (load shedding from 71% reactor power down to house load) was recalculated by SIVAT and the process model (i.e., system model NLOOP Unterweser). The very high concordance between the actual system behavior and the simulation results lead to the authorization for installing the modified SIVAT-validated TELEPERM XS I&C. Authorization to install the modified TELEPERM XS application functions was given based on the very high concordance between the actual system behavior and the SIVAT validation. Plant commissioning took place without findings concerning the new I&C application functions.

A number of test field tests were verified with SIVAT as part of the TELEPERM XS retrofitting for the Philippsburg 1 nuclear power plant. The very high concordance made it possible to implement individual changes in the TELEPERM XS configuration even after the test field tests. These modifications were verified and validated exclusively with SIVAT.

14.3 Operating Experience with SIVAT

Since the first SIVAT version was introduced in 1999, this tool has been applied in all TELEPERM XS I&C projects. AREVA NP has operating experience with the use of SIVAT for more than 20 project specific applications. No instances have been reported where a system tested using SIVAT did not perform as expected after installation. The projects that have been verified and validated with SIVAT through 2005 are listed in Table 14-1.

Table 14-1 - TELEPERM XS Projects Verified and Validated with SIVAT

Plant/nuclear power plant	TELEPERM XS system
Unterweser (Germany)	Reactor control and limitation
Neckarwestheim 1 (Germany)	Reactor control and limitation
Bohunice V1 (Slovakia)	Reactor safety system
Bohunice V2 (Slovakia)	Reactor safety system
Philippsburg 1 (Germany)	Emergency system, local nuclear monitoring
Research reactor FRM2 (Germany)	Complete safety I&C
Beznau 1 and 2 (Switzerland)	Reactor safety system and control
Tianwan 1 and 2 (China)	Complete safety I&C
Research reactor AKR2 (Germany)	Complete safety I&C
Biblis B (Germany)	Reactor control and limitation
Biblis A and B (Germany)	Emergency supply steam generator (secondary)
Paks 1-4 (Hungary)	Reactor safety system
Forsmark (Sweden)	Rod control
Oskarsham 1-3 (Sweden)	Neutron flux
Atucha (Argentina)	Reactor safety (second heat sink)
Diverse systems (Germany)	I&C for turbine-generator set
Emsland (Germany)	Reactor control
Kozloduy (Bulgaria)	Diesel control, coolant pressure monitoring
Grohnde (Germany)	Power distribution monitoring

14.4 Independent Review of SIVAT

In 2006, the Institut für Sicherheitstechnologie (Institute for Safety Technology known as ISTec), issued an assessment report about the TELEPERM XS tools, which also includes a third party statement about the purpose and suitability of SIVAT. The discussions regarding SIVAT are reproduced below.

6 VALIDATION TOOL SIVAT

6.1 Concept of the SIVAT

The Simulation and Validation Tool (SIVAT) provides capabilities to test and validate the original SPACE generated code of I&C functions against the specification. The application code is compiled and operated on the simulator workstation with no need to access the target hardware. The consequences of particular hardware malfunction of I&C functions can be simulated and analyzed. After the installation of the target system in the plant, there is a software test environment available to evaluate the effects of later modifications on the system. The SIVAT tool provides also the possibility to connect a process model in a closed loop configuration. The SIVAT tool CATS-SDE (Code Adaptation Tool for Simulator SDE) controls the automatic generation of the SIVAT simulator and the simulator environment. The generated code (function diagrams and the Run Time Environment¹ (RTE) as C code) serves as the input of the simulation.

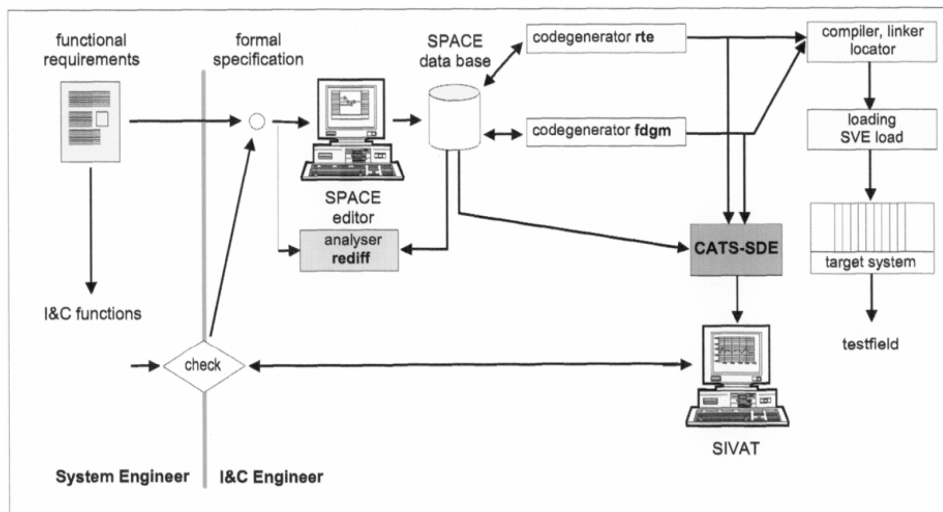


Fig. 3: Software validation using SIVAT

¹ That means the generated part of the RTE that is the interface of the RTE to the FD/FDG modules. Most of the RTE code is fix.

The main objectives of the SIVAT are to bring evidence that

- the correct FBs have been used,
- the FBs are correctly connected,
- the parameters of the blocks are correct,
- the I&C functions implement the specified behavior concerning the signal values as well as the signal status.

The availability of the SIVAT environment is independent of the availability of the target hardware. Therefore, SIVAT tests can be performed prior to test field tests, but also after the tests in the test field have been finished.

The validation of the I&C functionality of the application code is feasible in SIVAT because

- the functionality is contained in the function diagrams. From the function diagrams the C source code is generated automatically by the code generators of SPACE. This hardware independent C code is the source for the compiler for the target system of TXS and is also compiled and operated on the SIVAT environment. Therefore, the code of the application software used in the SIVAT environment is the original code used in the target system with few insignificant adaptations to the SIVAT simulator database. The project related application software is separated from the TXS platform system software by a clearly defined interface. The SIVAT environment provides this interface to the application software, too. The functional behaviour of the application software in the SIVAT environment and in the target system is the same (for insignificant differences see /3/, /4/).
- the system behaviour for assumed failures of components (I/O boards, CPUs, communication buses etc.) and systematic failures can be simulated
- safety set points and safety criteria can be tested
- signals and variables can be visualised
- partial functions and modules can be tested

The SIVAT tool has been successfully used in several validation procedures (see table 1).

Table 1: Major application of the SIVAT simulation environment

Project	Plant	Country
Control and limitation system	NPP Unterweser	Germany
RPS	NPP Bohunice V1	Slovakia
EKU emergency safety system	NPP Philippsburg 1	Germany
LKU-system for local core surveillance	NPP Philippsburg 1	Germany
Safety I&C	FRM-2	Germany
RPS, reactor control	NPP Beznau	Switzerland
RPS, reactor control and limitation system	NPP Tianwan	China

Especially the reports on the application of SIVAT in KKK (NPP Unterweser) and KKP-1 ECU (*/3/, /4/*) document the same functional results of the SIVAT tests and the corresponding tests in the test field. Differences have been discussed and justified.

The simulated run of I&C functions on the workstation is performed cyclically and synchronised, but not in real time. The real time aspect is not important, due to the fact that there is no time management in the system software. Time conditions are mapped to numbers of computation cycles. The sequence of the computation of the function diagrams in the simulation environment is only one special sequence of the computation of the function diagrams in the target system. Since the concept of the TELEPERM XS is based on an asynchronous behaviour of the different processing units, the simulated run of I&C functions will have no significant deviations from the functional behaviour of the target system.

7 SUMMARY

The TELEPERM XS platform and its environment comprise several software tools for development, documentation, analysis, verification and validation. Dependent on their role in the life cycle and the different safety significance of their outputs these tools are differently qualified.

The code generators that are specific for the TELEPERM XS platform are type tested following the same procedures applied to the on-line software, because their outputs are the C code of the application specific on-line software.

Compiler, linker, and locator are widespread used tool. Long time operational experience is available. Known bugs of these tools do not come to effect in TXS projects. During the type test of the TXS platform the usability of these tools were assessed by independent third party expert organizations with positive result.

The SPACE editor directly modifies the content of the specification database. From the data-base graphical representations of the project specification data are created. Thus, the results of the inputs can be verified. Additionally the independent documentation tool “fdprint” produces the paper documentation that can be used for verification. Until now, no functional failures have been detected.

The specification is stored in the standard data base system ORACLE that is used in a very large number of applications, also in safety relevant ones.

Documentation and analysis tools are used during project planning. They have no direct safety significance. Thus, these tools are designed and implemented based on the quality assurance procedures of the supplier. Third party assessment was not identified as necessary.

The verification tool “scanmic” is a simple tool to extract strings from MIC files and to calculate CRC check sums. It has no impact on the on-line software and no direct safety significance. It was designed and implemented using the internal quality assurance procedures of the supplier. Third party assessment was not identified as necessary.

The verification tool RETRANS was designed and implemented completely independent from the development of SPACE. This tool was validated by several different applications. It generates various text files that list potential inconsistencies. The decision about fault or planned deviation must be made by the assessor, a human being.

The validation tool SIVAT is suitable for validation of the functional behavior of the TXS application software. It was validated by several applications. The validation of SIVAT demonstrated the equivalence of the functional behaviour of the application software in the SIVAT environment and in the target system.

It should be noted that the SIVAT software is classified as Class D software, based on the requirements in FAW-TXS-1.1; consequently, no software type-testing, including third party assessment was required. SIVAT has been accepted for its mission by Technischer Überwachungs-Verein (German Technical Inspection Agency known as TÜV) in the course of the project-specific licensing for the German projects.

15.0 SIVAT CONFIGURATION MANAGEMENT PLAN

The SIVAT Configuration Management Plan describes the method that maintains the SIVAT software in a controlled configuration.

15.1 Use of TELEPERM XS Configuration Management Plan

SIVAT was originally developed during the years 1998-1999 to provide a simulation-based test environment to support the development of project-related TELEPERM XS Application Software.

The TELEPERM XS Topical Report describes the generic development and qualification process for the TELEPERM XS digital I&C system. SIVAT was developed under the same software configuration management plan described in the TELEPERM XS Topical Report. Specifically, SIVAT was developed in accordance with the AREVA NP GmbH Engineering Procedure FAW-TXS-1.5, Configuration Management Plan for the TELEPERM XS System Platform. NRC reviewed this procedure as part of the review of the TELEPERM XS Topical Report. NRC approved the TELEPERM XS Topical Report in a safety evaluation report issued in May 2000.

The TELEPERM XS configuration management process described in Engineering Procedure FAW-TXS-1.5 has evolved since the TELEPERM XS Topical Report was issued. A process change was made to add a change control board to the configuration management process. The process was enhanced to include additional detail describing configuration management tasks (e.g., introduction of a tracing sheet for each CR and more precise definition of the handling status of CRs). Specifically, there is an enhanced description of the platform configuration structure, unique component identifiers, version control, change control, release process, and documentation, with consideration of DIN EN ISO 10007 (Reference 11). This engineering procedure also addresses the requirements of the type tests for the TELEPERM XS system platform which covers the recommendations of relevant parts of DIN EN ISO 10007 and IEEE Std 828-1998.

SIVAT software, as is the case for all TELEPERM XS software, is managed using the Clearcase software configuration management system.

Changes to the SIVAT Tool are controlled via FAW-TXS-1.5, which establishes requirements to ensure that changes are controlled, documented, and tested. An overview of the TELEPERM XS software configuration management process is shown in Figure 15-1.

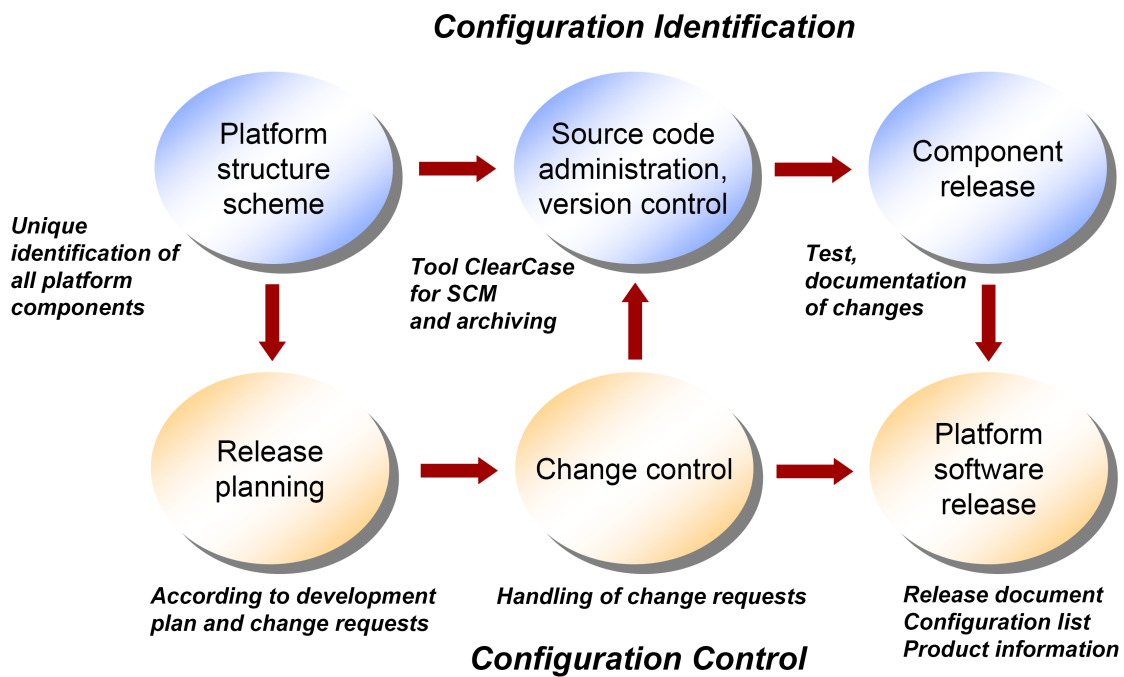


Figure 15-1 – Configuration Management Process Overview

15.2 SIVAT Life Cycle

The SIVAT Tool has been in use for many years. At this point, changes to the SIVAT Tool result from new user requirements, improvements based on operating experience feedback from the users, resolution of identified errors, and necessary adaptations due

to changes in related components. The life cycle of the SIVAT Tool is shown in Figure 15-2.

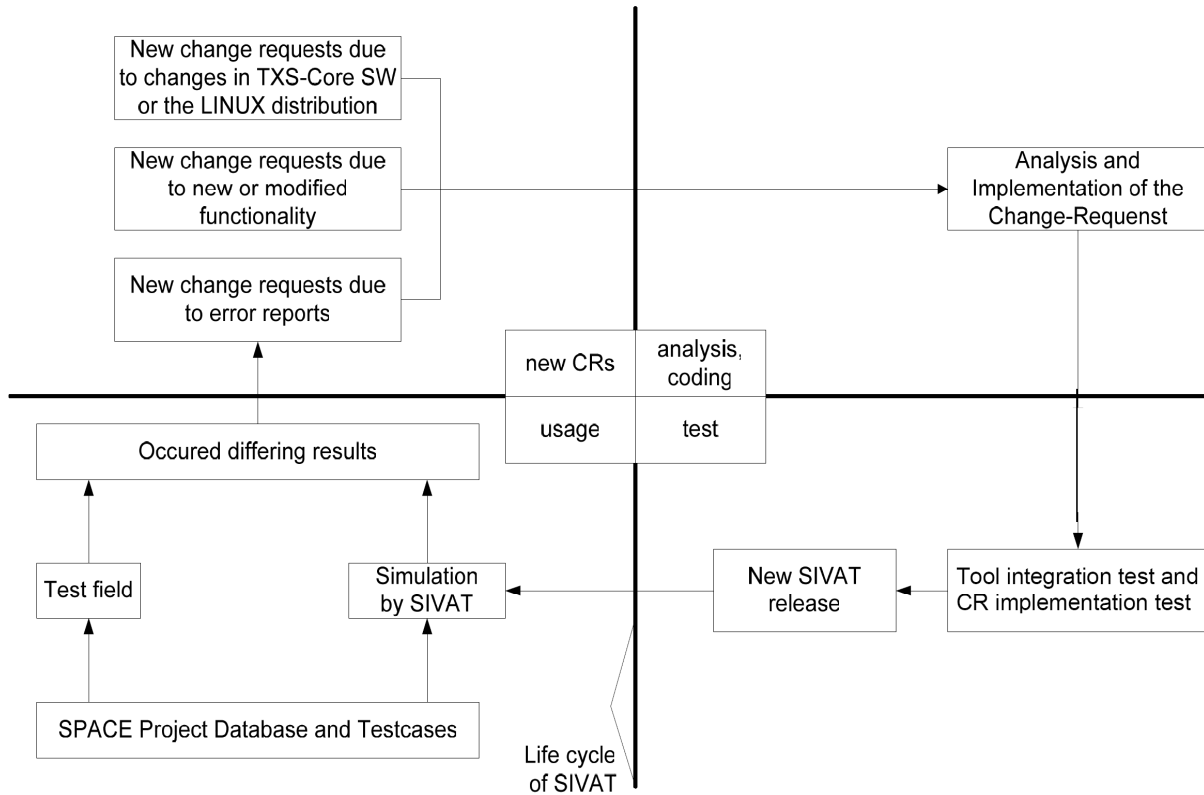


Figure 15-2 – SIVAT Life Cycle

CRs for the SIVAT Tool are analyzed and commissioned, if necessary, through the software configuration management process described above. The changes are implemented and the correct implementation of the changes is validated through testing (i.e., in the Integration and CR Implementation test). A new SIVAT version is released after successful completion of the specified testing.

15.3 Use of Software Program Manual Configuration Management Plan

AREVA NP intends to use of the SIVAT Tool to support validation testing of TELEPERM XS Application Software developed in accordance with the TELEPERM XS Software

Program Manual. As such, configuration management for SIVAT Tools used for U.S. TELEPERM XS projects will be controlled through the TELEPERM XS Application Software Configuration Management Plan described in Section 12 of the TELEPERM XS Software Program Manual.

The SIVAT software and associated documentation are classified as configuration items for the TELEPERM XS projects where they are used for Application Software testing. The versions of SIVAT Tool used on each release of the Application Software are controlled and recorded.

16.0 SIVAT TEST PLAN

SIVAT was developed in order to validate the Application Software functionality of I&C systems. After deployment, the results of a SIVAT simulation were compared with the results of the online system for certain project applications. It was shown that both systems show the same functional behavior. The same behavior of simulation and online system on functional level was shown for SIVAT release 1.2.4, as described in Section 14.0. The equivalence of the results must be shown for further releases of SIVAT. The SIVAT Test Plan outlines the methods to be used to test future releases of SIVAT.

16.1 Background Information

The SIVAT Tool was developed in order to replace existing simulator solutions such as UNISYS. It was intended that SIVAT simulates the functional behavior of TELEPERM XS I&C systems (called online systems below) on the level of I&C application functions. The simulation models are generated from the original TELEPERM XS Application Source Code which is obtained from the SPACE Code Generators. The code is slightly adapted in order to use the centrally managed memory of the simulator instead of the memory of the online systems. SIVAT Release 1.2.4 was tested against a set of input, output, and state data which were measured during factory acceptance tests of online systems in the test field. The online systems which provided the data were delivered to the NPPs Unterweser (KKU) and Emsland (KKE). As a result of the SIVAT tests, it can be stated that the release 1.2.4 shows the same functional behavior as the online system in the test field, as described in Section 14.0.

16.2 Scope of Testing

In general, the tests of the SIVAT Tool can be separated into a CR Implementation test component and a Tool Integration component.

16.2.1 Change Request Implementation Testing

CR Implementation testing is designed to validate that the modified SIVAT software meets the new or modified requirements established for the CR.

The CR Implementation test is designed by a different person than the developer who made the SIVAT software changes. The test developer is familiar with the functionality of SIVAT and the changes which were made. The test developer considers the effects of the CR and establishes the test cases, considering possible side effects of the change.

16.2.2 Tool Integration Testing

The Tool Integration test validates the following attributes:

- SIVAT can be installed on an engineering work station,
- All installed tools and configuration files are correctly installed,
- All tools can be called on an actual TELEPERM XS database, and
- A simulator that works can be generated in conjunction with the SPACE Tool Code Generators.

The test performer is familiar with the operation and use of SIVAT.

16.3 Test Documentation

Both, the CR Implementation and Tool Integration tests are documented in a single test specification. The test cases are established by the test developer. The test cases are documented in the test specification. The test specification is reviewed by another person (usually a developer of SIVAT). The review is documented in the review protocol.

The test cases are carried out on dedicated test machines, which have the same installation as the machines used for TELEPERM XS Application Software simulation testing. The test results are documented in a single test report.

All test findings are resolved or reconciled prior to the release of the modified SIVAT software.

17.0 CONCLUSIONS

This Topical Report described the Simulation Validation Test Tool (called SIVAT) developed by AREVA NP to support the development of project-related TELEPERM XS Application Software. This report described:

- The concept of TELEPERM XS simulation and the principle of operation of the SIVAT and
- The high quality development process used to develop SIVAT.

This topical report described the concept of the TELEPERM XS simulation and the principle of operation of the SIVAT. This report showed that the I&C functionality represented in the Application Software can be effectively validated with the SIVAT Tool. The use of a NRC-approved simulation validation tools has been described in AREVA NP document ANP-10272, Revision 1, Software Program Manual for TELEPERM XS™ Safety Systems Topical Report (Reference 29), which is referred to as the TELEPERM XS Software Program Manual.

The use of the SIVAT Tool to support TELEPERM XS Application Software verification and validation has important benefits. The early detection of Application Software faults through validation testing with SIVAT serves to reduce project risks earlier in the development process.

AREVA NP requests that the NRC issue a Safety Evaluation Report that approves ANP-10303NP, Revision 0, SIVAT: TELEPERM XS™ Simulation Validation Test Tool Topical Report. AREVA NP intends to use the SIVAT Tool to support validation testing of TELEPERM XS Application Software developed in accordance with the TELEPERM XS Software Program Manual.

18.0 REFERENCES

18.1 U.S. Regulations

1. 10 CFR Part 21, "Reporting of Defects and Noncompliance."
2. 10 CFR Part 50, Appendix B, "Quality Assurance Criteria for Nuclear Power Plants and Fuel Reprocessing Plants."

18.2 U.S. Regulatory Guidance

3. Regulatory Guide 1.152, Revision 2, January 2006, "Criteria for Digital Computers in Safety Systems of Nuclear Power Plants."
4. Regulatory Guide 1.168, Revision 1, February 2004, "Verification, Validation, Reviews, and Audits for Digital Computer Software Used in Safety Systems of Nuclear Power Plants."
5. Regulatory Guide 1.169, September 1997, "Configuration Management Plans for Digital Computer Software Used in Safety Systems of Nuclear Power Plants."
6. Regulatory Guide 1.170, September 1997, "Software Test Documentation for Digital Computer Software Used in Safety Systems of Nuclear Power Plants."
7. Regulatory Guide 1.171, September 1997, "Software Unit Testing for Digital Computer Software Used in Safety Systems of Nuclear Power Plants."
8. Regulatory Guide 1.172, September 1997, "Software Requirements Specifications for Digital Computer Software Used In Safety Systems of Nuclear Power Plants."
9. Regulatory Guide 1.173, September 1997, "Developing Software Life Cycle Processes for Digital Computer Software Used in Safety Systems of Nuclear Power Plants."
10. NUREG-0800, Standard Review Plan, Chapter 7, Branch Technical Position (BTP) 7-14, "Guidance on Software Review for Digital Computer-Based Instrumentation and Control Systems."

18.3 International Standards

11. DIN EN ISO 10007, "Quality management, guidelines for configuration management"
12. IEC 60880, Nuclear Power Plant Instrumentation and Control Systems Important to Safety: Software Aspects for Computer-Based Systems Performing Category A Functions, May 2006

13. IEC 61226, "Nuclear Power Plant Instrumentation and Control Systems Important to Safety: Classification of Instrumentation and Control Functions," 2005

18.4 U.S. Industry Standards

14. IEEE Std 7-4.3.2-2003, "Standard Criteria for Digital Computers in Safety Systems of Nuclear Power Generating Stations."
15. IEEE Std 610.12-1990, "Software Engineering Terminology."
16. IEEE Std 828-1990, "Standard for Software Configuration Management Plans."
17. IEEE Std 829-1983, "Standard for Software Test Documentation."
18. IEEE Std 830-1993, "Recommended Practice for Software Requirements Specifications."
19. IEEE Std 1008-1987, "IEEE Standard for Software Unit Testing."
20. IEEE Std 1012-1998, "Standard for Software Verification and Validation."
21. IEEE Std 1028-1997, "Standard for Software Reviews."
22. IEEE Std 1042-1987, "Guide to Software Configuration Management."
23. IEEE Std 1074-1995, "IEEE Standard for Developing Software Life Cycle Processes."
24. IEEE Std 1228-1994, "IEEE Standard for Software Safety Plans"

18.5 Regulatory Review Precedents

25. NRC Safety Evaluation Report for Siemens Topical Report EMF-2110(NP), Revision 1, "TELEPERM XS: A Digital Reactor Protection System," May 5, 2000.
26. Siemens letter from James F. Mallay to NRC dated September 1, 1999, Supporting Documentation for Review of EMF-2110(NP) Revision 1, "TELEPERM XS: A Digital Reactor Protection System," NRC:99:037.
27. Siemens letter from James F. Mallay to NRC dated December 16, 1999, EPRI and QA Documentation Supporting Review of EMF-21 10(NP) Revision 1, "TELEPERM XS: A Digital Reactor Protection System," NRC:99:052.

18.6 AREVA NP Documents

28. Siemens Topical Report EMF-2110, Revision 1, "TELEPERM XS: A Digital Reactor Protection System," September 1, 1999.

-
29. ANP-10272, Revision 1, Software Program Manual for TELEPERM XS™ Safety Systems Topical Report
 30. AREVA NP Document No. 56-5015885, "Quality Management Manual."
 31. AREVA NP GmbH Engineering Procedure FAW-TXS-1.1, Phase Model for the Development of Software Components for TELEPERM XS
 32. AREVA NP GmbH Engineering Procedure FAW-TXS-1.5, Configuration Management Plan for the TELEPERM XS System Platform
 33. AREVA NP GmbH Engineering Procedure FAW-TXS-1.6, Software Verification and Validation Plan
 34. AREVA NP Administrative Procedure 1717-06, "Corrective Action Program (WebCAP)."