

**Abstract:**

This CWD gathers the set of contributions in the FAMILIES project that have been developed under the scope of the task 3.2 Execution Qualities. Topics covered in this CWD are related with methodologies, techniques and reference models for dealing with run-time system quality attributes when developing a System Family architecture. The contributions included in this CWD have been grouped in two main sections. Section I contributions are focused on topics related with security aspects of System Families, while Section II has been oriented to other system quality attributes such as performance, reliability and availability.

Keywords:

Run-Time, Architecture, Quality Attributes, Security, Performance, Reliability, Availability, Reference Model, Methodology

List of Authors:

CWD Editor: Miguel A. Oltra (Telvent)

Section I Editor: Tor Erlend Fægri (ICT-Norway)


- Chapter 1: Juha Savolainen (Nokia)
- Chapter 2: José Luis Arciniegas, José Luis Ruíz, Juan Carlos Dueñas, Rodrigo Cerón (UPM)
- Chapter 3: Tor Erlend Fægri, Svein Hallsteinsen, Ivar Sandstad, Jens Glattetre (ICT-Norway)
- Chapter 4: Miguel A. Oltra (Telvent)
- Chapter 5: Chris Broerse (Philips)

Section II Editor: Anne Immonen (VTT)

- Chapter 6: Laurent Rioux (Thales), Sebastien Gerard (CEA), Hubert Dubois (CEA)
- Chapter 7: Marcel Weijenborg (Philips)
- Chapter 8: Anne Immonen (VTT)



- Editor: Miguel Ángel Oltra
- Partner: Telvent
- Date: 24 July 2005
- Number: TLVT-CWD-Task32
- Version: 1.2
- Status: Final
- Level: Consortium Wide
- Contributors: CEA, ICT-Norway , Nokia, Philips, Telvent, Thales, UPM, VTT
- Work Package: 3. Family Quality
- WP Leader: Jesús Bermejo (Telvent) Accepted:



Contents

- Introduction
- Partners involved
- Objectives of the Task 3.2
- Task 3.2 & the FAMILIES Reference Framework
- Family Relevance of Task 3.2
- Sections Overview
- Task 3.2 & Previous Projects


TELVENT FAMILIES Task 3.2 CWD
Execution Qualities
© Telvent; Miguel A. Oltra

17/10/2005
Eureka Σ! 2023 Programme, ITEA project ip02009, FAMILIES

3


In this slide is indicated the table of contents that are included in the FAMILIES Task 3.2 Execution Qualities. Firstly an introduction about the problems being tackled within this task is provided. Secondly, the list of partners per country involved in this task are presented. The third part of the CWD will consist in a description of the objectives covered by this task, by indicating what is inside and outside the boundaries of the task in terms of topics that are covered within it. A mapping of all the contributions included in this task against the FAMILIES Reference Framework is of interest in order to obtain a high overview of the topics covered in partners contributions centred in the activities being covered both in domain and application engineer scopes.

The next topic of the CWD is an indication of the relevance of Task 3.2 within the FAMILIES project work. Following an overview of the sections in which is divided this CWD is provided and the relation of this task with previous projects (ESAPS and CAFÉ) is indicated. Finally a chapter overview is outlined for each partner contribution as a presentation of the main topics covered in each partner contribution, that will be presented in more detail in each section in which it is divided this CWD.

|  | |
|------------------------------------------------------------------------------------------------|----------------------------------------------------------|
| Contents (Cont.) | |
| • Chapters | |
| • Section 1, Security | |
| • Section 1, Introduction | |
| • Chapter 1. Quality determination and documentation techniques (Nokia) | |
| • Chapter 2. System family security (UPM) | |
| • Chapter 3. Security Reference Architecture (ICT-Norway) | |
| • Chapter 4. Security issues in dynamically deployable SFs (for distributed systems) (Telvent) | |
| • Chapter 5. Improving security quality (Philips) | |
| • Section 1, Conclusion | |
| • Section 2, Other Run-time QAs | |
| • Section 2, Introduction | |
| • Chapter 6. Quality of Service for Real-time and Embedded Systems (Thales & CEA) | |
| • Chapter 7. Resource usage (Philips) | |
| • Chapter 8. Predicting Reliability and Availability at the Architectural Level (VTT) | |
| • Section 2, Conclusion | |
| • Task 3.2 Conclusions | |
| • Task 3.2 Relation to FEF | |
| • The Authors Picture Gallery | |
| TELVENT | FAMILIES Task 3.2 CWD |
| Execution Qualities | 17/10/2005 |
| © Telvent; Miguel A. Oltra | Eureka Σ! 2023 Programme, ITEA project ip02009, FAMILIES |
| 4 | |

Chapters are organised in two sections, one that compiles partners contributions related to security quality attributes, and the other one that compiles other run-time quality attributes. Each section is divided in three parts: Introduction, chapters and conclusion. Finally the CWD presents the main conclusions achieved by the work done by each partner within the scope of the FAMILIES project, and also a relation of each chapter with the FAMILIES Evaluation Framework (FEF). To conclude an authors picture gallery shows the names and contact addresses for each contributor in this task.

Several of the chapters included in this deliverable are part of the future research book as a result of partners research. Results from chapters 2 and 4 will be included in the research book with the title: "Architecture reasoning in support of system family evolution; an example on security"; results from chapter 3 will be included in the research book with the title: "A reference architecture for security in system families"; and results from chapter 8 will be included in the research book with the title: "A method for predicting reliability and availability at architectural level".



Introduction

- **What can be considered within the scope of execution qualities?**
 - System family qualities, gathered in the scope of execution qualities category are addressed in task 3.2
 - Availability, reliability, safety, security, dependability among a long set of -ilities are system quality attributes that must be satisfied
- **Execution qualities have an impact on the architectural design of a system family**
 - How to build/design secure, reliable and dependable systems as quickly as possible?
 - How to assure a high level of availability, reliability and safety in critical systems?
 - How to model quality attributes when designing systems?
- **These non-functional aspects are present in a whole system family**
 - Techniques and methods that guarantee the quality of a system in an organisational context are proposed
 - Reference models and frameworks for dealing with some of these -ilities are also proposed

TELVENT FAMILIES Task 3.2 CWD
 Execution Qualities
 © Telvent; Miguel A. Oltra

5 Eureka Σ! 2023 Programme, ITEA project ip02009, FAMILIES

17/10/2005


Under the umbrella of Execution Qualities there are a set of aspects that must be taken into account. The SEI (Software Engineering Institute) defines a taxonomy of quality measures that a developed system may satisfy. They define a tree category based on five quality measures (Need satisfaction, performance, maintenance, adaptive and organisational). From this high level categorisation, only those under the performance categorisation are considered in the boundary of this task 3.2 (availability, reliability, safety, security, dependability, ..., among others).

Partners contributions in this task are addressing problems dealing with execution qualities oriented towards a system family approach. Those quality attributes must be guaranteed when designing a system family. The main problem appears due to the fact that these run-time system aspects must be taken into account during the initial phases of the system development, but can not be validated/verified until the final steps of the system implementation. As can be foreseen execution qualities have an impact on the architectural design of a system family. A set of questions dealing with the architectural design of system families in terms of the development of system non-functional aspects are tried to be answered within the different partners contributions of this task.

Among question that are addressed in the scope of this task are:

- How to build/design secure, reliable and dependable systems as quickly as possible?
- How to assure a high level of availability, reliability and safety in critical systems?
- How to model quality attributes when designing systems?

From a system families approach the non-functional aspects are present in the products developed in a family. Due to this reason, a set of techniques and methods are proposed in some of the contributions included in this CWD to guarantee the quality of the system architecture. Moreover, in other contributions are proposed reference models and frameworks that deal with some of the set of -ilities under the umbrella of execution qualities.



Introduction (Cont.)

- The results of partners' contributions have been grouped in two sections that represent the main topics covered by each partner contribution
- Security section, where several security aspects of a system family are covered in different approaches
- Other run-time quality attributes, where reliability, availability and performance are the quality attributes being covered
- Contributions are mainly centred on the proposal of techniques, methods and models for dealing with these non-functional aspects at architectural level

TELVENT FAMILIES Task 3.2 CWD
Execution Qualities
© Telvent; Miguel A. Oltra

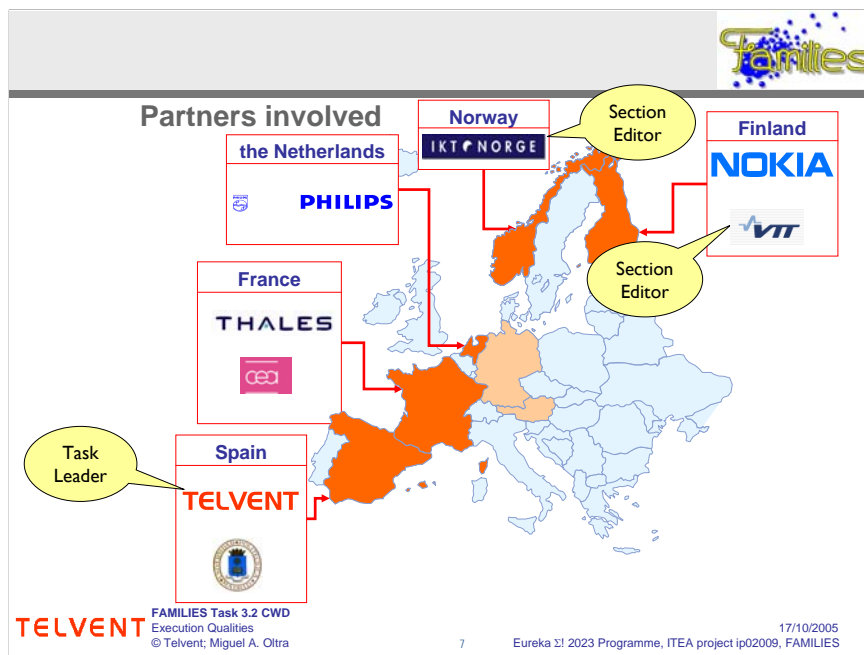
6 Eureka Σ! 2023 Programme, ITEA project ip02009, FAMILIES 17/10/2005

Following an overview of the organisation of this CWD is given. Partners' results have been grouped in two main sections, representing both the main topics covered by each partner contribution.

One of the section is called Security Section, where several security aspects of a system family are covered in a different approach for each partner.


The other section, has been called Other run-time Quality Attributes, where reliability, availability and performance are the set of quality attributes covered by partners contributions.

As a resume, the contributions in this CWD are mainly centred on the proposal of techniques, methods and models for dealing with the previously mentioned non-functional aspects at architectural level design.



This slide indicates the roles of the partner involved in the task 3.2. Partners from five European countries are present: Nokia and VTT (Finland), Thales and CEA (France), Philips (The Netherlands), ICT-Norway (Norway) and Telvent and UPM (Spain). The main roles of this task are task leader and section editors:

- Task leader: Telvent
- Security section editor: ICT-Norway
- Other run-time Quality Attributes: VTT



Objectives of Task 3.2

- **Purpose**
 - Covers System Family (SF) execution (run-time) quality attributes
 - Dependability attributes (availability, reliability and safety) for critical systems
 - Resources management/usage (performance)
 - Security in terms of connectivity requirements
 - Usability due to increasing features and complexity of systems
 - Develop quality models for security and deployment of services in a SF context (implications, solutions, design guidelines, ...)
 - Study of architecture level patterns and their influence on security quality attributes (outsourced or COTS)

TELVENT FAMILIES Task 3.2 CWD
Execution Qualities
© Telvent; Miguel A. Oltra


17/10/2005
Eureka Σ! 2023 Programme, ITEA project ip02009, FAMILIES

The original aim of the FAMILIES task 3.2 is expressed in the FAMILIES Full Project Proposal (FPP), where it is indicated that this task covers SF execution quality attributes. Among the set of attributes indicated in the FPP are:

- Dependability attributes such as availability, reliability and safety (accordingly with the SEI proposed taxonomy) for critical systems.
- Performance system aspects such as resources management and resources usage.
- Security, indicated in terms of connectivity requirements that must satisfy a system when it is interconnected with another ones.
- Usability aspects of the system design, taking into account the increasing number of features and the complexity of the final systems.

Moreover within the scope of the task it is the development of quality models related with security and deployment aspects of services in a SF context, where an analysis of implications, solutions and design guidelines, ..., should be proposed.

Finally, the study of architecture level patterns and their influence on security quality attributes related with outsourced components or COTS of a system are also taken into account.



Objectives of Task 3.2 (cont.)

- **Boundary**
 - Inside task 3.2
 - Run-time system quality attributes (performance, scalability, dependability, usability, security, ...)
 - Architecture models (architecture definition, component based distributed systems, ...)
 - Business system quality attributes integration (3rd party assets or COTS)
 - Outside task 3.2
 - Development time system quality attributes (modifiability, adaptability, portability, reusability, ...)
 - Business quality attributes (time-to-market,...)
 - Non run-time qualities related to the architecture (conceptual integrity, buildability, correctness, completeness, ...)

TELVENT FAMILIES Task 3.2 CWD
Execution Qualities
© Telvent; Miguel A. Oltra

17/10/2005
Eureka Σ! 2023 Programme, ITEA project ip02009, FAMILIES


In this slide is shown what is inside the boundary of the task, and what is outside the boundary of it when dealing with execution qualities.

Inside the issues covered in the task are:

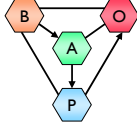
- Run-time system quality attributes such as performance, scalability, dependability, usability, security, ...
- Proposals of architectural models oriented toward different aspects of the system architecture design: architecture definition, component based distributed systems, ...
- Business system quality attributes integration concerning to 3rd party assets or COTS

Outside the boundary of the task are:

- Development time system quality attributes such as modifiability, adaptability, portability, reusability, ...
- Quality attributes related with business implications: time-to-market, ...
- Non run-time qualities related to the architectural design: conceptual integrity, buildability, correctness, completeness, ...



Family Relevance of Task 3.2



- **Task 3.2 Relation to BAPO**
 - **Business**
 - Quality attributes have a closed relationship with Business needs
 - Quality aspects must be assured for COTS and third party assets (impact on global quality of system architecture)
 - **Architecture**
 - Mainly focused on architectural reference models and architecture level patterns
 - Quality attributes covered by partners: security, fault tolerance, performance, availability and reliability
 - **Process**
 - Not foreseen
 - **Organisation**
 - Not foreseen


FAMILIES Task 3.2 CWD
 Execution Qualities
 © Telvent; Miguel A. Oltra

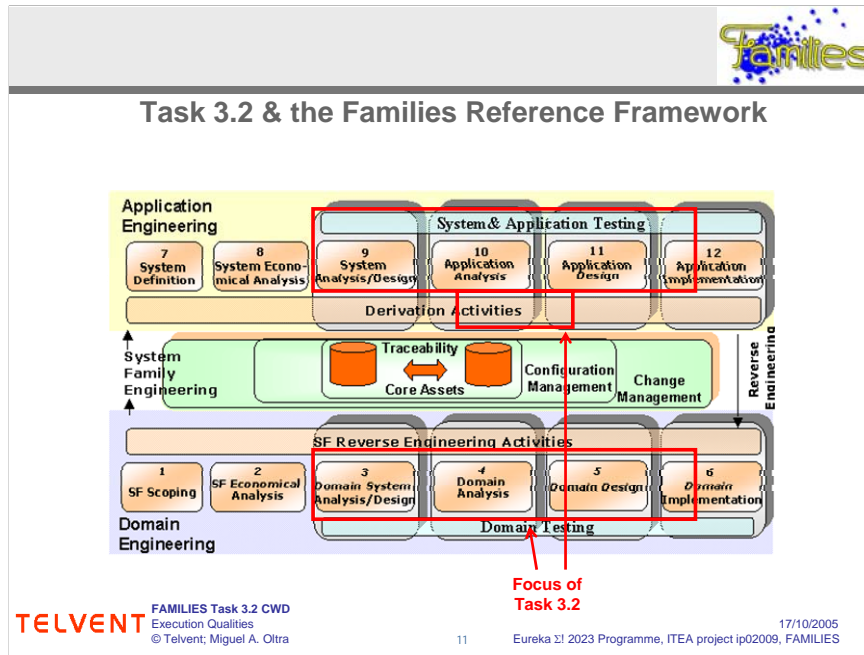
10

17/10/2005
 Eureka Σ! 2023 Programme, ITEA project ip02009, FAMILIES

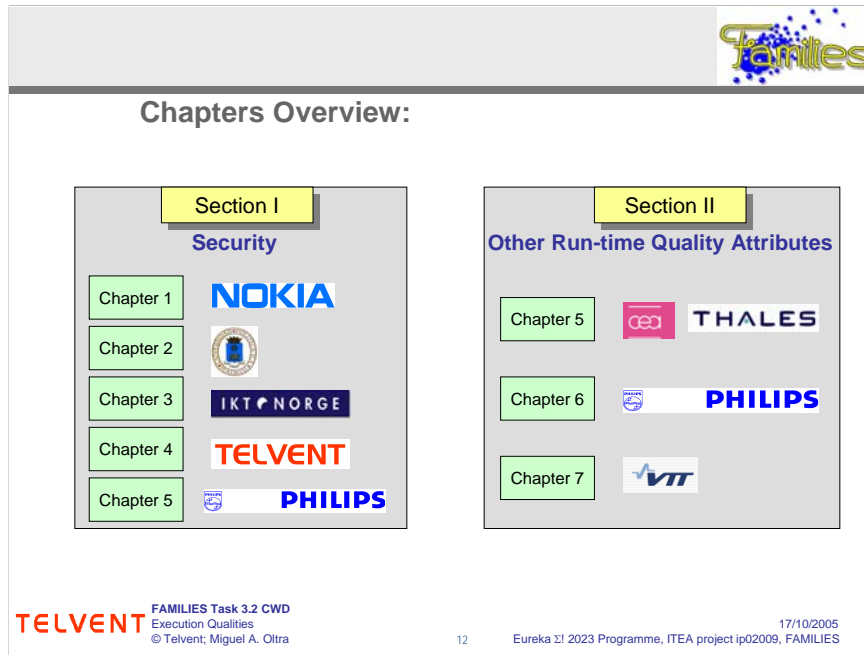
The relevance of task 3.2 with regards to the FAMILIES project, is illustrated in this slide, where the relation with the BAPO model is indicated. The BAPO model represents four dimensions of companies that develop software. The four dimensions are: Business, Architecture, Process and Organisation. With regards to the dimensions task 3.2 contributions have a relation with both Business and Architecture dimensions; mainly centred in the Architecture dimension of the model, but with some connotations related to the Business dimensions.

The task 3.2 relation with the Business dimension is due to the fact that quality aspects must be assured or satisfied for third party assets and COTS included in the system architecture, due to the impact that they may represent to the global quality of the system architecture.

With regards to the architectural dimension, task 3.2 partners contributions are mainly focused on architectural reference models and architecture level patterns. Among the different quality attributes covered in the several contributions are: security, fault tolerance, performance, availability and reliability.




In this slide is resumed the mapping of all the contributions of task 3.2 against the Families Reference Framework. Following for each chapter will be expressed in more detail where is exactly focused each of the contributions. As it is reflected in the figure the content of the CWD is mainly focused both in Domain and Application Engineering activities related to: System Analysis/Design, Analysis and Design. Some work has been done in the activities related with Implementation. Moreover, one chapter is focused on derivation activities as indicated in the figure.



All partner contributions have been grouped into two main sections, depending on the focused problems that are dealing with each contribution.

On one hand, Section I deals with problems related to security aspects. Different approaches are presented in the five chapters of this section, covering each one different security aspects of a System Family. Reference architectures, documentation techniques, secure life cycle development among other topics are covered in the chapters presented within this section.

On the other hand, Section II has been denoted with the title Other Run-time Quality Attributes due to the fact that in the three chapters of this section, quality attributes are being covered such as performance, availability, reliability when dealing with the design of System Families.



Section I Overview: Security

- **Section editor: ICT-Norway**
 - Tor Erlend Fægri (tor.e.fegri@sintef.no)
- **List of contributions**
 - Chapter 1. Quality determination and documentation techniques (Nokia)
 - Chapter 2. System family security (UPM)
 - Chapter 3. Security Reference Architecture (ICT-Norway)
 - Chapter 4. Security issues in dynamically deployable SFs (for distributed systems) (Telvent)
 - Chapter 5. Improving security quality (Philips)
- **List of participants**
 - Nokia (Juha Savolainen)
 - UPM (José Luis Arciniegas, José Luis Ruíz, Juan Carlos Dueñas, Rodrigo Cerón)
 - ICT-Norway (Tor Erlend Fægri , Svein Hallsteinsen, Ivar Sandstad, Jens Glattetre)
 - Telvent (Miguel A. Oltra)
 - Philips (Chris Broerse)


TELVENT FAMILIES Task 3.2 CWD
Execution Qualities
© Telvent; Miguel A. Oltra, © ICT-Norway; Tor E. Faegri 13

17/10/2005
Eureka Σ! 2023 Programme, ITEA project ip02009, FAMILIES

The main objective of the work presented in this section is to introduce new knowledge about how security, a cross-cutting concern affecting a wide range of aspects of software intensive systems, can be dealt with in the context of product family engineering.

Security is a topic that has received significant attention lately. Dealing with security requirements is not a new topic in software engineering. But extensive use and increasing dependence on networked information systems has been the target of a wide range of threats that has caused the whole IT industry to reapply focus upon how to deal with the challenge.

Drawing benefits from a large body of existing research within the field of product family engineering, these contributions seek to push the knowledge barriers further. They show that in order to address security as a specific quality, we require new principles, techniques, tools and experiences. Many such values are presented. These contributions contain innovative work that will assist in the handling of security in the coming years of product family engineering.



Section I Overview: Security (Cont.)


- **Problems and objectives of the chapters**
 - To provide the architectural models needed to describe, measure and analyse quality properties. This will allow the answering of questions such as:
 - How to define security qualities towards the architecture
 - How to document these non-functional requirements
 - What kind of metrics may be associated with these non-functional requirements
 - To provide a security system family life cycle model
 - By covering the whole system development process through a secure development
 - By providing an architectural design based on security standards and required technologies for its implementation
 - By defining scenarios and demonstrators for validating specific security aspects of the architectural design

TELVENT FAMILIES Task 3.2 CWD
Execution Qualities
© Telvent; Miguel A. Oltra, © ICT-Norway; Tor E. Faegri 14

17/10/2005
Eureka Σ! 2023 Programme, ITEA project ip02009, FAMILIES

Among the problems and objectives being covered in Section I in the five chapters included in it are:

- The provision of architectural models needed to describe, measure and analyse quality properties during system design phase. This will allow the answering of questions such as: How to define security qualities towards the architecture?, How to document these non-functional requirements?, What kind of metrics may be associated with these non-functional requirements?, ...
- The provision of a security system family life cycle model that covers the whole system development process through a secure development. This family life cycle will provide an architectural system design based on security standards and with the identification of the required technologies for its implementation. Scenarios and demonstrators will be defined for validating specific security aspects of the architectural design.



Section I Overview: Security (Cont.)

- **Problems and objectives of the chapters (cont.)**
 - To provide a decision support framework assisting the system family architect in designing and evaluating software architectures that are confronted with security quality requirements
 - By encoding architectural design rules in a flexible structure that can be easily maintained
 - By considering architectural principles as an extremely valuable source of knowledge
 - By respecting the need of organisations to adopt and refine the framework to their own environment
 - To provide a security reference model based on standards for distributed heterogeneous systems
 - By providing an architectural design based on security standards and required technologies for its implementation
 - By providing security design guidelines on System Families for component based systems through the identification of system required countermeasures
 - By identifying and defining a validation scenario covering several security quality attributes (accounting, availability, confidentiality, integrity) of the reference model


TELVENT FAMILIES Task 3.2 CWD
Execution Qualities
© Telvent; Miguel A. Oltra, © ICT-Norway; Tor E. Faegri 15

17/10/2005
Eureka Σ! 2023 Programme, ITEA project ip02009, FAMILIES

More problems and objectives being covered with regards to security aspects in chapters 3 and 4 are:

- The provision of a decision support framework assisting the system family architect in designing and evaluating software architectures that are confronted with security quality requirements. This framework will encode architectural design rules in a flexible structure that can be easily maintained and will consider architectural principles as an extremely valuable source of knowledge. Another particularity or characteristic of this framework is that it will respect the need of organisations to adopt and refine the framework to their own environment.

- The provision of a security reference model based on standards for distributed heterogeneous systems, by means of an architectural design based on security standards and required technologies for its implementation. This will be accomplished with the provision of a set of security design guidelines on System Families for component based systems through the identification of system required countermeasures. Moreover, a identification and definition of a validation scenario covering several security quality attributes (accounting, availability, confidentiality, integrity) of the reference model is mandatory for refining the reference model.



Section I Overview: Security (Cont.)


- **Problems and objectives of the chapters (cont.)**
 - To describe a viable approach to the effective responding to incident reports in a product family engineering organisation
 - Where different concerns arise both at the divisional level and the business unit level
 - Where the frequency of reports is high and requires rapid and effective response from the groups
 - In order to ensure the lowest possible exposure to security threats in medical applications

TELVENT FAMILIES Task 3.2 CWD
Execution Qualities
© Telvent; Miguel A. Oltra, © ICT-Norway; Tor E. Faegri 16

17/10/2005
Eureka Σ! 2023 Programme, ITEA project ip02009, FAMILIES

Finally in chapter 5 the following problems and objectives will be covered:

- The description of a viable approach in order to obtain an effective responding to incident reports in a product family engineering organisation. In this problem, different concerns arise both at the divisional level and the business unit level in an organisation. Also has to be taken into account that the frequency of reports is high and requires rapid and effective response from the different groups; in order to ensure the lowest possible exposure to security threats in medical applications developed by the organisation.



Section I Overview: Security (Cont.)

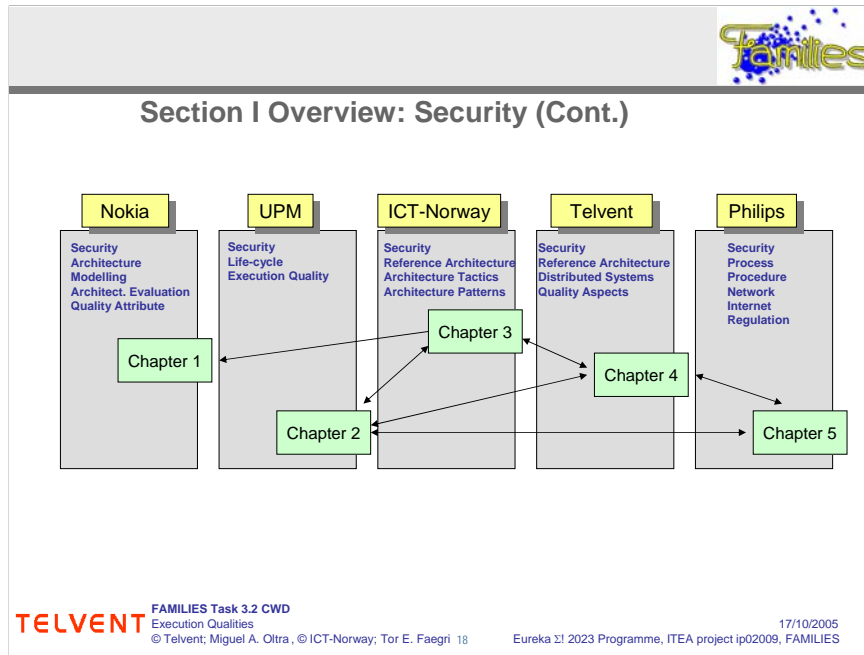
- **Overview of results**
 - (Nokia) Documentation techniques to describe, measure and analyse security quality aspects by means of architectural modelling
 - (UPM) Security system family life cycle based on well known security standards (Common Criteria, OMG Security Specification, CIM-DMTF user profile)
 - (ICT-Norway) A decision support framework forming a reference architecture for security in system families. The framework provides support to the design and evaluation of product architectures. It does this through capture and refinement of architecture design knowledge useful for the developing organisation
 - (Telvent) Target platform independent security reference model based on standards and covering security countermeasures to guarantee security architectural quality aspects in a component based architecture
 - (Philips) A process for response improvement to incident reports (related to security aspects) that may occur in a product family engineering organisation

TELVENT FAMILIES Task 3.2 CWD
Execution Qualities
© Telvent; Miguel A. Oltra, © ICT-Norway; Tor E. Faegri 17

17/10/2005
Eureka Σ! 2023 Programme, ITEA project ip02009, FAMILIES

This slide summarises the different results achieved in each chapter:

- Chapter 1: Results from this chapter are a set of documentation techniques to describe, measure and analyse security quality aspects by means of architectural modelling
- Chapter 2: Among the main result of this chapter is the proposed security system family life cycle based on well known security standards (Common Criteria, OMG Security Specification, CIM-DMTF user profile)
- Chapter 3: A decision support framework forming a reference architecture for security in system families is the main result of chapter 3. This framework provides support to the design and evaluation of product architectures, by means of a capture and refinement of architecture design knowledge useful for the developing organisation.
- Chapter 4: As a consequence of the work done in chapter 4, the main result is a target platform independent security reference model based on standards. This reference model covers security countermeasures that must be guaranteed to assure the security architectural quality aspects in a component based architecture.
- Chapter 5: The main result of chapter 5 is the definition of a process for response improvement to incident reports (related to security aspects) that may occur within a product family engineering organisation



Commonalities/differences/interrelations of Contributions:

Chapter 1 (Quality determination and documentation techniques): In this Chapter, security architectures are modelled by employing a combination of security specific models and generic model constructs. Aspects related with system security are shown by defining specific types that constrain how the models can be created. During the design phase a refinement of models has to be done. The intention in this Chapter is to use the classification of mechanisms presented in Chapter 3 in the documentation mechanisms proposed on it.

Chapter 2 (System family security): the content of this chapter includes a model and traceability results related to a secure development by means of a security system family life cycle based on standards (analysis, design, implementation, testing and also operation and maintenance are treated taking into account security aspects). The work in this chapter has been done in cooperation with the work done in Chapter 4 where the life cycle is used in the definition of a security reference model for distributed systems. Also results from Chapter 3 may be taken into account in order to refine the life cycle phases. Results from Chapter 2 may be also useful for results in Chapter 5, where process and procedures for dealing with security aspects in interconnected systems through Internet are proposed.


Chapter 3 (Security reference architecture): in this Chapter, a reference architecture that supports the capture, use and maintenance of knowledge related to designing system family architectures that are faced with security requirements is presented. A quality model for specifying security requirements is presented, a set of architectural solutions based on tactics and patterns for addressing security requirements are indicated in the contribution and moreover a decision model in order to meet architectural requirements are encoded in architectural knowledge, by supporting the selection of architectural approaches that fits the requirements. Results in this Chapter are valid for the rest of Chapters, due to the fact, that guidelines for dealing with security aspects at architectural level may be useful for the rest of the partners. A collaboration has been settled between ICT-Norway, Telvent and UPM, due to the fact that commons scenarios have been addressed in their contributions.

Chapter 4 (Security issues in dynamically deployable SFs (for distributed systems)): in this Chapter, a reference model based on standards addressing security aspects in distributed systems is covered. The proposed security concerns indicated in this Chapter have been gathered in a way that can be useful independently of the target platform. Moreover an scenario for model validation has also been elaborated in this Chapter. Results from this Chapter 4 may be useful to validate both the model and security guidelines presented in Chapter 3. Also some similarity appears with the target system environment in the Chapter 5; results from both contributions may be enriched by each other.

Chapter 5 (Improving security quality): Chapter 5 is coping with the improvement of security requirements, embedding security in processes and deployment of security throughout marketing, development and service into maintenance. Requirements included in this Chapter appear from the fact that they are driven by business considerations in regulated markets. Legislation has a high impact in the system specially when security aspects must be guaranteed. Common aspects of security appears between the different business units that have to be considered when developing products. The security life cycle proposed in Chapter 2 and also the reference architecture based on standards proposed in Chapter 4 may be useful for the identification of security requirements that may improve the process and procedures indicated in this Chapter.

Section II Overview: Other Quality Attributes

- **Section Editor**
 - Anne Immonen (anne.immonen@vtt.fi)
- **List of contributions**
 - Chapter 6. Quality of Service for Real-time and Embedded Systems (Thales & CEA)
 - Chapter 7. Resource usage (Philips)
 - Chapter 8. Predicting Reliability and Availability at the Architectural Level (VTT)
- **List of participants**
 - Thales (Laurent Rioux)
 - CEA (Sebastien Gerard, Hubert Dubois)
 - Philips (Marcel Weijenborg)
 - VTT (Anne Immonen)




Section II Overview: Other Quality Attributes (Cont.)

- **Problems and objectives of the chapters:**
 - To provide a standard solution to develop Real-Time Embedded (RTE) systems:
 - Modelling the Quality-of-Service (QoS) of RTE systems with UML (Unified Modelling Language)
 - Using component-based approach
 - Validating and simulating UML models with QoS
 - To enable the broader usage of software components
 - Having components that adapt to multiple contexts
 - Having different component implementations
 - To predict reliability and availability (R&A) of system family from the architectural models
 - Defining R&A goals and evaluation criteria
 - Representing R&A properties in architectural models
 - Evaluating that the architecture meets the criteria

TELVENT FAMILIES Task 3.2 CWD
Execution Qualities
© Telvent; Miguel A. Oltra, © VTT; Anne Immonen 20 Eureka Σ! 2023 Programme, ITEA project ip02009, FAMILIES 17/10/2005

Among the problems and objectives being covered in the three chapters of Section II are:

- The provision of a standard solution to develop Real-Time Embedded (RTE) systems by means of modelling the Quality-of-Service (QoS) of RTE systems with UML (Unified Modelling Language). The approach being followed is based on a component model approach and will be useful for validating and simulating UML models with QoS.
- How to solve the problem of enabling the broader usage of software components has been accomplished in chapter 7 by means of having components that may be adapted to multiple contexts. This will be done having different component implementations.
- The prediction of reliability and availability (R&A) of system family from the architectural models is a not easy task. Trying to solve or facilitate this by defining R&A goals and evaluation criteria and representing R&A properties in architectural models has been develop in chapter 8. Also it is interesting to answer how to evaluate that the architecture meets the required criteria.



Section II Overview: Other Quality Attributes (Cont.)

Overview of results:

- **(Thales & CEA) UML profile for real-time embedded systems**
 - a standard way to model real-time and embedded systems, which enables
 - 1) definition of a gateway among tools, 2) use of MDA approach, and 3) development of tools for RTE purposes
- **(Philips) Context awareness components**
 - an architectural approach where components
 - 1) are prepared to be used in various contexts, 2) learn dynamically the context, and 3) recognise the intended use and act accordingly
- **(VTT) A method for predicting reliability and availability (R&A) at the architectural level**
 - a method that predicts R&A using three phases:
 - 1) defining R&A goals, 2) representing R&A in architectural models, and 3) analysing the architecture

TELVENT FAMILIES Task 3.2 CWD
Execution Qualities
© Telvent; Miguel A. Oltra, © VTT; Anne Immonen

21

17/10/2005
Eureka Σ! 2023 Programme, ITEA project ip02009, FAMILIES

The UML profile is a standard way to model real-time embedded systems. Among the profile benefits:

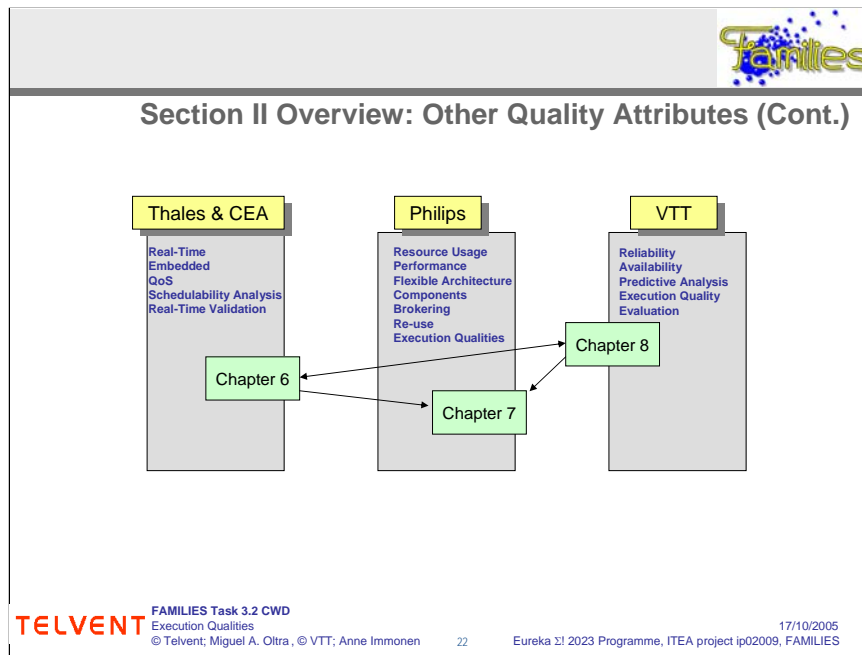
- gateway between several tools (UML tools, validation tools and simulations tools)
- permitted use of the MDA approach for developing real-time embedded systems
- tools for RTE engineers and architects

Context awareness components anticipate their usage context and adapt CPU and memory usage according to each specific context. Among the context awareness components benefits:

- flexible components (outside-in learning: components learn from environment, inside-out learning: components provide optional interfaces)
- a flexible architecture that allows easy prototyping and implementing alternative approaches (variation points: adding or swapping new variations)

The RAP (Reliability and Availability Prediction) method assists in definition of reliability and availability goals for system family and systems, architecture modeling and also reliability and availability analysis from architectural models. Benefits of the method:

- Reliability and availability of the systems of the system family increase
- Quality of components from different component suppliers can be validated and proved in the context of system family
- Maintenance costs decrease and moreover less resources, modifications and fault repairs are needed
- Risks are lower as it can be ensured that architecture meets the requirements

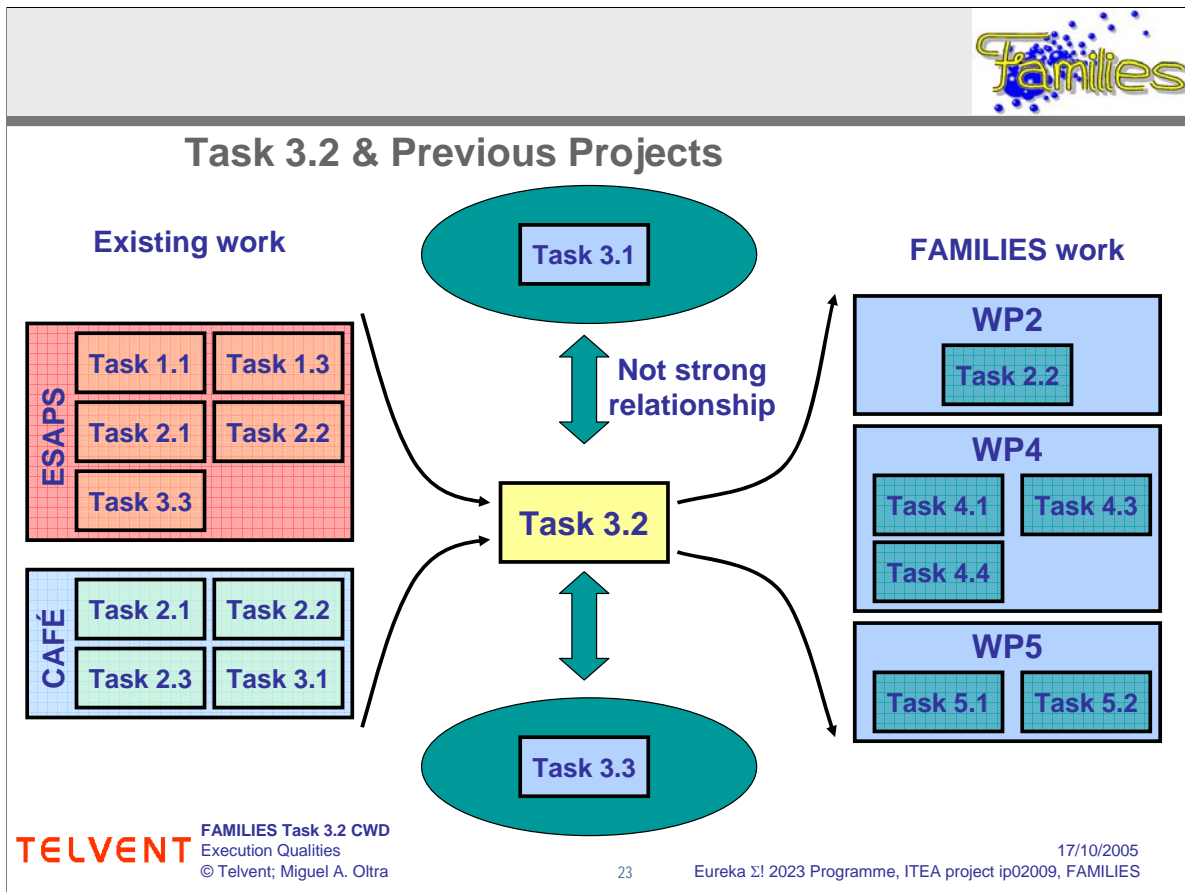


Commonalities/differences/interrelations of Contributions:

Chapter 6 (Quality of Service for Real-time and Embedded Systems) provides a standard profile for modelling quality-of-service for real-time embedded systems. This profile might be applied for modelling performance analysis (Chapter 7) and reliability and availability (Chapter 8) of systems. Thus, performance, reliability and availability analysis are *RTEQoS* (*real-time and embedded quality-of-service*) characteristics.

Chapter 7 (Resource usage) provides component and architecture models that allow the architecture and its components to adapt themselves to different resource requirements depending on the target contexts.

Chapter 8 (Predicting Reliability and Availability at the Architectural Level) provides a three-phased method for predicting reliability and availability from the architectural models. Phases 1 and 2 of the method can be applied to other quality attributes as well (e.g. performance or any other quality-of-service characteristics). Phase 1 helps to define quality goals and phase 2 assists in representing quality attributes in architectural models. The third phase of the method (evaluation) is attribute-dependent.



The figure tries to summarise the different relationships among the work done within the scope of task 3.2 Execution qualities and previous work done in CAFÉ and ESAPS projects, and moreover the relationships with another contributions tackled within the FAMILIES project.

Topics covered in several tasks of both ESAPS and CAFÉ projects are close related to the problems that are being covered in the several topics within the task 3.2 of the FAMILIES project. Following are summarised the identified task from previously mentioned projects:

ESAPS

- Task 1.1 Architectural analysis and modelling
- Task 1.3 Aspect analysis and modelling
- Task 2.1 Definition of System Family processes
- Task 2.2 Reference architecture
- Task 3.3 System family variant configuration and derivation

CAFÉ

- Task 2.1 Requirements engineering
- Task 2.2 Platforms
- Task 2.3 Design for quality
- Task 3.1 Change management

Following is presented the relationships among Task 3.2 and the rest of the tasks being covered within the FAMILIES project:

WP3:

- Task 3.1 Needs fulfilment qualities (Not strong relation)
- Task 3.3 Evolution, adaptation and maintenance qualities (Relationships among VTT contributions)

WP2: Some contribution can fit quite well with proposed best practices

- Task 2.2 System families maturity practices

WP4: Partners working on MDE and MDA case studies

- Task 4.1 Domain and application modelling practices and techniques (Thales &CEA;Telvent & UPM):
 - "Methodological guide for model-based variability modelling"
- Task 4.3 Model transformation for MDFE (VTT)
- Task 4.4 Model Driven Family Engineering Supporting Practices (Telvent & UPM)

WP 5: Families integration: relationship with inclusion of third party or COTS

- Task 5.1 Architecture consequences of integration
- Task 5.2 Process and organisation consequences of integration (Telvent & UPM)



Shaping a World
of Convergence



Families Task 3.2 CWD
Execution Qualities

Section I: Security

Miguel A. Oltra
miguel.oltra@telvent.abengoa.com



Task 3.2
Execution Qualities



Chapters: Section I

- Chapter 1. Quality determination and documentation techniques (Nokia)
- Chapter 2. System family security (UPM)
- Chapter 3. Security Reference Architecture (ICT-Norway)
- Chapter 4. Security issues in dynamically deployable SFs (for distributed systems) (Telvent)
- Chapter 5. Improving security quality (Philips)



TELVENT Shaping a World of Convergence

Families Task 3.2 CWD

Quality determination and documentation techniques

Security modelling and analysis

Juha Savolainen
Nokia Research Center
Juha.Savolainen@nokia.com

NOKIA

Task 3.2
Execution qualities

Abstract:

Architectural documents are really valuable only if valuable they directly answer to the questions posed by the architecturally significant requirements. In this work we show how security concerns can be showed in architectural diagrams.

Keywords:

Architecture, Design, Quality attribute, Modelling, Architecture evaluation, Variability modelling in quality attributes

Relation to previous work in ESAPS and CAFÉ:

Task 2.1, Task 2.2 of ESAPS and CAFÉ Task 2.1, Task 2.2, and Task 3.1

Relation to other tasks in WP3 or other Work Packages of Families:

Task 3.2 chapter 3

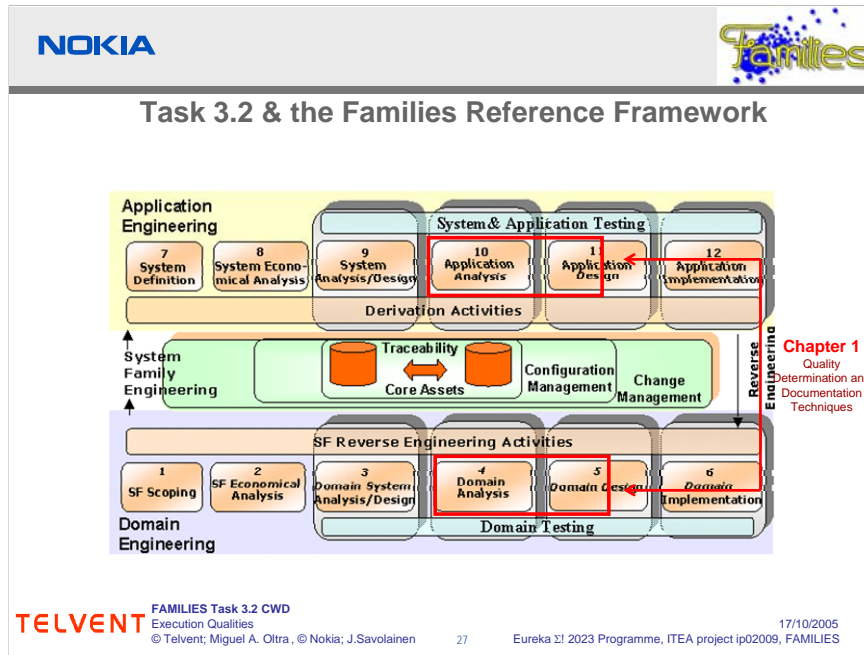
Acronyms used in the contribution & Glossary of unusual Terms used:

ASR: Architecturally Significant Requirement

QA: Quality Attribute


References:

- 1.Savolainen, J., Oliver, I., Mannion, M., Zuo, H., Transitioning from Product Line Requirements to Product Line Architecture, in the proceedings of COMPSAC 2005, to appear.
- 2.Kuusela, J., Savolainen, J., Requirements Engineering for Product Lines, in International Conference on Software Engineering ICSE2000, IEEE, 2000, pp. 61-69.
- 3.J. Savolainen, and J. Kuusela, "Consistency Management of Product Line Requirements", In Proceedings of the Fifth IEEE International Symposium on Requirements Engineering, IEEE, Toronto, Canada, August 27-31, 2001.
- 4.Savolainen J, Vehkomäki T., Mannion M. "An Integrated Model for Requirements Structuring and Architecture Design", Proceedings of the Seventh Australian Workshop on Requirements Engineering, pp. 19-33, 2002.
- 5.Jackson, M., Problem Frames, Addison-Wesley,2000.




•Chapter 1: Quality determination and documentation techniques (Nokia)

Chapter 1 is centred on both Domain and Application Engineering levels in activities related to mechanisms for analysis and design activities.

NOKIA

Introduction & Problem Description

- **Goals of the architecture documentation**
 1. Give an introduction that helps the understanding
 - It must be concise, clear specification that concentrates to the main roles of the architecture
 2. Demonstrate that the architecturally significant requirements are fulfilled
 - It must contain the main ASRs
 - Key issues is to answer the challenge presented by the ASRs
 - The fulfilment of the ASRs must be apparent
- **Problem: How to describe and analyse quality properties from the architectural models?**
 - Which models are needed?
 - What kind of enteritis should be described and how?
 - What kind of annotations are needed?
 - What views should be used?
 - What kind of measures can be defined?
 - How the analysis is done?

 **FAMILIES Task 3.2 CWD**
Execution Qualities
© Telvent; Miguel A. Oltra, © Nokia; J.Savolainen

28


17/10/2005
Eureka Σ! 2023 Programme, ITEA project ip02009, FAMILIES

The software architecture is a risk management approach that tries to control the fulfilment of the quality attributes in the final system. This is done by producing models of the system that allow analysing the characteristics of the system before it is implemented. This relies our ability to express those properties in an easily communicated manner that makes the fulfilment of the wanted characteristics explicit.

Effective communication of architectural properties can be only achieved with a limited documentation. When the amount of the architecture documentation increases the benefits of the documentation decreases rapidly. It is very hard to find inconsistencies from a document that spans over 500 pages.


We argue that if it is possible to describe architecture in a way that provides concise document that directly addresses the wanted characteristics we are able to more effectively find out potential problems of the architecture. In addition, we claim that it is possible to attach various measures to those architectural views in a way that assists evaluating the properties of the architecture.

Therefore, our main research question is to find the correct ways to describe software architecture, in the product line context, that allows easy specification of the wanted quality attribute. We have chosen security to represent our approach.

NOKIA

Relevance & Expected Benefits

- **Need for smooth refinement of quality properties**
 - Transition from the conceptual mechanisms towards the realization
- **Give common ways to show the quality properties in selected architectural diagrams (using UML)**
 - Easier communication and improved quality determination by sharing experiences
- **Result: Ability to model architectures that can be analysed easily against the QAs**
 - Focus on the Security
- **Benefit: Earlier quality determination**
 - Lowered costs by reducing needs for redesign
 - Higher quality systems that more likely satisfy the key customer requirements

 **FAMILIES Task 3.2 CWD**
Execution Qualities
© Telvent; Miguel A. Oltra, © Nokia; J.Savolainen


29

17/10/2005
Eureka Σ! 2023 Programme, ITEA project ip02009, FAMILIES

In our previous work we have investigated how it is possible to describe product line requirements [3] and how we can connect requirements and the quality attributes to the relevant design choices [2]. In this work we focus on connecting the wanted characteristics of the system in to the actual architectural models.

The earlier we can find potential problems in our systems the better off we are. Many architectural evaluation methods have been proposed that can also be used. However, even the evaluation relies on the fact that the architecture has been created and is, at least partially, finished. If we could advance the earliest evaluation point of the architecture to the actual design phase – many potential benefits may arise.

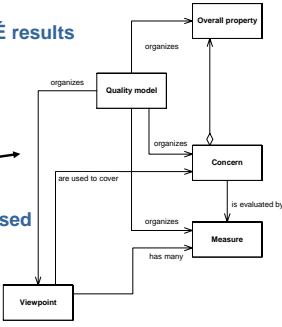
Earlier detection of the actual properties of the architectures potentially lowers cost by reducing the amount of effort that may be, potentially, reengineered. Full lifecycle quality enforcing mechanisms bring promise of higher quality systems that better match the actual needs of our customers.

NOKIA


Approach & Expected Results

1. Collect existing viewpoint definitions
2. Tailor the viewpoints for the sub-characteristics in the quality model
3. Define measures for viewpoints and characteristics
4. Apply in concrete cases

- Approach is based on the ESAPS and CAFÉ results
 - ESAPS
 - Task 2.1
 - Task 2.2
 - CAFÉ
 - Task 2.1
 - Task 2.2
 - Task 3.1
- Existing quality models
- Existing metamodel
- It is also based on the viewpoints actually used
 - Collected from the company ADs



```

            graph TD
                Viewpoint -- organizes --> Quality_model[Quality model]
                Viewpoint -- organizes --> Concern
                Viewpoint -- organizes --> Measure
                Viewpoint -- are used to cover --> Concern
                Viewpoint -- has many --> Measure
                Quality_model -- organizes --> Overall_property[Overall property]
                Concern -- organizes --> Overall_property
                Measure -- is evaluated by --> Concern
            
```

TELVENT FAMILIES Task 3.2 CWD
Execution Qualities
© Telvent; Miguel A. Oltra, © Nokia; J.Savolainen

30

Eureka Σ! 2023 Programme, ITEA project ip02009, FAMILIES

17/10/2005

We build on top of previous results done in the CAFÉ and ESAPS projects [see e.g.4]. These include using the already defined quality metamodels and defined ways to describe quality attributes.

These models are now applied using the collected practical viewpoints. These viewpoints show the current ways that we in Nokia use to describe architectures. The viewpoints are adapted to accommodate the quality characteristics by adding new types into the model and deriving existing types with quality annotations.


We have chosen to use existing viewpoints to describe the qualities. This is chosen mainly for the practical reasons, since those viewpoints already exist in the company. However, there is a more general issue whether a generic viewpoint or a quality attribute specific viewpoint is better choice to support this work.

Using generic views has many benefits. They are easily understood, since the already known view is only slightly modified to add the quality information. This makes them easy to be used to model many products and variants, since they are not as domain specific nor do they concentrate only to one quality, which may not be of interest of another product. The generic view is, therefore, ideal to describe tradeoffs among QAs, since all of them can be potentially described in the same architectural view.

On the other hand quality specific views concentrate one quality at a time reducing the amount of entities that must be presented, since the view concentrates on the issues that directly address the quality. This allows easy analysis and easier formulation of rules.


Next step in our approach is to define measures for each of the viewpoints to actually say whether the selected quality is fulfilled by the concrete architecture specified in the architectural view. If the architecture is described using the quality annotated viewpoint we can describe if the selected solution is contradicting some wanted quality characteristics.

Finally we use a concrete case study to demonstrate how the approach is used to create a real system.

NOKIA

Risks

- **The usefulness of the model**
 - More specific model is always more useful
- **Domain independence**
 - More independent on the domain characteristics – less useful
- **Relies on the architect's abilities**
 - The software architect must do the mapping between the system characteristics and the architectural elements



 **FAMILIES Task 3.2 CWD**
Execution Qualities
© Telvent; Miguel A. Oltra, © Nokia; J.Savolainen

31

17/10/2005
Eureka Σ! 2023 Programme, ITEA project ip02009, FAMILIES

As with any method, a more concrete method that really solves and evaluates the concrete architecture thoroughly is preferred by its users. However, such a method always carries a lot of domain knowledge. This makes adapting such a method outside of its application domain difficult. A careful balance must be achieved between the usefulness and the domain independence of our approach.

In this approach, there is no formal way to prove that the requirements are satisfied by the architecture, rather we explicitly rely on the abilities of the particular software architect to make the connection.



Type-based architecting complex systems

- **Typing is in the essence of architecting**
 - Pursue for simplicity
 - Improved expressiveness
- **Architecture is about constraining**
 - Constraining how the lower level abstractions can be implemented
- **Architecture is risk management**
 - If there is no risk – then there is no need for architecture


TELVENT FAMILIES Task 3.2 CWD
Execution Qualities
© Telvent; Miguel A. Oltra, © Nokia; J.Savolainen 32 Eureka Σ! 2023 Programme, ITEA project ip02009, FAMILIES 17/10/2005

An effective architecture requires that it is not a random collection of entities with clever names. Good architecture is composed of limited set of types that have well known properties and restrictions. Therefore, the type should describe the usage and the intent of that set of architectural elements, but even more importantly – the type should constrain how the type can be used.

Restrictions are the key for maintaining the architectural consistency. When the reference architecture is designed by formulating the architectural rules to be embedded into the architectural types, then those architectural rules are maintained during the system design. In an ideal case the architectural rules are maintained by the tool set used by the software architects and developers.

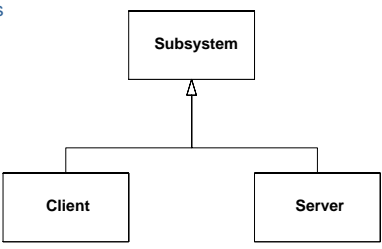
In time an effective set of architectural types create a common language within the project team. Everyone knows what is assumed when an element is declared to be a server or when a client connects with the server.

Having a common language makes communication much easier because the team does not go into the common details of every element, rather they can focus on the differentiating facts. The types and their know properties, therefore, makes exposing potential problems easier.

NOKIA


What should we do then?

- **After defining the common definition for each view**
 - Define the architectural archtypes that correspond the wanted characteristics
 - E.g. <<client>> and <<server>> that are both of the type <<subsystem>>
 - Use these archtypes to populate the architecture
 - Each of the archtype constrains the architecture further
 - Estimate the properties of the architecture using the know constraints of the architectural elements



```

graph BT
    Client[Client] --- Node1[ ]
    Server[Server] --- Node1
    Node1 --> Subsystem[Subsystem]
  
```

TELVENT FAMILIES Task 3.2 CWD
Execution Qualities
© Telvent; Miguel A. Oltra, © Nokia; J.Savolainen

33

17/10/2005
Eureka Σ! 2023 Programme, ITEA project ip02009, FAMILIES

Some of the types can be more or less domain independent. The generic terms such as “subsystem” or “layer” must be defined so that their definition is clear. Then domain dependent types can be refined from the more generic types. The properties of the generic types are inherited by the domain specific constructs, but then should add new constraints how these types should be used in the domain specific architecture.



Most real life, large software projects require that the architecture is represented from multiple view. Each of these views describe the system from a different angle, focusing on the certain characteristics of the system. Some of these views are common to most of the system. Ideally, these views are populated by the domain specific types that are derived from the generic types that are part of the particular view.

A common architectural view described decomposition. Each non trivial system should show how it is divided into parts. The whole system is divided into subsystems witch can be further divided into other modules. First it is important to define what are the terms used.

Some of the relevant questions are:

- Are the subsystems only used in the highest level?
- Can a subsystem contain other subsystems?
- What does the subsystem contain?
- If modules, then what is their definition?
- Does the module contain actual code or binaries or some conceptual elements?
- How is everything eventually mapped to the assets that really exists, classes, files, files structures?

The architectural type system must be grounded on reality, on things that can be verified to exist. This is a process that should be done to every view. On the lowest level there has to be file folders, actual files, build scripts, DLLs and so forth.



Typing and Quality Characteristics

- **Refinement and constraints**
 - Refinement of qualities and mechanisms
 - Improving the accuracy of estimations
- **Ideal usage of architectural types**
 - Define the types
 - Use requirements and problem domain knowledge
 - Estimate the properties of each type
 - Annotate with quality characteristics
 - Formulate rules on how the types can be used
- **Make the satisfaction of the quality characteristics explicit**
 - Show in the architecture descriptions how each quality is achieved
 - Support refinement


TELVENT FAMILIES Task 3.2 CWD
Execution Qualities
© Telvent; Miguel A. Oltra, © Nokia; J.Savolainen 34 Eureka Σ! 2023 Programme, ITEA project ip02009, FAMILIES 17/10/2005

The design process increases the understanding of both the design and the requirements that have been imposed to the system. When we design systems, we realise how the chosen solutions fulfil the requirements. Similarly, the design choices deepen our understanding on the various ways to satisfy the requirements. This gives us insight whether the requirements are realistic and if a system implementing them is possible to be built.

Ideally, we can show refinement in both the requirement and design side. While refining the system design into its final realisation – we should continuously try to match the design choices and the requirements why those choices were made.

A key thing is to be able to made the mapping in a suitable level of abstraction. Both requirements and design decisions should be refined throughout the architecting process and the connections should happen at the same level of an abstraction. If we take a very generic quality attribute such as security or performance, that can be mapped to almost any design element. Every line of code has some implication on the performance of the system and coding mistakes may jeopardize the security of the system regardless where they take place.

But if we can make the connection in the right level then good traceability can be achieved. The refinement of performance scenario – “response time of two seconds for credit card validation” can be connected to design choices. Now it is possible to map the requirements to the corresponding architectural elements. This allows justifying the architectural decisions with the requirements and previous design decisions.

NOKIA


Transitioning from requirements to features

- **Requirements are situated in the problem domain.**
 - They are used to describe the needs of customers.
- **During product line development a product line model of requirements can be constructed and used to make selections to generate a new product.**
 - It can be helpful to organize the model as a forest, in which the requirements are related to each other in parent-child relationships.
- **Features are situated in the solution domain.**
 - During product line development a product line model of features can be constructed and used to make selections to generate the assets of a new product.
 - Information on the requirements of the product family is needed to create the feature model.
- **The construction of the feature model requires the knowledge on the architecture – the feature model should reflect the variability that exists in the software architecture.**
 - Similarly how the architecture partially determines the structure of the feature model also the features influence the requirements.
- **A set of rules can be defined to connect the variability in the requirements into variability of the features**

TELVENT FAMILIES Task 3.2 CWD
 Execution Qualities
 © Telvent; Miguel A. Oltra, © Nokia; J.Savolainen

35

17/10/2005
Eureka Σ! 2023 Programme, ITEA project ip02009, FAMILIES

The rules for mapping requirement variability to the features variability is specified as follows:

Rule 1 (Mandatory Requirements): If any requirement in the set of requirements specifying a feature is mandatory then the selection criterion of the feature is itself mandatory

Rule 2 (Optional Requirements): If any requirement in the set of requirements specifying a feature is optional and no mandatory requirements exist in that set, then the selection criterion of the feature is itself optional.

Rule 3 (Obsolete Requirements): If all requirements in the set of requirements specifying a feature are obsolete then the selection criterion of the feature is itself obsolete.



Rule 4 (Non-Reusable Requirements): If all requirements in the set of requirements specifying a feature is non-reusable then the selection criterion of the feature is itself obsolete.

Rule 5 (Multiple and single adaptor mapping to optional): If the children of a single or multiple-adaptor requirement each depend upon mutually exclusive features (or set of features each) that are independent of any other requirements, then each of the child requirements should be treated as optional for the selection constraint propagation purposes and thus (by earlier rules) their related features are optional too.

Rule 6 (Feature Composition): if a requirement R1 has child requirements R1.1 and R1.2, and feature F1 has child features F1.1 and F1.2, and R1.1 specifies F1.1 and R1.2 specifies F1.2, then this can be composed to the simple relationship R1 specified F1. Rules 1 to 4 then apply to set the selection criterion of F1.

Rule 7 (Variable mapping to mandatory): If a feature is specified or implied by each variable requirement wherever it is in the forest of requirements then its selection constraint value is mandatory.

Because of the limited space, we do not cover this part of the process in detail. An interested reader should consult the relevant publication [1].



Representing Security Concerns

- **Problem domain models**
 - Problem model is important for security since it allows describing how the security concern relates to the overall problem that the system is intended to solve
- **Structural architecture models**
 - In architecture models we can connect the requirements for security to the mechanisms that we use to satisfy the security requirements
 - Often those diagrams are made in various times during the software development promoting the ability to specify both requirements as well as mechanisms in various levels of refinement
 - Architectural diagrams are also optimal place to analyse the tradeoffs between requirements
- **Deployment diagrams**
 - The deployment eventually captures how the security can be practically done
 - It is crucial to describe the physical security aspects

TELVENT FAMILIES Task 3.2 CWD
Execution Qualities
© Telvent; Miguel A. Oltra, © Nokia; J.Savolainen 36 Eureka Σ! 2023 Programme, ITEA project ip02009, FAMILIES 17/10/2005


Security concerns can be represented in multiple views on the architecture. Different aspects of the security are exposed in these architectural views. In this paper, we use three different viewpoints: problem domain, structural architecture, and deployment.

Problem domain models are ideal to capture requirements and features, whereas structural and deployment models reflect how the model entities are connected to the properties of those selections.

Structural architecture shows the mapping between the detailed security requirements and the architectural elements.

Deployment describes how the architectural elements are in fact connected to actual hardware units.

Our architectural viewpoints try to support incremental design process. Initially, requirements impose constraints on which kind of design alternative can be chosen. Then the general approach is chosen. An architecture style with assigned responsibilities limit the number of actual architectural mechanisms that can be chosen in the form of different tactics. Finally, an actual implementation mechanism has be selected in combination with the tactics to complete the security approach for the system.

NOKIA

Problem domain modelling

- **The problem domain modelling specifies the security concerns as they are represented within the current context**
 - The problem domain model can be based on the types defined for the generic security concerns
 - Those aspects of the security model that are not in concern of the current problem are ignored
 - The problem domain model shows how the security affects the problem
- **Generic constructs**
 - Concept or an entity and their definitions
 - Relationship between entities
- **Security specific constructs**
 - Security specific entity roles (as defined by Sintef)
 - Security requirements affecting entities and relationships

TELVENT FAMILIES Task 3.2 CWD
Execution Qualities
© Telvent; Miguel A. Oltra, © Nokia; J.Savolainen

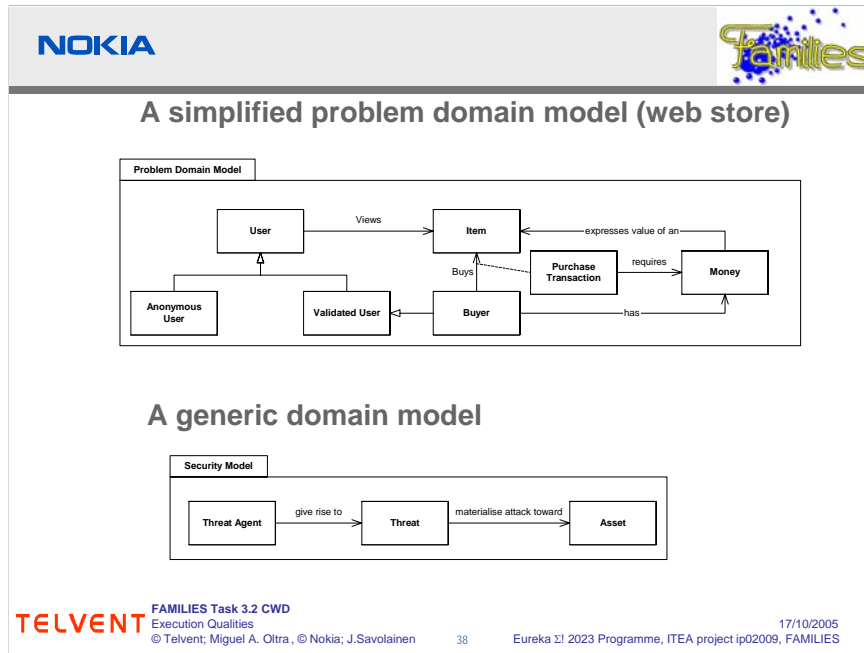
37

17/10/2005
Eureka Σ! 2023 Programme, ITEA project ip02009, FAMILIES

The goal of the problem domain model description is to create a model of the domain that has all the relevant constructs and allows discussing the key requirements of the system. The requirements should be possible to be phrased using the terms in the problem domain model. There should be a way to express any requirement in the model.

Some authors even suggest to model the requirements themselves graphically [5]. Even though this may be beneficial to make the connections between the requirements and problem domain model constructs very concrete, doing this for a large system may be an overkill. Therefore, we assume that the requirements are done using natural text.

The final problem domain model can be obtained by combining two different models. First, we do the initial problem domain model as always. Then we pick the relevant constructs from the generic security model. Finally we combine these two models as described in the next two slides.



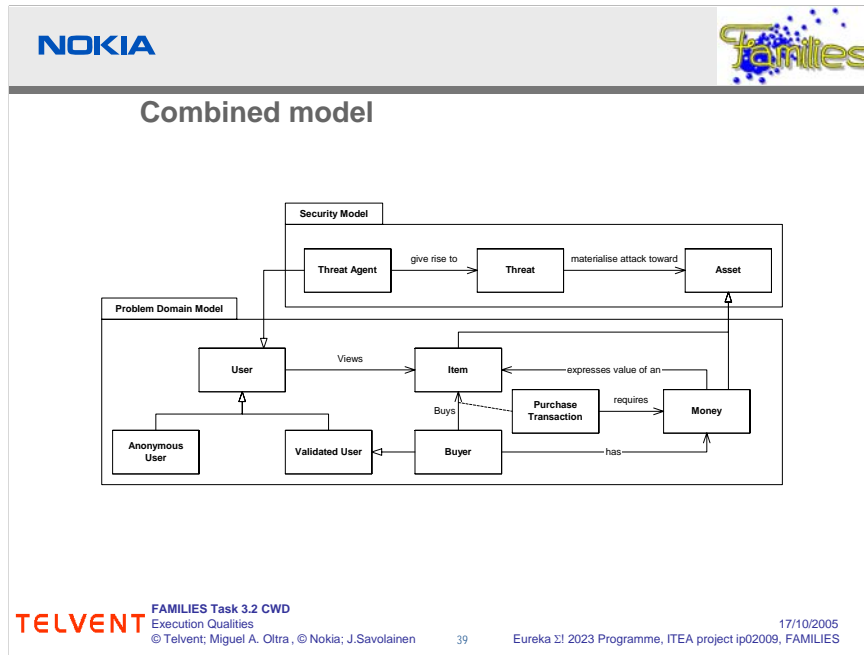
The simplified domain model describes the key concepts of the web store application. It only defines the main terms of the web store without any specific focus to the security concern. There is an user that views items. If the user logs in to the application then he becomes a validated user. This type of user can act as a buyer that purchases items. This happens via a purchase transaction that requires money.

This generic security model is adapted from the SINTEF's conceptual model that itself is based on the common criteria. For each concern (here security) a generic model can be created. This model defines the main terms and concepts that can be used to describe what security means.

As described here, the basic domain model is not really tied to the security concern – it rather defines quite generic concepts of any web store. Naturally any real domain model would be more extensive and the constrains in the model be thoroughly defined.

The generic security model, on the other hand, has no information on the web stores. It has a generic model that can be applied to many occasions where a security is a concern.



These two models, independently, provide only limited value. But combining the models brings out the true value of the approach.



In this slide we present the combined model. It shows how the generic security model can be adapted to the current context.

We see that the threat agent is now defined to be a type of a user. And that the money is an asset where the threat focuses on.

It is clear that the model is a simplified version of the complete, real world model. One could easily discover other types of threat agents besides the actual user. But for this example the new combined model provides enough information.



Structural architecture

- **The structural architecture describes the division of the system into pieces and specifies how the chosen techniques facilitate achieving security properties**
 - The structural architecture uses the basic types but extends those by describing the concrete mechanisms
 - The mechanisms used affect security characteristics
- **Generic constructs**
 - Components
 - Connectors
 - Ports, Interfaces
 - Protocols
- **Security specific constructs**
 - Security zone + Threat specification
 - Security mechanisms (e.g Firewall)

TELVENT FAMILIES Task 3.2 CWD
Execution Qualities
© Telvent; Miguel A. Oltra, © Nokia; J.Savolainen 40 Eureka Σ! 2023 Programme, ITEA project ip02009, FAMILIES 17/10/2005

The structural architecture is clearly part of the solution domain. It shows how the system is divided into components and how those components are connected together.

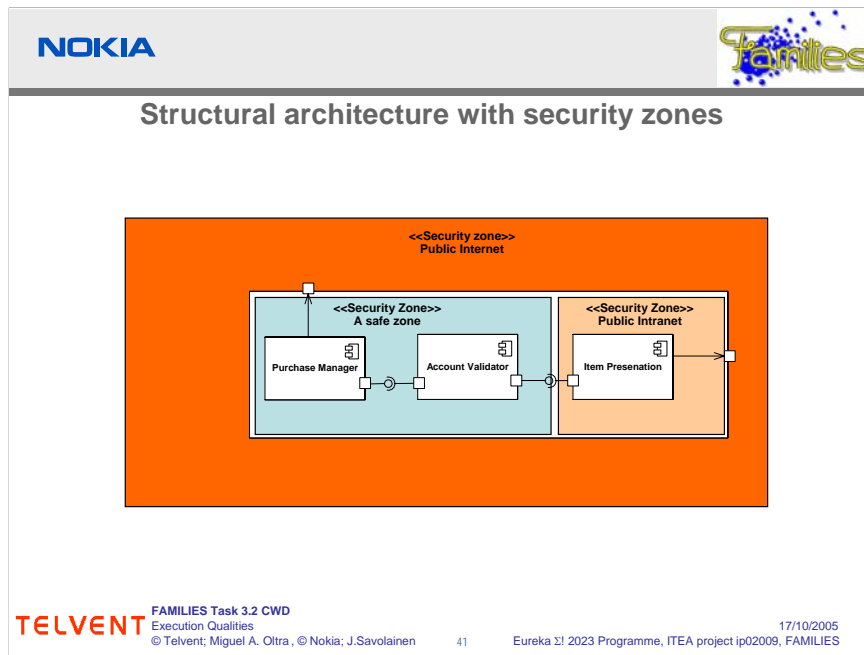
Additionally, it describes the security specific constructs in the same diagrams as the architecture. The main types of the security constructs are different security mechanisms and security zones.

Modern application development cannot rely on truly safe zones. Also behind e.g. the firewall various techniques should be used to secure local computers. Clearly multiple tiers of countermeasures provide better security than only one boundary layer relying in one specific technique.

However, security zones still play a role in the architecting secure systems. Different zones allow rationalizing on security levels and techniques, since the threats vary along the zone.

It is very natural to use different methods for boundary crossing interaction across e.g. public internet than for communication that takes place inside a boundary e.g. in the intranet. Therefore, typically the security zones align with the deployment of the system, but the the concern is still driven by the security problem not by the architecture. The security zones should be selected and scoped based on the threat that this area is exposed to.

However, aligning the security zones with other similar overlapping areas reduces the number of types and allows easier expressing on important concepts making the architectural type language more expressive.



Here we have used the same web store example in the structural architecture model.



The architecture is composed of three components. Item presentation component describes items to users. It can be used by both validated and anonymous users. This means that no account validation can be required by the users. Therefore, we declare this component is part of the public internet security zone.

The remaining two components are the account validator and the purchase manager. The account validator is responsible to allow user to register into the service and validate them as registered users that can make purchases. The account validator thus uses the security tactic access control.

The purchase manager is responsible for allowing the validated user to make a purchase transaction. This component guarantees that the user will eventually get the item that he purchased and that the money is eventually charged from the user's credit card (using the external interface).

The external interface from the purchase manager is specified by the credit card authorization organization and uses security tactic encryption when connecting over the security zone public internet.

The components and security zones also allow easy specification of architectural rules. For our system we formulate a generic security rule – use encrypted connection (SSL) when transmitting confidential information over security zone public internet.



Deployment architecture

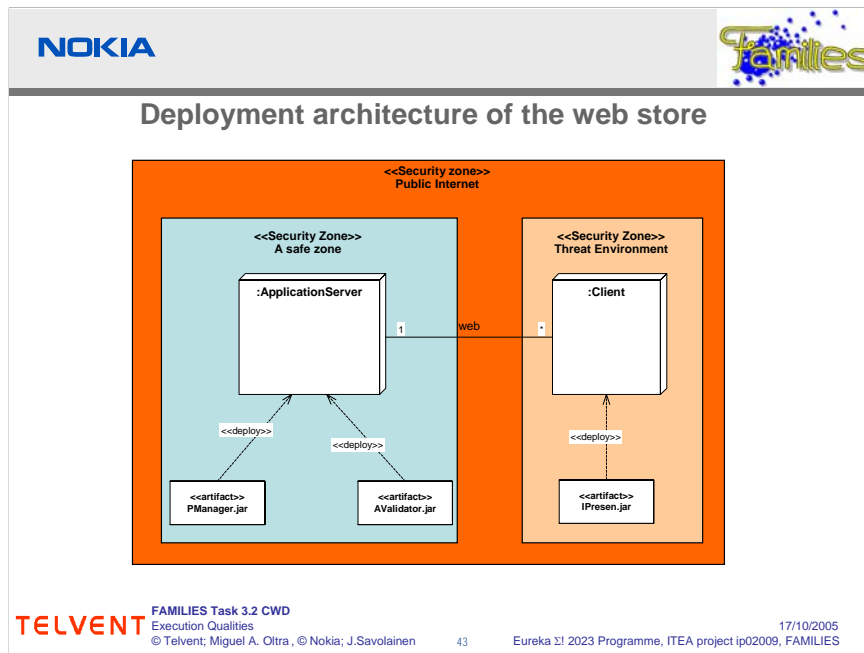
- **Generic constructs**
 - Node
 - Communication link
 - Artifact
- **Security specific constructs**
 - Mapping security zones to nodes
 - Mapping threats to nodes
 - Node specific security mechanisms (e.g. HW based encryption)

TELVENT FAMILIES Task 3.2 CWD
Execution Qualities
© Telvent; Miguel A. Oltra, © Nokia; J.Savolainen 42 Eureka Σ! 2023 Programme, ITEA project ip02009, FAMILIES 17/10/2005

Deployment architecture describes how the components are mapped to the actual hardware. The generic constructs are the nodes, artefacts that are deployed into the nodes and communication links between them.

Security concern is presented in this viewpoint by security zones overlapping some nodes and communication links, threats on generic constructs and security mechanisms that are used as countermeasures to the threats.


The deployment view defines constraints on the allocation of software entities to processing nodes. On the abstract level this means specifying constraints on deployment and on concrete level actual mapping of the software element to the processor. Nodes represent either hardware devices or software execution environments. Nodes can be (and often are) nested and artifacts are deployed on the nodes.



In this slide, we show a very simplified picture of the deployment architecture of our web store. It describes that all of the components are deployed as their own jars. The Item Presentation component is actually deployed on the client machine where it allows us to make a special, customized entrance to our web store. This, however, means that this component is part of the high risk environment – here represented as threat environment. Since the component is deployed to the client machine it makes it very vulnerable for reverse engineering, communication sniffing and other hacking approaches. We have decided that no major countermeasure is used here. The component only takes care of the presentation and allows SSL connections to take place, but otherwise it does not contain any such information that would make it a valuable asset to attack the web store.


The server side is deployed in to a safe domain. Both the Account Validator and Purchase Manager components reside in this zone. The main countermeasures are physical ones. The computer running the web store is placed in safe location, locked server room. Further security could be obtained by having the two components running in different machines. Then getting the root password into one system would not jeopardize the whole web store. But even then the ability to get root access to the Account Validator allows the hacker to soon get purchase transactions to the other users credit. For this reason we decided to only run intrusion detection software in the server machine with automated scripts to close the connection (and make alarm) if intrusion takes place.

The communication between the server and the client side takes place through public internet where non closure information is a clear security concern. Use encryption as our countermeasure. Again the threats and countermeasures are not shown here for simplicity.

NOKIA

Conclusions

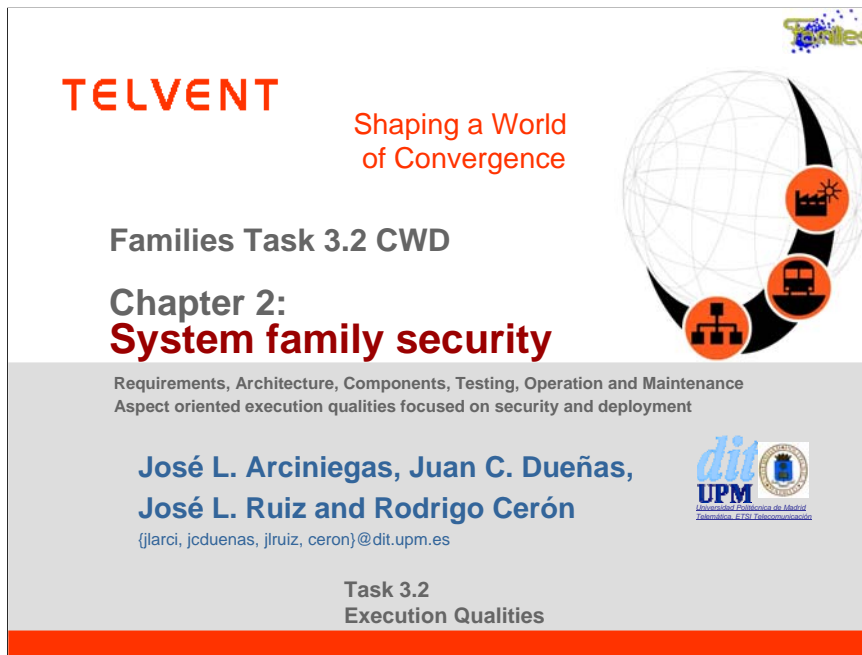
- Security architectures can be modelled by combining security specific models and generic model constructs
- Security aspect is shown by defining specific types that constrain how to models can be created
- Both the models of the problem/system as well as solutions should be refined when the design progresses

 FAMILIES Task 3.2 CWD
Execution Qualities
© Telvent; Miguel A. Oltra, © Nokia; J.Savolainen

44

17/10/2005
Eureka Σ! 2023 Programme, ITEA project ip02009, FAMILIES

In this work, we briefly introduced ways to connect the security characteristics to architectural models. The architecture documentation should provide answers to key questions. What is the overall structure of the system and how it fulfils the architecturally significant requirements.



TELVENT Shaping a World of Convergence

Families Task 3.2 CWD

**Chapter 2:
System family security**

Requirements, Architecture, Components, Testing, Operation and Maintenance
Aspect oriented execution qualities focused on security and deployment

**José L. Arciniegas, Juan C. Dueñas,
José L. Ruiz and Rodrigo Cerón**
{jlarci, jcduenas, jlruiz, ceron}@dit.upm.es

**Task 3.2
Execution Qualities**

Abstract:

This is the UPM contribution to Families Task 3.2 – Execution Qualities. Our contribution is titled “System family security”, it includes requirements, architecture, components, testing, operation and maintenance, aspect oriented execution qualities focused on security and deployment.

Keywords:

System family, Security, life-cycle, execution quality.

Relation to other tasks in WP3 or other Work Packages of Families

Task 5.2 Assets recovery in system family

Acronyms used in the contribution & Glossary of unusual Terms used

AAA: Authentication, Authorization and Accounting

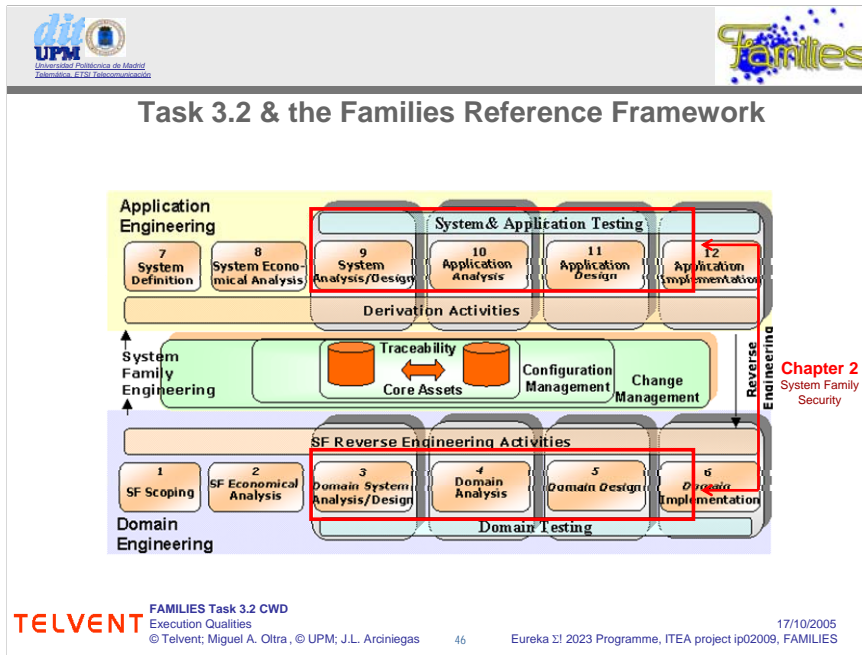
IETF: Internet Engineering Task Force

DMTF: Distributed Management Task Force

MDA: Model Driven Architecture


References

1. SINTEF. Tor Erlend Fægri, Svein Hallsteinsen. Memo Concerns Security reference model. 2003
2. Common Criteria for Information Technology Security Evaluation, version 2.1, August 1999. Available at <http://www.commoncriteria.org>
3. Security Service Specification Version 1.8. March 2002
4. X.200 OSI Reference Model <http://www.itu.int/rec/recommendation.asp?type=items&lang=e&parent=T-REC-X.200-199407-1>
5. The Java Security Architecture for JDK 1.2. Version 1.0, Sun Microsystems, October 1998. <http://java.sun.com/products/jdk/1.4/docs/guide/security/spec/securityspec.doc.html>
6. Gregor Kiczales, John Lamping, Anurag Mendhekar, Chris Maeda, Cristina Videira Lopes, Jean-Marc Loingtier, John Irwin. Aspect-Oriented Programming. Published by Springer-Verlag. 1997
7. JAC Java Aspect Components, Distributed and Dynamic Aspect-Oriented programming in Java. <http://jac.aospys.com>
8. Zope Corporation <http://www.zope.org/Members/pje/Wikis/TransWarp/HomePage>
9. Renaud Pawlak. CEDRIC Research Report: A Notation for Aspect-Oriented Distributed Software Design. Laboratoire CEDRIC-CNAM, 55 rue Turbigo, 75003 Paris, France. 2002
10. <http://www.eclipse.org/aspectj/>
11. James W. Cooper. Aspects, Concerns, and Java. 2003
12. J. Viega, J. Bloch and P Chandra. Applying Aspect-Oriented programming to security. Cutter IT Journal. 2001
13. Tom Idermark, Malte Lilliestråle and Jesper Vasell. An Electronic Service's enabler. 1999. http://www.ericsson.com/about/publications/review/1999_01/files/1999015.pdf
14. C. de Laat, G. Gross, L. Gommans, J. Vollbrecht and D. Spence. Generic AAA Architecture. 2000
15. DMTF. Core Specification 2.8 (UML diagram). 2003
16. DMTF. CIM User and Security Model White Paper. 2003
17. OSGi Service Platform, Release 3, http://www.osgi.org/resources/spec_download.asp




•Chapter 2: System family security (UPM)

Chapter 2 is centred both in Domain and Application Engineering activities, centred in the whole activities, but mainly centred in analysis and design activities.




UPM
Universidad Politécnica de Madrid
Teleinformática #101 Telecomunicaciones



Introduction & Problem Description

- Execution qualities are “out of business” in software engineering
- Need to integrate “qualities” to the usual engineering cycle
- Then, integrate “qualities” to the system family engineering practices
- Specific communities deal with specific qualities: performance, availability, security, etc
- There is no homogeneous view to execution qualities, such as security
 - Security is a extend and complex area
 - There are several partial models
 - Bad practices in software
 - Different treatments depend of life cycle (analysis, design, implementation, testing, operation and maintenance)
- Security is a special problem that spreads over the full lifecycle
 - The main actual force for maintenance
 - The source for actual system families
 - Technology is dynamic
 - New attacks, new threats, new requests for solutions



FAMILIES Task 3.2 CWD
Execution Qualities
© Telvent; Miguel A. Oltra, © UPM; J.L. Arciniegas

17/10/2005
Eureka Σ! 2023 Programme, ITEA project ip02009, FAMILIES

The software engineering defines methodologies, methods, strategies and techniques supported on tools. Traditionally the centre of attention is the functionality of a system. Usually, quality is mentioned in the description of the system and checked when the system is implemented. But how to achieve that? Execution qualities are "out of business" in the software engineering. Therefore we need to integrate qualities to the usual software development process.

If the execution qualities are poorly considered in the classical development process, what is the situation in the systems family engineering?. In this contribution, we study what current practices are done in software engineering and how they can be used in systems family engineering.

There are several execution qualities, but in this contribution is intended to cover system family security aspects, which includes an overall vision about security aspects in system families. The security is only one example, other qualities as performance, availability, portability and so on, can be covered following the same process.

Obviously security is a frequent issue covered in the software system architectural designs and since a long period of time. Nowadays one software tendency is the evolution towards distributed systems architectures through Internet, with an increasing number of inconveniences to keep in mind (e.g. threats, exploits, attacks, software components with an uncertain origin to be installed in the system, ...) and that can be harmful for this kind of systems, in several terms; data privacy, content protection, service interruption, etc.

There is no homogeneous view for security

- Security is an extensive and complex area (several variables can affect security: used technology, network vulnerabilities, new risks, expert users, etc.)
- There are several partial models (CIM model, OMG model, CC model, etc)
- Bad practices in software (Inefficient information management, bad habits in programming, reduced security tests, etc)
- Different treatments depend on the lifecycle (special practices should be achieved per each lifecycle phase, for example: in analysis phase, possible risks should be located, or in design phase, some alternative solutions should be presented)


Moreover, several technologies are involved in issues related to security, from physical level (such as interconnection protocols) to application level (such as user passwords or user permissions to a certain application). How to manage all these different aspects in a simple model is not easy. This document does not try to reinvent the wheel (happening so many times) but it tries to deal with the definition of a security framework valid for systems in which their architectures are based on a component model approach.

The proposed model is based on previous models defined in standardisation working groups (that cover security aspects of the architectural design) and standardised models (Common Criteria model, DMTF model, OMG model), and also oriented toward specific platforms based on Java technologies and the component model approach. The overall idea is to provide a generic abstract security model that reflects common particularities from several platforms in aspects related to security, that can be required at the same time in these platforms.


Security is a special problem that spreads over the full lifecycle

- The main actual force for maintenance
- The source for actual system families
- Technology is dynamic
- New attacks, new threats, new requests for solutions

We consider suitably covered functional aspects in software engineering, so functional aspects are not treated in this contribution.




UPM
Universidad Politécnica de Madrid
Teleinformática - FTSI Telecomunicaciones



Relevance & Benefits

- **Security system family lifecycle model and traceability**
 - Covers the whole development process (secure development)
 - Thinking about secure systems
 - Adapt traditional systems in order to obtain secure systems
- **Scope**
 - Security requirement analysis, security meta-models
 - Security design, security transformations
 - Security implementations, security aspects
 - Security testing, security checking tools
 - Security operation and maintenance
- **Integration in the system family engineering framework**
- **Integration of approaches, methods, models, etc**
- **Focusing on main problem: SECURITY IS THE MAIN DRIVER FOR CHANGE/EVOLUTION NOWADAYS**



FAMILIES Task 3.2 CWD
Execution Qualities
© Telvent; Miguel A. Oltra, © UPM; J.L. Arciniegas

17/10/2005
Eureka Σ! 2023 Programme, ITEA project ip02009, FAMILIES

The security is relevant over full lifecycle, we try to cover all development process in order to obtain secure system families. The full lifecycle will cover the following development phases:

1. Requirement analysis: to provide a set of security requirements templates, that could be later made specific for product requirements. This set of general requirements is obtained as a result of the domain analysis performed in the security area.

2. Design: to provide a set of security architectural models, by means of UML profiles for security (in the same sense that Quality of Service profiles are available). Several sources will be used for this purpose (OMG, DMTF, IETF, W3C, for example). Guidelines for architectural assessment from the security viewpoint will be given.


3. Implementation: to provide several design and implementation mechanisms in order to perform the security architectural models, and to compare them. Among the mechanisms we foresee are: the usage of security-aware component models (a comparison of component models with respect to security is mandatory), the usage of “aspects” models and technology, the usage of language specific mechanisms (e.g. JAAS), the usage of specific servers (e.g. AAA servers). The impact of each mechanisms with respect to other quality attributes of the systems will also be studied.

4. Testing: to provide security-aware testing mechanisms based on the usage of threat scenarios for testing products and product families. Thus, the discovering of new security attacks can be aligned with regression testing.


5. Operation and maintenance: to provide mechanisms for the security supervision for deployable systems, and a roadmap for security maintenance. In fact, we understand that security is nowadays the main force for corrective maintenance.

We try to obtain a integrated system family engineering framework. It will cover the full development lifecycle with a specific focus in “security aspects”.

We consider the security as a new challenge for software evolution. Users demand more security (personal or corporative data, risks for possible attacks, virus, etc). The companies should offer more “reliable” services. Designers, architects, developers should change their practices in order to obtain secure systems. We consider the security as, the main driver for system change and evolution.




UPM
Universidad Politécnica de Madrid
Teleinformática - FTI Telecomunicaciones



Approach & Description of Results

- **Security activities during the SF lifecycle**
- **State of the art**
 - Contribution analysis
 - Technologies and how they are used
 - Current standards
- **Provide previously created elements in each of the lifecycle assets**
- **Identify the variation points in there**
- **Validation**
 - Scenarios (accounting, availability, trustworthiness and integrity)
 - Demonstrators and case studies
- **Complete security package**

- **Others**
 - Service Oriented Architecture
 - Practical usage of MDA
 - Identification of MDA transformation techniques
 - Practical usage of Aspect Oriented Programming
 - ... and the link between them
 - Used in different scenarios



FAMILIES Task 3.2 CWD
Execution Qualities
© Telvent; Miguel A. Oltra, © UPM; J.L. Arciniegas

17/10/2005
Eureka Σ! 2023 Programme, ITEA project ip02009, FAMILIES

Activities with respect to security during the SF lifecycle

1. Requirement analysis, security meta-models
2. Design, security transformations
3. Implementations, security aspects
4. Testing, security checking tools
5. Operation and maintenance

State of the art

We consider the most important sources, based in standard information and serious organizations, such as OMG, W3C, IETF, DMTF and other work in system security. A brief summary of security by each organization and other approaches of specific tendencies such as OSGi, WebServices or CERT will be presented. And then will be shown a set of security tools frequently used in the world.

But currently there exist several technologies. We will reuse them and will provide already created elements in each of the lifecycle assets


Obviously, solutions have common points, but also several ways to solve the same security problem. From our point of view we will try to identify variations points and similarities.


For validation we will consider real scenarios (accounting, availability, trustworthiness and integrity), where alternative solutions will be checked. Security features, such as secure deployment, secure remote management, secure access, protection in user data, etc, will be dealt with.

Others

The security, as other quality attributes, affects to several elements of a system. Therefore to increase the security, several components should be improved. An alternative (maybe the best solution) is the Service Oriented Architecture (SOA). The SOA isolates parts of the system into services. The SOA is an architectural style, whose goal is to achieve loose coupling among service providers and service consumers. A service provides a functionality that is well-defined, self-contained, and does not depend on the context or state of other services. Other alternative would be to consider Aspect Oriented Programming (AOP), in this case, spread characteristics (security) are encapsulated in aspects. Theoretically, we could modify or improve an aspect by means of maintaining system functionalities. However, AOP is a immature technology, but can be an important one in the future.


The security involves full lifecycle, then support for transformations among models would be needed, so it would imply the practical usage of MDA, SOA or AOP.


Universidad Politécnica de Madrid
Teleinformática - FTI Telecomunicaciones



Results

- **Security activities during the SF lifecycle**
 - Security requirement analysis
 - Identification and authentication.
 - Authorization and access control.
 - Security auditing to make users accountable for their security related actions.
 - Security of communication.
 - Non-repudiation and cryptography
 - Administration of security information.
 - Security design
 - Security framework for system family
 - Security implementations
 - Alternative solutions from W3C, IETF, DMTF, OMG, WS, others
 - Security testing
 - Tools
 - Security operation and maintenance
 - Supported in the security framework


FAMILIES Task 3.2 CWD
Execution Qualities
© Telvent; Miguel A. Oltra, © UPM; J.L. Arciniegas

17/10/2005
Eureka Σ! 2023 Programme, ITEA project ip02009, FAMILIES

Security requirement analysis:

The set of requirements are related with the security aspects: identification, authentication, authorisation, access control. Security auditing to make users accountable for their security related actions, security considerations during communications, non-repudiation and cryptography of messages and packages and security information administration.

Security design:

In this phase a Security framework is proposed for system families that should be a reference to achieve secure systems. This framework puts together a set of standards and technologies frequently used.

Security implementations:

Alternative solutions from W3C, IETF, DMTF, OMG, WS and other sources have been considered.


Security testing:

Tools with capabilities for model traceability and transformation will be used to check both models and scenarios.


Security operation and maintenance:

The framework is designed thinking also in configuration and maintenance operation during run-time.






UPM
Universidad Politécnica de Madrid
Teleinformática - FTSI Telecomunicaciones



Results

- State of the art

| | |
|-------------------------|----------------------------------------------------------------------------------------------------------|
| Security in DMTF | CIM - UML Core Specification |
| Security in OMG | Security Service Specification |
| Security in WS | Web Services Security |
| Security in W3C | Digital Signatures, HTTP/1.1 protocol, eCommerce and Security in web services |
| Security in IETF | Intrusion Detection Exchange Format, Extended Incident Handling, IP Security Protocol, Kerberos WG, etc. |
| Security in OSGi | OSGi 3.0 Specification |
| Security in Java | Java 2 SDK, v 1.4 Security Documentation |
| Others... | Security supported in AOP |

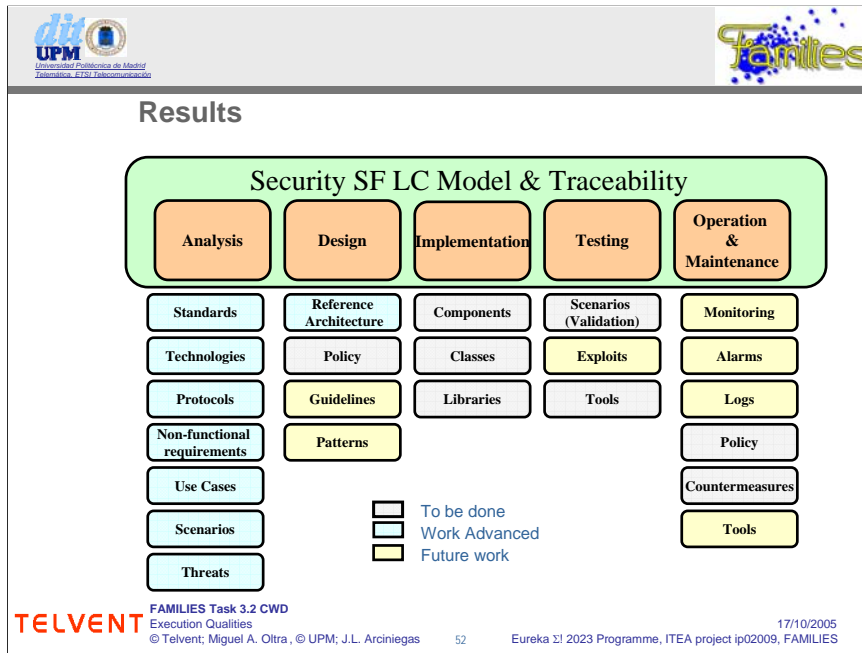


FAMILIES Task 3.2 CWD
Execution Qualities
© Telvent; Miguel A. Oltra, © UPM; J.L. Arciniegas

17/10/2005
Eureka Σ! 2023 Programme, ITEA project ip02009, FAMILIES

Our source has been in first place standards most used such as DMTF (CIM - UML Core Specification), OMG (Security Service Specification), WS (Web Services Security).

- W3C (Network security, authentication services, message validation, personal privacy issues, cryptography, Digital Signatures, HTTP/1.1 protocol, eCommerce and Security in web services)
- IETF (Intrusion Detection Exchange Format, Extended Incident Handling, IP Security Protocol, Kerberos WG, Public-Key Infrastructure (X.509), Securely Available Credentials, Secure Shell, Secure Network Time Protocol, Transport Layer Security, XML Digital Signatures, etc.)
- OSGi (OSGi 3.0 Specification)
- Security in Java (Java 2 SDK, v 1.4 Security Documentation)
- Security supported in AOP
- and a big amount of papers and books related with security aspects.



In the figure is represented the mapping of the advanced work so far and the work to be done both from Telvent and DIT-UPM jointly.

So far, in blue background boxes are indicated the advanced work. Mainly, analysis and design phases of the security life cycle model has been achieved.

- Analysis activity: security standards have been analysed, technologies, protocols and non-functional requirements have been identified to deal with the domain oriented solution.
- Design activity: a security reference architecture model has been defined, some use cases have been identified, and an scenario focused on deployment has been proposed.

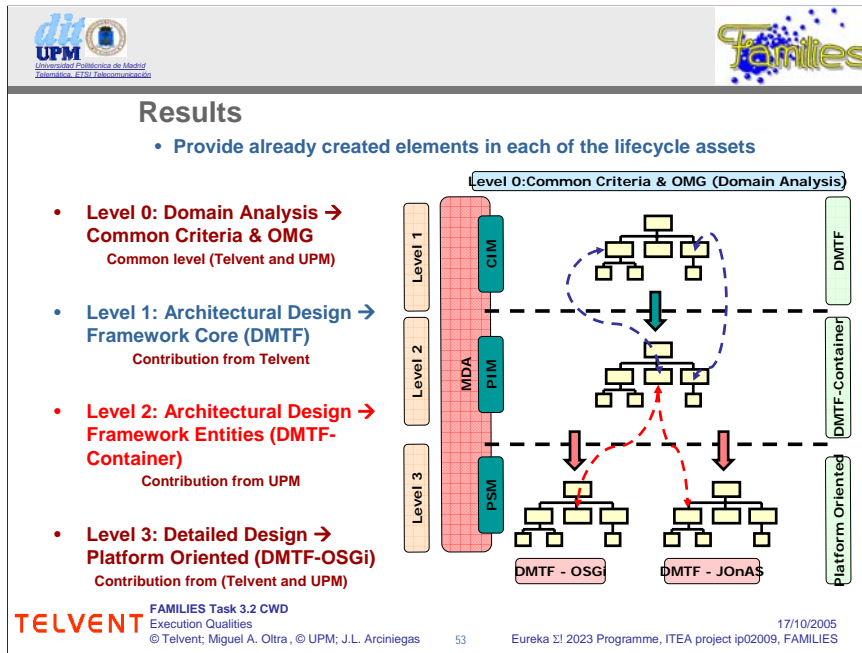
The work to be done in order to finalise with the security life cycle phases are indicated with grey background colour in the figure. The work to be done on these activities is:

- Design: definition of the SF security policy
- Implementation activity: components, classes and required libraries
- Testing activity: validation of the proposed scenario on a demonstrator, and tools for validating the goodness of the designed security reference model
- Operation and maintenance activity: definition of the security policy for the operation and maintenance of the family and countermeasures required to achieve required security quality aspects in the whole family

In yellow colour are presented other activities that should be part of the security SF LC, they will be treated in future works.

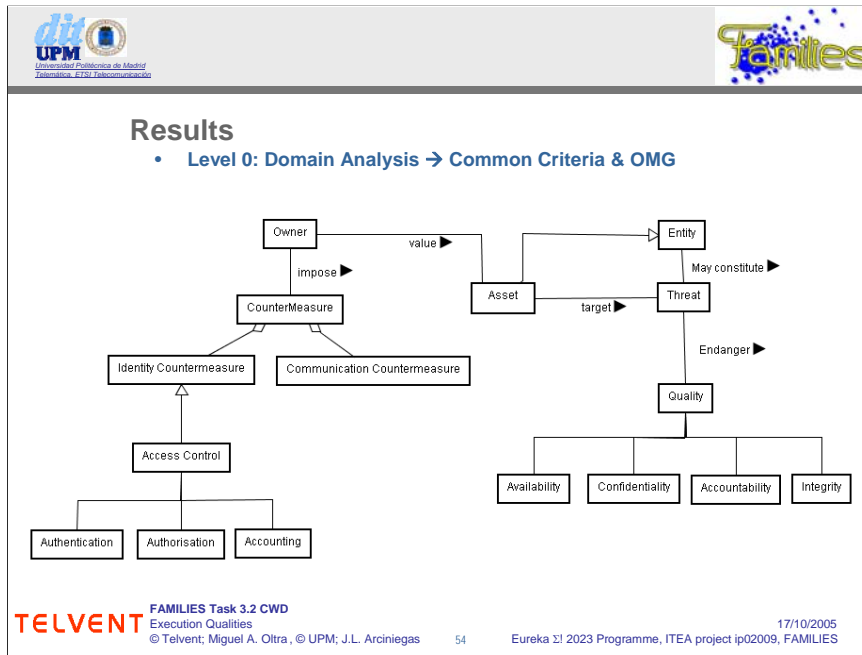
Security SF LC Model and traceability must be considered as a generic model, any platform can be chosen for its implementation.

The development process is not necessarily a sequence of activities as the "water fall" model. This model allows the movement in both directions (traceability).



The followed strategy to define the security reference architecture model is shown in the figure. The Platform oriented model is achieved by means of the definition of another models (DMTF-Container and DMTF models). Those models can be seen as the CIM (Computation Independent Model) and the PIM (Platform Independent Model) in a MDA approach. While, the platform oriented model can be seen as the PSM (Platform Specific Model) on a MDA approach. Four levels has been identified in order to cover both domain analysis and design.

- Level 0: Domain Analysis → Common Criteria & OMG
- Level 1: Architectural Design → Framework Core (DMTF)
- Level 2: Architectural Design → Framework Entities (DMTF-Container)
- Level 3: Detailed Design → Platform Oriented (DMTF-OSGi)



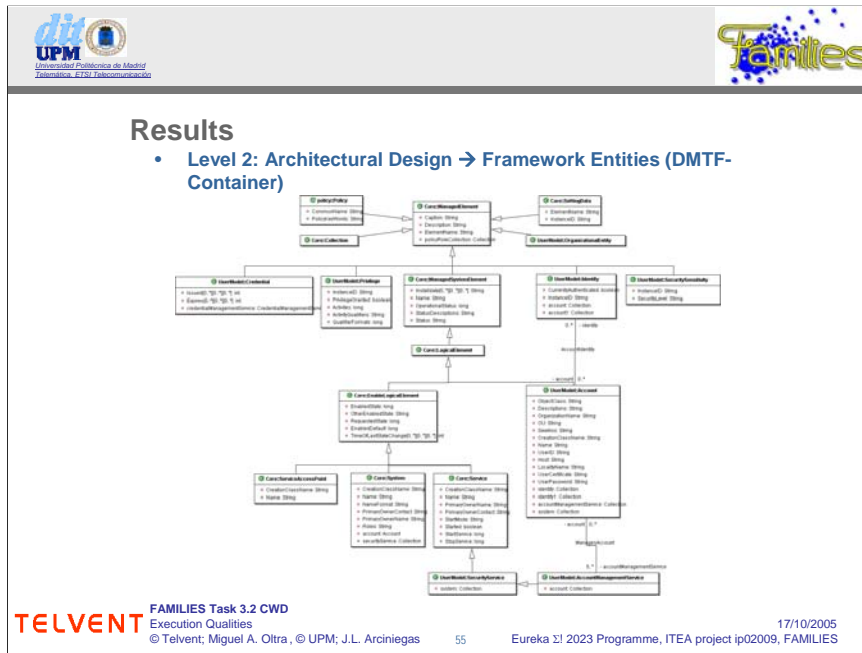
The Figure seeks to illustrate that it is the endangering of security quality aspects such as confidentiality that leads to the imposition of security countermeasures such as authentication .

In addition, the Figure shows the relationship between the main objects visible in different views for three types of security countermeasures.

1. Authentication of principals and security associations (which includes authentication between clients and targets) and message protection.
2. Authorization and access control (i.e., the principal being authorized to have privileges or capabilities and control of access to objects).
3. Accountability -- auditing of security-related events and using non-repudiation to generate and check evidence of actions.

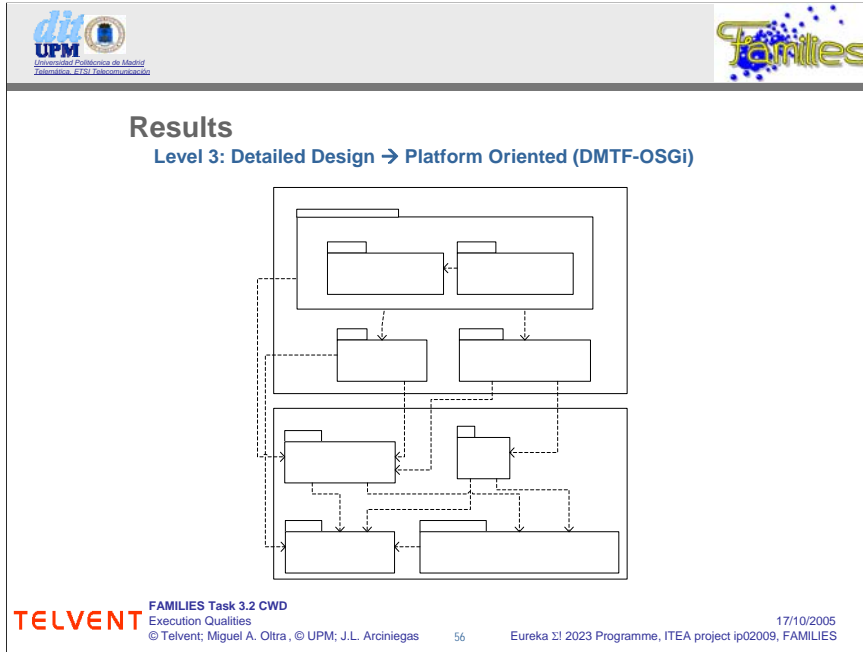
The OMG has considered as relevant quality aspects to security the following ones:

1. Confidentiality - Information is disclosed only to users authorized to access it.
2. Integrity - Information is modified only by users who have the right to do so, and only in authorized ways. It is transferred only between intended users and in intended ways.
3. Accountability - Users are accountable for their security-relevant actions. A particular case of this is non-repudiation, where responsibility for an action cannot be denied.
4. Availability - Use of the system cannot be maliciously denied to authorized users.



Based in security framework core, specific security elements have been defined, an overview is shown in the Figure, the most important elements will be describes as follow:

- **Policy**, allows defining security policies that managed elements uses for their authentication.
- **Collection**, was defined in the security framework core
- **SettingData**, was defined in the security framework core
- **Credential**, is used to specify the type of credential used in some secure transaction, such as shared secret, kerberos ticket, Unsigned Public key, public key certificate, biometric credential, named shared IKE secret, and so on.
- **Privilege**, is used to give priorities to managed elements in order to obtain preferences between other managed elements in a specific situation.
- **SecurityService**, is used to define security services, authentication, authorization and accounting
- **Identity**, is used for authentication purposes.
- **SecuritySensitivity**, defines security quality, for example in order to obtain access levels
- **OrganizationEntity**, allows to define some involved roles in a secure scenario
- **Account**, is a persistent register of managed element, here its traces are stored



Security Agent (Based in DMTF)

Access Control Countermeasures

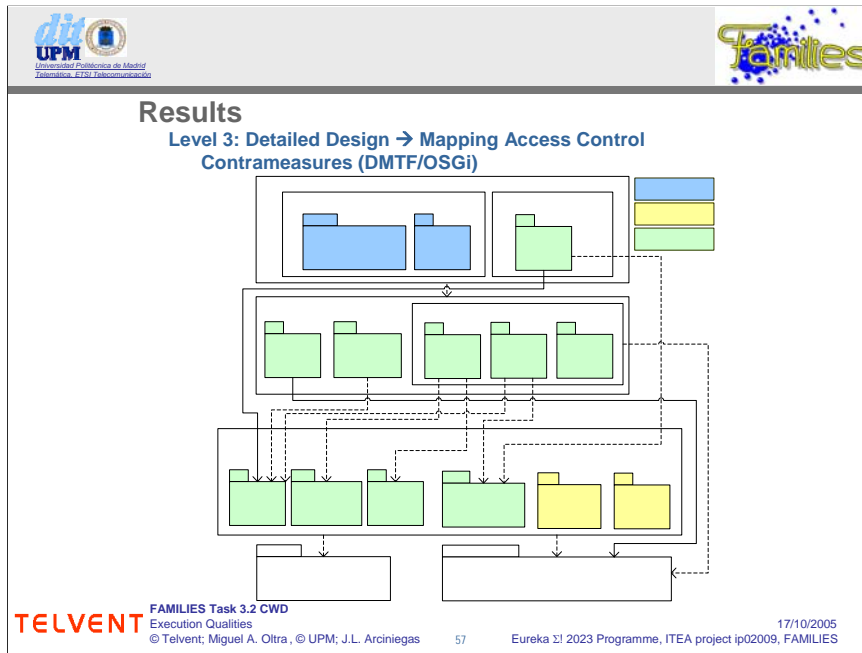
The Figure shows how security aspects are knitted together by security areas. Both authentication management and authorization management are grouped under Access Control Countermeasure. Access Control Countermeasure plays an important role because must guarantee the safe and trusted access to platform services and resources to which each allowed identity in the platform has the proper privileges. Communication deals with security aspects related with networking access from platform identities toward Internet and remote connections from Internet. Finally the Account Management deals with auditing information related with accessing, connections, permissions changes, etc, for taking possible countermeasures. Following these grouping is going to be treated in more detail.

Authentication Management

Communication Security

Moreover, the Figure indicates what are the dependencies among these grouped security aspects and also with java.security.Permission, OSGi Services, OSGi framework and OSGi util. A help for clarifying the indicated relationships in the Figure, could be given by this example. Suppose that an identity tries to change remotely a property of a service for which it has the proper permissions. Firstly network security must be applied to communication. Secondly access control countermeasure must be also applied. If the identity is validated and has the proper permissions then the property can be changed by the trusted identity.

java.security.Permission



Managed by Access Control Countermeasure Authentication Management

Certification Authority

Creden

Mapping process is a tedious labour, in this case qualities doesn't have an absolute standard and the CIM from DMTF was considered as the most general standard. As a consequence, the mapping process will be done taking into account only the security part defined in the CIM model.

Security Admin

DMTF co

Other difficulties are the different views of the security architecture (conceptual, static and dynamic views may be considered). In special dynamic conformance don't was included in this analysis because in this case we are comparing standards, not implementations, and the behaviour is not available (some standards are only recommendations).

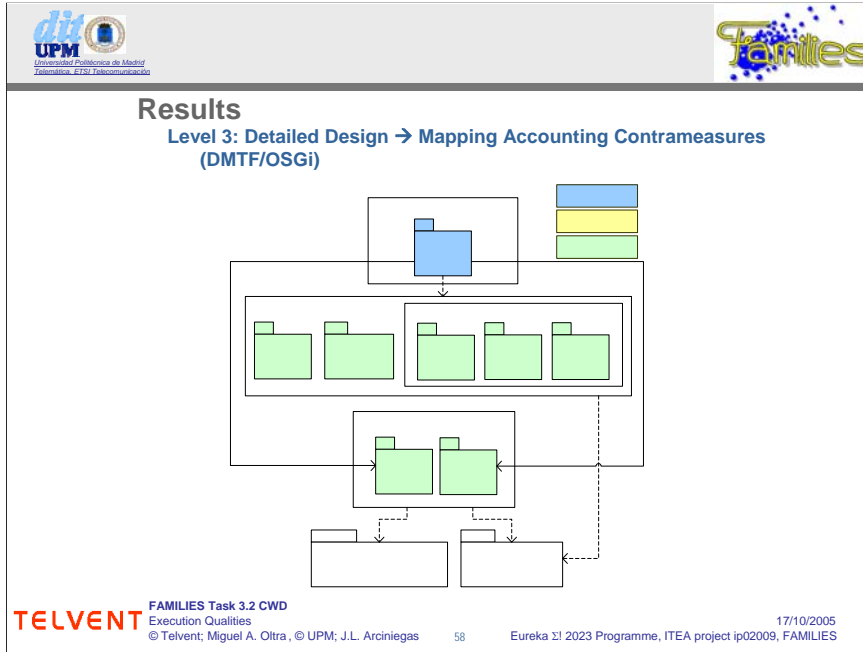
The next three slide sums up the main results of the mapping process:

- Extra-functionalities defined in security CIM are presented in blue colour, they are not supported in OSGi specification. In the real scenario these components could be required, a third one could support their functionalities, for example using Web Service Security (WSS).
- Extra-functionalities found in OSGi are illustrated in yellow colour, however, they are specific for OSGi context, OSGi is service oriented and these components allow to register and to manage services.
- Common components are represented in green colour. These components don't have an accurate equivalence, but after our analysis we have found clear similarities and commonalities.

Managed by Access Control Countermeasure

useradmin packageadmin device

java.security.Permission



OSGi specification doesn't define a accounting component (or service), only defines two packages (services in terms of OSGi) related with it "log" and "tracker", they are defined for general purposes, but can be used for accounting tasks/activities.

Several accounting functionalities are defined in the DMTF core (resources, policy and settings data). They should be assumed as functionalities by an OSGi based platform.

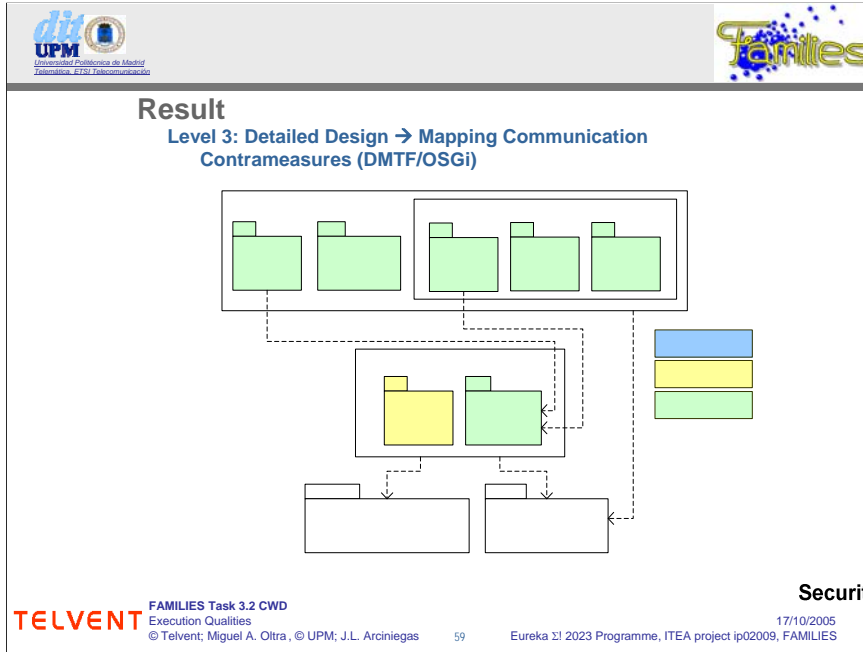
Other additional functionalities are required in OSGi domain with respect to accounting tasks, as result of this mapping new requirements to OSGi based platforms are presented.

Accounting Manag
 Accou
 Security Admin
 DMTF cor
 Identity Organisator Resou

Managed by Accountin

log

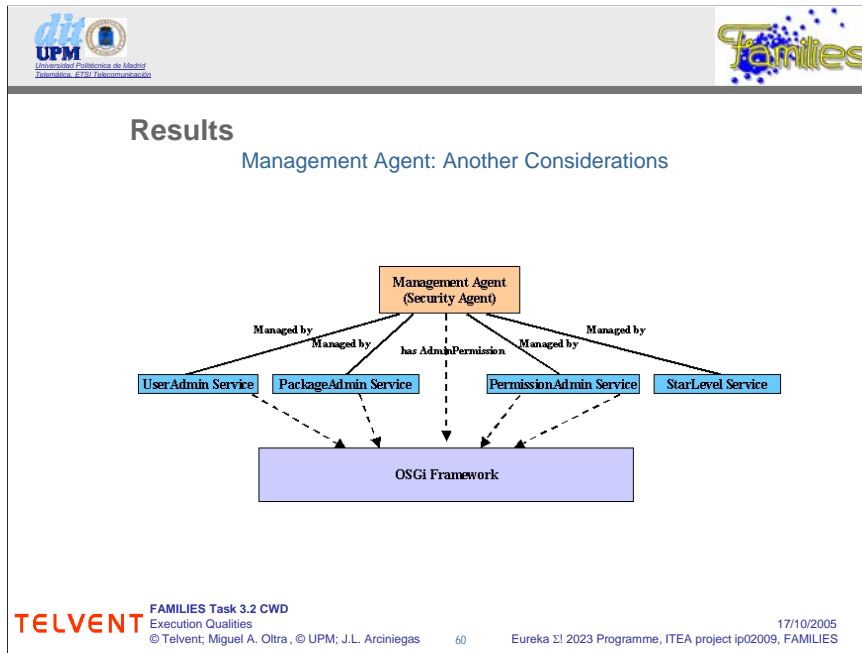
java.security.Permission



OSGi specification doesn't define a communication security component (service), Identity defines two packages (OSGi specified services) related with it "wireadmin" and "url"; they are for communication purposes, but they should be improved considering security aspects.

Several communication functionalities are defined in the DMTF core (resources, policy and settingdata), they should be assumed by OSGi based platforms, but like in java specification, security permissions must be supported by the platform.

Other additional functionalities are required in OSGi domain with respect to communication security tasks, as a result of this mapping new requirements to OSGi based platforms are presented.



Following are presented some security considerations extracted from the OSGi specification. The Figure represents how is delegated a set of security aspects to a Management Agent. This one has AdminPermission in the framework, so it is the only entity allowed to change/modify properties (realise the management aspects) and permissions in services defined by the specification or deployed in the OSGi based platform.

The Management Agent can perform the following actions over the permissions of a bundle:

- The Management Agent controls the permissions policy
- Get bundle permissions
- Set bundle permissions
- Update bundle permissions
- Delete bundle permissions

Results

- **Validation (Scenario of validation)**
 - A system manager deploys a new service component (bundle)
 - Deployment at runtime through internet connection.
 - Threats
 - Message spoofing, Identity supersede
 - Message sniffing
 - Platform damage
 - Exploit information from platform
 - Countermeasures
 - System Manager authentication
 - Validation of the integrity of the message
 - Admin privileges on the system to allow installation
 - Confidentiality of the message

TELVENT FAMILIES Task 3.2 CWD
Execution Qualities
© Telvent; Miguel A. Oltra, © UPM; J.L. Arciniegas

61

17/10/2005
Eureka Σ! 2023 Programme, ITEA project ip02009, FAMILIES

Validation scenario: Description

A system manager deploys a new service component (bundle) within the reference architecture of a remote platform (Service Gateway).

Environment, infrastructure/context

Component (bundle) deployment in a distributed managed system at runtime through internet connection. To provide confidentiality to the communications through Internet is required data encryption at application level. To provide authentication and message integrity, message signing is required.

Threats to security

- Message spoofing, Identity supersede

Spoofing definition: "Getting one computer on a network to pretend to have the identity of another computer, usually one with special access privileges, so as to obtain access to the other computers on the network". In this scenario, spoofing can appear, when someone tries to send a request message to the Service Gateway with the credentials of the System Manager, in order to achieve the authentication as the System Manager on the Service Gateway.

- Message sniffing

The System Manager credentials, can be obtained from message request sent through Internet. With these credentials, malicious attacks can be done against the Service Gateway, trying to supersede the System Manager identity.

- Platform damage

A deployment request message is sent to the Service Gateway, containing information for deploying a malicious component over it. The malicious component can be considered a Trojan Horse.

- Exploit information from platform

A malicious component deployed on the platform, can damage/change information stored on the Service Gateway. Also, information can be collected from the Service Platform.

Countermeasures necessary/useful

- System Manager authentication

A proof of data origin must be provided in the request message. This proof of data origin must include the credentials of the System Manager. This credentials are verified by means of the "Identity Access" functionality. This will allow to proof the identity of the System Manager, and in consequence its authentication on the Service Gateway is validated. The "Remote Access" service must obtain the credentials of the System Manager, and provide them to the "Identity Access".

- Validation of the integrity of the message

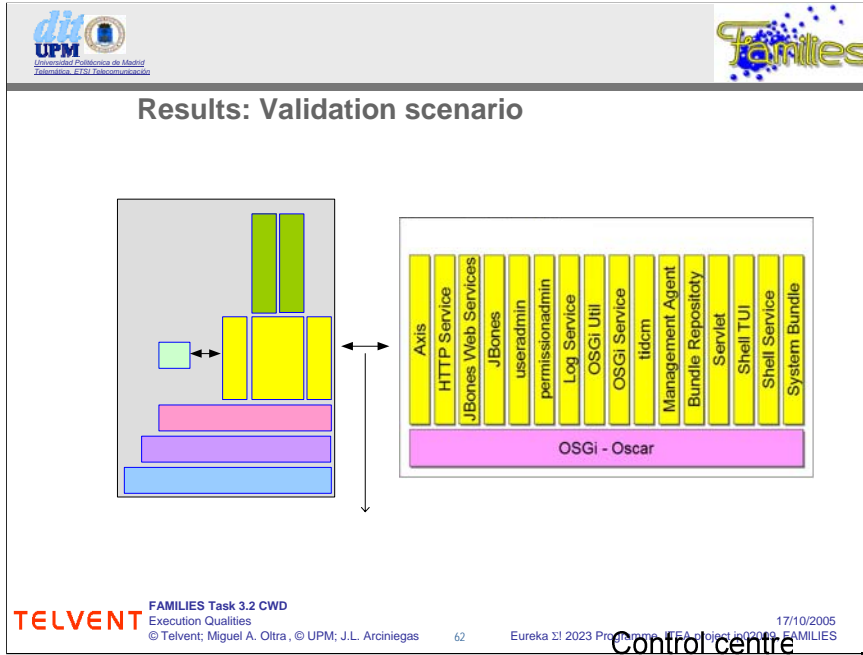
The integrity of the message must be guaranteed in order to avoid the identity supersede of the System Manager on request messages. The integrity of the message must be achieved by means of the inclusion of System Manager's signature and the inclusion of time stamp information in the request message sent to the Service Gateway. The "Message Integrity" must check that both signature and time stamp are valid both together.

- Admin privileges on the system to allow installation

The "Identity Access" must also check that the System Manager has the required privileges (permissions) for achieving the requested deployment service of the Service Gateway. The System Manager privileges are set on the "User Admin Service". The System Manager requires Admin Permission in order to deploy a component in the Service Gateway.

- Confidentiality of the message

The confidentiality of the message is provided by means of message encryption. The System Manager encrypts the request message with an encryption algorithm. The "Communication Encryption" service must de-encrypt the message. In order to achieve this, the Service Gateway must have the required information for de-encrypt the request message.



The scenario was implemented with OSGi support, but additional technologies should be used in the scenario for guaranteeing the different system security aspects. The following components are required:

a) Basic components from Oscar

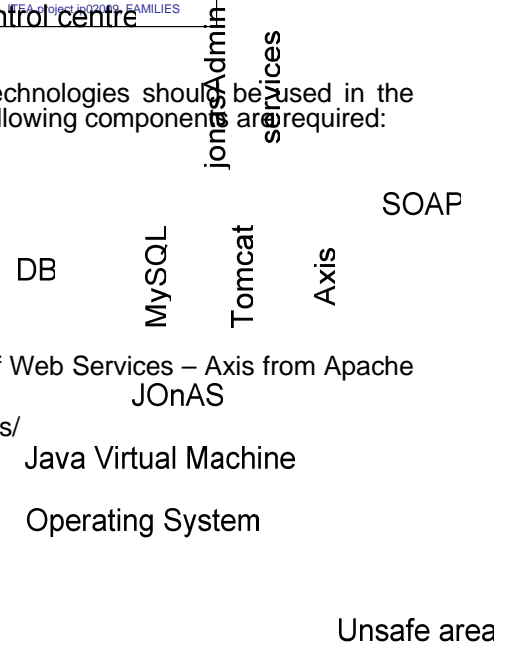
- PermissionAdmin
- UserAdmin


b) Basic infrastructure over Oscar:


- Axis Bundle (Web Services Support in OSGi), it is an adaptation of Web Services – Axis from Apache project: <http://ws.apache.org/axis/>
- JBones (Deployment Bundle): <http://forge.os4os.org/projects/jbones/>

c) Technology selected related to security:

- jonasAdmin and services
- MySQL database
- Tomcat
- Axis
- JOnAS
- JVM
- Operating system (Windows or Linux)
- Oscar (OSGi)
- JBones bundle
- SOAP as communication protocol





Universidad Politécnica de Madrid
Teleinformática #101 Telecomunicaciones



Results: Validation Scenario

- **Used tools: Nessus, NeWT Security Scanner and Retina network security scanner.**
- **Process:**
 - Step 1, test with Nessus to control centre and service middleware (ports 9080 and 80), as result some warnings were detected (2 warnings and 15 notes).
 - Step 2, test with NeWT security scanner to check navigation service and remote file access, as result a warning was detected
 - Step 3, test with Retina network security scanner to check the whole system, no additional security risk was detected
 - Step 4, solve possible risk detected with the above tools
- **Suggestions:**
 - Use XML encryption
 - Use XML digital sign
 - A security agent is required (see Telvent contribution)



FAMILIES Task 3.2 CWD
Execution Qualities
© Telvent; Miguel A. Oltra, © UPM; J.L. Arciniegas

17/10/2005
Eureka Σ! 2023 Programme, ITEA project ip02009, FAMILIES

Following are presented some reports results from network security scanner tools tests performed:

Nessus test

List of open ports:

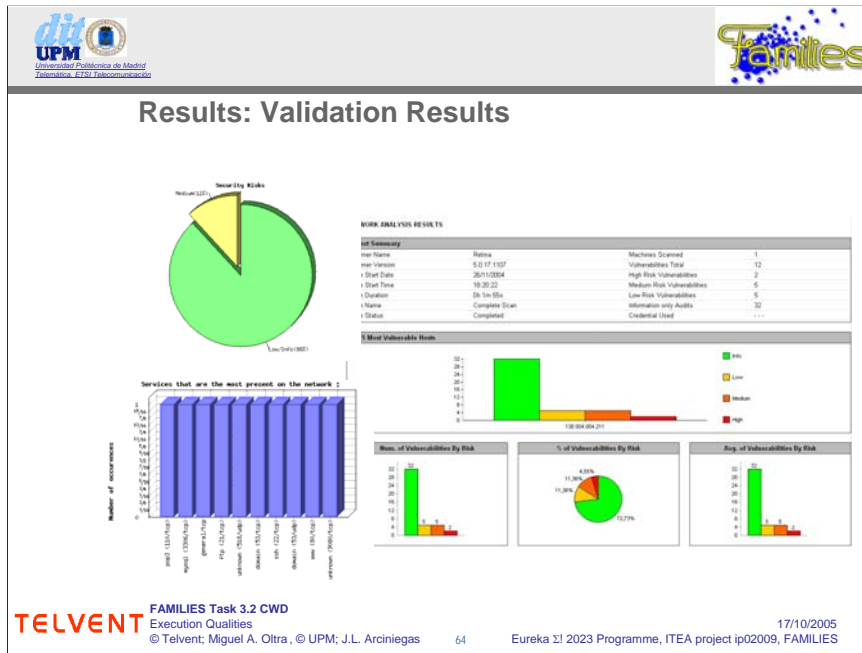
- unknown (9080/tcp)
- www (80/tcp) (Security notes found)
- domain (53/udp) (Security notes found)
- ssh (22/tcp) (Security warnings found)
- domain (53/tcp) (Security warnings found)
- unknown (518/udp) (Security notes found)
- ftp (21/tcp) (Security notes found)
- general/tcp (Security notes found)
- mysql (3306/tcp) (Security notes found)
- pop3 (110/tcp) (Security notes found)

NeWT security scanner test

14 Open Ports, 22 Notes, 3 Infos, 0 Holes.


Retina network security scanner test

20 vulnerabilities on the network




Some data as result of Nessus, NeWT Security Scanner and Retina network security scanner tools
 In the figures are shown percentage of vulnerabilities and risks, number of occurrences of services that are the most present in the network.

Although no critical vulnerabilities were detected, some warnings should be taken into account resolved, in the scenario, confiability is important and every possible vulnerability should be considered.




UPM
Universidad Politécnica de Madrid
Teleinformática #101 Telecomunicaciones



Conclusion & Outlook

- **The full lifecycle for security in services-applications was identified**
- **The most relevant standard organisations and other sources have been checked in order to obtain a whole state of the art**
- **Several created elements have been identified and located in respective lifecycle assets**
- **Some variation points have been identified**
- **A security model is presented**
- **The security model is based on MDA and AOD**
- **A basic scenario was implemented in order to validate the model**
 - Using some tools some vulnerabilities were detected
 - New requirements are suggested after validation process, the most important, a security agent is needed (it is been developed in Telvent Contribution)

TELVENT FAMILIES Task 3.2 CWD
Execution Qualities 17/10/2005
© Telvent; Miguel A. Oltra, © UPM; J.L. Arciniegas 65 Eureka Σ! 2023 Programme, ITEA project ip02009, FAMILIES



TELVENT

Shaping a World
of Convergence

Families Task 3.2 CWD

Chapter 3:
**A reference architecture for
security in system families**

**Tor Erlend Fægri, Svein Hallsteinsen, Jens
Glattetre, Ivar Sandstad**

Tor.E.Fegri@sintef.no, Svein.Hallsteinsen@sintef.no,
Jens.Glattetre@superoffice.no, Ivar.Sandstad@superoffice.no

ICT NORWAY

Task 3.2
Execution Qualities

Abstract: This document presents a reference architecture that supports the capture, use and maintenance of knowledge related to designing system family architectures that are faced with security requirements.

Keywords: Reference architecture, quality, security, architecture tactics, architecture patterns.

Relation to previous work in ESAPS & CAFÉ:

Café task 2.3 – design for quality.

Relation to other tasks in WP3 or other Work Packages of Families

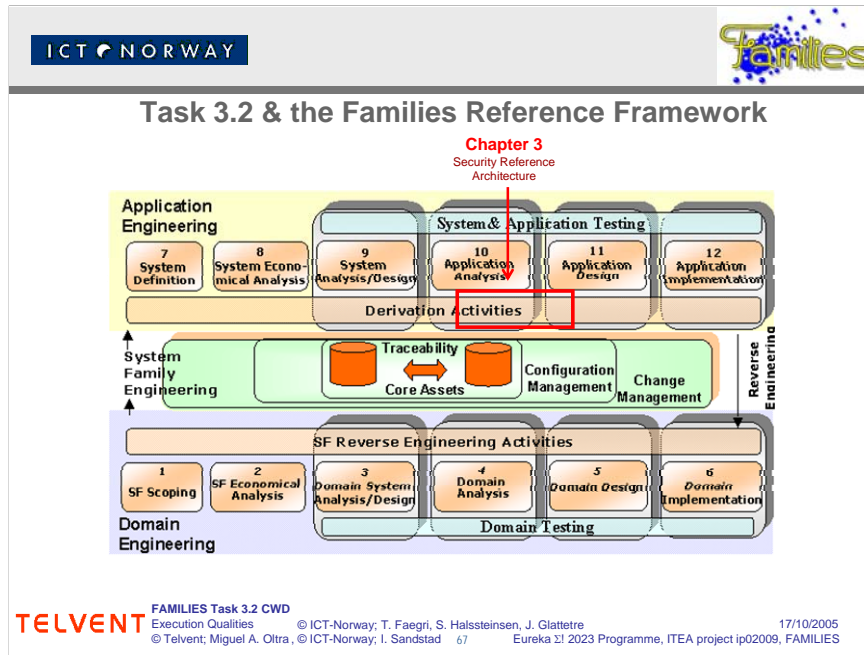
Task 3.2 Improving security quality (Philips Medical Systems)

Task 3.2 System family security (UPM)


Acronyms used in the contribution & Glossary of unusual Terms used: Reference architecture, decision model, quality model, architecture principle, architecture tactic, architecture pattern.

References: The work summarised is going to be published as part of the Families project research book (Editors: Timo Kakolo and Juan Carlos Dueñas). Preliminary title: Tor Erlend Fægri, Svein Hallsteinsen: "A reference architecture for security in system families".

Positioning of the work in the FEF: Supports capabilities required for level 3b in architecture dimension. Does not address business, process or organisational dimension.




- Chapter 3: Security Reference Architecture (ICT-Norway)
 Chapter 3 is focused on derivation activities at Application Engineering level.

ICT NORWAY


Introduction & Problem Description

- **Building software products that provide a suitable level of security is hard, but of critical importance for many companies:**
 - Motivated by continually increasing demand for open and flexible IT systems (“realtime enterprise”)
- **Security is a quality of a software system that bears on its ability to ensure correct access for its users:**
 - Confidentiality, integrity, availability and accountability.
- **The architecture has a great impact on the ability to provide the required level of security:**
 - Security must be designed into the system from the beginning through the use of appropriate architectural solutions
- **However, similar to other qualities, it must be balanced appropriately:**
 - Complicated by PFE
 - Security impacts most other qualities (e.g. performance, usability etc.)
- **Motivates scientific approach to security in PF Architectures**



FAMILIES Task 3.2 CWD

Execution Qualities

© Telvent; Miguel A. Oltra, © ICT-Norway; I. Sandstad

© ICT-Norway; T. Faegri, S. Halssteinsen, J. Glatteire


Eureka Σ! 2023 Programme, ITEA project ip02009, FAMILIES

17/10/2005

Increasingly, security requirements constitute a significant portion of the total set of requirements towards many software systems. Arguably, the most important aspect contributing to this trend is the seemingly continually growing demand for more open and flexible IT systems.

The architecture of a software system is important for the system’s ability to satisfy its requirements. That is, if the architecture is carefully designed the resulting system has a better chance of meeting the expectations. Constructing software systems of any significant size or complexity require considerations to the architecture. By architecture we mean its conceptual organisation in components, connectors and the relations between them.

The reference architecture presented here supports the architect in the SFE approach. We treat security requirements as a natural source of variability among the family members. In order to capture and manage knowledge related to security architecture design we propose a reference architecture for system family engineering. A reference architecture is in essence a knowledge repository with a structure to support architecture design.



ICT NORWAY

Relevance & Benefits

Relevance

- Arguably, many SW companies will adopt (or mature existing) PFE practices in the near future (2-4 years)
- As dependency on - and complexity of - SW systems increases, security is likely to remain a key challenge
- Few scientific efforts have been reported in the area

Benefits


- Guidelines, blueprints, decision models and architectural solutions will bring benefits to stakeholders in SW architectures for PF's having to deal with security
 - Improved SW quality
 - Best practices, increased confidence
 - Experience sharing

TELVENT FAMILIES Task 3.2 CWD
Execution Qualities © ICT-Norway; T. Faegri, S. Halssteinsen, J. Glattetre 17/10/2005
© Telvent; Miguel A. Oltra, © ICT-Norway; I. Sandstad 69 Eureka Σ! 2023 Programme, ITEA project ip02009, FAMILIES

For various reasons, it may be beneficial for an organisation to deliver multiple products with similar capabilities into its markets. We call such a set of products a system family. System families bring the additional challenge of managing variability among family members in a cost-efficient manner. The field of System Family Engineering (SFE) address these concerns, and has produced a large body of knowledge. The presented reference architecture builds upon these ideas while applying them in a security-focused setting.

To reduce cost, an organisation will typically strive to introduce a certain level of standardisation of the architecture between the family members. This can be at different levels, for example in terms of technical platforms, prescribed frameworks, general quality requirements or recommended architectural solutions. In order to accommodate standardisation, the product architect must first consider the implications of the already prescribed requirements and architecture solutions.


Already prescribed quality requirements must be reconciled with the product requirements. Architectural solutions identified as contributors towards the quality requirements of the product must be reviewed and aligned with already standardised solutions.

ICT NORWAY

Approach & description of results - Overview

Encoding security architectural knowledge in a reference architecture

- Objectives
 - Providing architectural guidelines for application architects dealing with security requirements
 - Verify if this approach is feasible
 - If so, verify if approach is useful
- Method
 - Quality model: Language for specifying security requirements
 - Architectural solutions: Set of architectural approaches (tactics → patterns) that promise to address security requirements
 - Decision model: Encoded architectural knowledge supporting the selection of architectural approaches that promise to meet requirements
- Approach
 - Developed in cooperation with SuperOffice
 - Based upon previous work related to adaptability




FAMILIES Task 3.2 CWD
Execution Qualities
© Telvent; Miguel A. Oltra, © ICT-Norway; I. Sandstad

© ICT-Norway; T. Faegri, S. Halssteinsen, J. Glattetre
Eureka Σ! 2023 Programme, ITEA project ip02009, FAMILIES

17/10/2005

Architecture design is a knowledge intensive art that heavily depends upon experience. In order to encode and reuse this knowledge, the software architecture community has created the concepts architecture tactics and architecture patterns. They are all documented, reusable architectural solutions that promise to address certain concerns in software architectures. They are, essentially, representations of knowledge of how particular problems can be solved.

ICT NORWAY


Approach & description of results – Reference architecture main components

- **Three main elements:**
- **1) Quality model**
 - Representing, reasoning about security requirements that may influence the architecture
 - Often conflicting requirements, but they are more easily resolved with the appropriate language and the right level of abstraction
- **2) A catalogue of architectural solutions addressing security**
 - A wide range of solutions, each promising to address security qualities
 - Includes effects on different qualities, not just security qualities
- **3) Decision support**
 - Assisting the design and evaluation of architectures
- **Target users:**
- **System and application architects. Architecture evaluators.**

TELVENT FAMILIES Task 3.2 CWD
 Execution Qualities © ICT-Norway; T. Faegri, S. Halssteinsen, J. Glatteire
 © Telvent; Miguel A. Oltra, © ICT-Norway; I. Sandstad 71 Eureka Σ! 2023 Programme, ITEA project ip02009, FAMILIES

17/10/2005

The reference architecture consists of three main parts:

1. A *security sub model* that supports the development of a risk assessment profile for the assets covered by the system. The risk assessment profile assists the architect in deciding what requirements should be set for the system and their internal priorities. The risk assessment profile is also helpful in the process of determining the most appropriate countermeasures;

2. An *architecture sub model* that incorporates architectural solutions which promise to address security related requirements;

3. A *decision support sub model* that supports capturing, specifying and reasoning about requirements for the system family members. Requirements are formulated as scenarios representing variation points. One scenario represents one variation point. A variation point will normally represent multiple variants. Now, in the presented reference architecture, not all of the scenarios contain multiple variants. In the development of the guidelines, we decided it was useful to capture this security architecture design knowledge despite the lack of direct variability aspects.

Together, these three sub models give the architect an integrated environment for architectural security design.

Approach & Results – Conceptual model

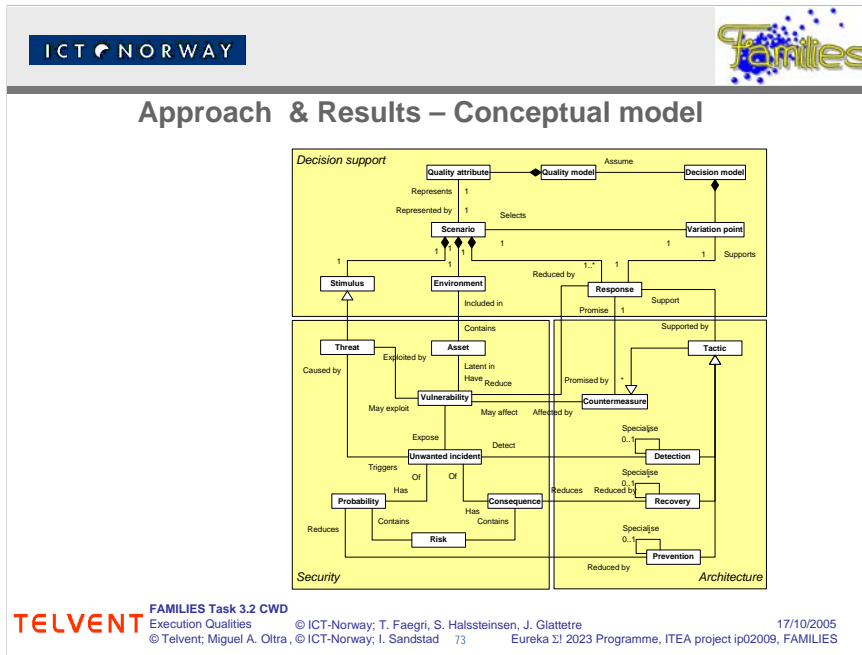
- **Three themes**

- Decision support
- Security
- Architecture


- **Developed in cooperation with companies**

- Initial scenarios developed together with central stakeholders in SuperOffice
- Scenarios were later refined, abstracted and extended with input from Telvent/UPM and other companies

- **Conceptual quality model draws upon previous work done in CAFÉ (*design for quality*)**

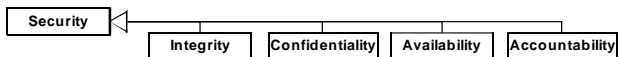


This conceptual model shows the three sub models with their core concepts and their inter relationships. The decomposition into three sub models also supports its extensibility. Also, our architectural solutions are organised in taxonomy of tactics and patterns. New tactics and patterns may be added to the existing ones in order to represent other architectural solutions. In practice, many organisations develop or refine their own architectural solutions. However, the adopting company must be able to associate effects on quality attributes from the architectural solution.


ICT NORWAY

Approach & Results – Quality model

- The architecture of a software system will to a large degree dictate the potential to achieve wanted qualities.
- This quality model enables *systematic reasoning* about security related qualities of software systems.
- The quality model captures key security related requirements to software systems in a concise and consistent manner.



```
graph TD; Security[Security] --- Integrity[Integrity]; Security --- Confidentiality[Confidentiality]; Security --- Availability[Availability]; Security --- Accountability[Accountability];
```




FAMILIES Task 3.2 CWD
Execution Qualities
© Telvent; Miguel A. Oltra, © ICT-Norway; I. Sandstad

© ICT-Norway; T. Faegri, S. Halssteinsen, J. Glattetre
17/10/2005
Eureka Σ! 2023 Programme, ITEA project ip02009, FAMILIES

Although useful at an overall level, the quality “security” from ISO 9126 it is too vague to be useful in requirements engineering. Further more, Jung et.al. shows that security as a sub characteristic from “functionality” is problematic [34]. By defining security using more concrete terms, this problem might be somewhat alleviated. To precisely capture and reason about security requirements, security is broken down into the four intermediate level security quality attributes *integrity*, *confidentiality*, *availability* and *accountability*.

Our experience is that this breakdown is very useful as it assists in the determination of security requirements. For example, during risk assessment, it helps improving common understanding among the participants. By considering each of the four generic security quality attributes in turn, one can more easily determine what quality properties are relevant or not for the particular application. As a trivial example, in the context of an application that serves public information via the Internet, confidentiality might be a low priority quality, but integrity is likely to be of high importance.

ICT NORWAY


Approach & Results – Quality model example

| Integrity quality | Decisions/ scenarios |
|------------------------------------------------------------------|----------------------|
| Withstand attacks within group of cooperating applications | <scenario links> |
| Secure use of COTS components | <scenario links> |
| Security in mobile systems, dynamic security boundaries | <scenario links> |
| Secure use of exposed communication infrastructures | <scenario links> |
| Security against unauthorised manipulation in user's application | <scenario links> |

TELVENT FAMILIES Task 3.2 CWD
 Execution Qualities © ICT-Norway; T. Faegri, S. Halssteinsen, J. Glattetre
 © Telvent; Miguel A. Oltra, © ICT-Norway; I. Sandstad 75

17/10/2005
 Eureka Σ! 2023 Programme, ITEA project ip02009, FAMILIES

For each business requirement there are multiple scenarios exemplifying the requirement. The scenarios are documented in the decision model.

Although we do not discuss the requirement specification process in more detail here, we assume that the application designer is supported by a risk assessment of the system. In security engineering, good requirements can only be made after assessing the risks pertaining to the assets encompassed by the system. Broadly speaking, risks are events that can jeopardize the security qualities. It is important that the risk assessment is done at a suitable level of detail so as to give a good understanding of which assets are worth protecting and the level of risk that each of these assets is exposed to. This creates the basic decision framework for the application designer when determining the security qualities that shall apply for the application.

ICT NORWAY

Approach & Results – Architectural solutions

- In the context of architecture, a tactic is a principle applied to achieve a certain effect, i.e. to achieve a quality requirement
- This is modeled in the reference architecture:

TELVENT FAMILIES Task 3.2 CWD

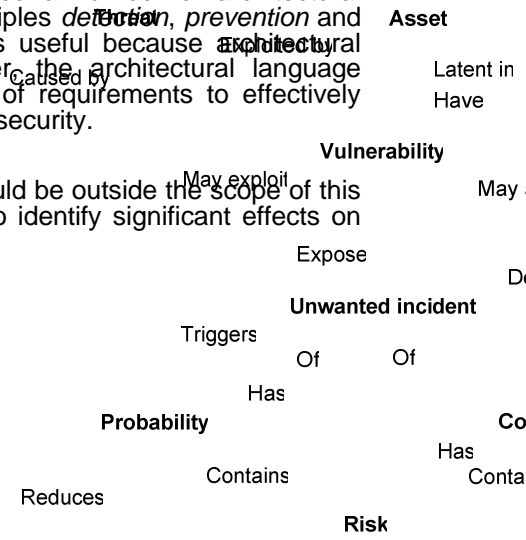
Execution Qualities © ICT-Norway; T. Faegri, S. Halssteinsen, J. Glattetre


© Telvent; Miguel A. Oltra, © ICT-Norway; I. Sandstad 76 Eureka Σ! 2023 Programme, ITEA project ip02009, FAMILIES

17/10/2005

One way to structure architectural solutions is according to the kind of principles they specialise or implement. In the context of security, we have identified and structured a number of architectural solutions. We have structured them according to the three high-level principles *detection*, *prevention* and *recovery*. We can not claim that this structure is the only one which is useful because architectural solutions will also address requirements other than security. However, the architectural language presented here enables application architects dealing with those kinds of requirements to effectively identify tactics and patterns that address particular requirements related to security.

Architectural principles are not described to a great level of detail. This would be outside the scope of this work. However, we provide references to further documentation. We also identify significant effects on quality attributes for each pattern.



ICT NORWAY


Approach & Results – Decision support ex.

Quality attribute: Confidentiality → Withstand attacks in a group of cooperating applications

Environment: Application a_c provides a set of services that makes sensitive information available to other collaborating applications over the Internet.

| Stimuli | Response | Resolution |
|--------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------|------------|
| An application a_m attempts to invoke services from application a_c without the required authorisations. | The architecture prevents a_m from accessing non-authorized services from a_c . | V. 1 |
| | The architecture allows a_m access to the services, but all accesses are logged. This facilitates recovery. | V. 2 |

Scenario resolution:

| Ref. | Approach | Architectural solution |
|------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------|
| V. 1 | The architecture requires that a_m is both authenticated and authorised before being allowed access to the services. | Prevention. Access control. (Component) authentication. Authorisation. |
| V. 2 | The architecture acknowledges that availability of information may be more critical than preventing access to it. However, by logging all accesses to the information, liability is put on the application a_m . | Recovery. Liability transfer. Digital certificates. Auditing. |

TELVENT FAMILIES Task 3.2 CWD Execution Qualities
© ICT-Norway; T. Faegri, S. Halssteinsen, J. Glatteire
17/10/2005

© Telvent; Miguel A. Oltra, © ICT-Norway; I. Sandstad 77
Eureka Σ! 2023 Programme, ITEA project ip02009, FAMILIES

This is an example scenario, representing a certain aspect of confidentiality (maintaining confidentiality in an application integration setting). In this example, the scenario has two distinct responses – representing two alternative ways to affect the quality attribute. One is to prevent the incident from occurring, i.e. reduce its probability through access control. The other is to reduce the consequence of the incident by transferring liability. Each of the two variants is subsequently described in more detail in the scenario resolution part. The tactics and patterns that will help the architect in reaching the desired effect are also documented. Lastly, architectural decisions will normally have side effects (i.e. they may affect more than one quality attributes). In this scenario, this is documented in the column “other affected quality attributes”. **Quality attribute:** Confidentiality → Withstand attacks in a group of cooperating applications


Environment: Application a_c provides a set of services that makes sensitive information available to other collaborating applications over the Internet.

StimuliResponseResolution An application a_m attempts to invoke services from application a_c without the required authorisations. The architecture prevents a_m from accessing non-authorized services from a_c . **V. 1** The architecture allows a_m access to the services, but all accesses are logged. This facilitates recovery. **V. 2**

Scenario resolution:


Ref.ApproachArchitectural solution **V. 1** The architecture requires that a_m is both authenticated and authorised before being allowed access to the services. **Prevention. Access control. (Component) authentication. Authorisation.** **V. 2** The architecture acknowledges that availability of information may be more critical than preventing access to it. However, by logging all accesses to the information, liability is put on the application a_m . **Recovery. Liability transfer. Digital certificates. Auditing.**

The scenario is presented in two main parts. The main part, on the top, contains the environment, stimuli and response. The second part describes how the response can be accomplished. This scenario encompasses two different architectural decisions; V.1) is to reduce the probability of a security breach or V.2) is to reduce the consequences. The column “Tactics, patterns” refer to the architectural solutions one can employ in order to reach the response for the variation point. Finally, other affected quality attributes are indicated in the last column. We have included this information in order to show that architectural solutions may affect multiple quality attributes.

ICT NORWAY

Conclusion & Outlook

- **Representing architecture design knowledge in a reference architecture is useful in order to support the capture, use and maintenance of such knowledge**
 - Assists in clarifying threats, risks and incidents in the system
 - Assists in expressing security requirements
 - Assists in deciding upon architectural solution
 - The reference architecture provides important support in the strategic planning of how to deal with security for a system family
- **We plan to experiment with specializations of the reference architecture for specific problem domains in order to observe the effects of adoption in a company**
 - Reference architectures are most useful when the level of abstraction is appropriate for the company using it
 - This level must be determined in each case



FAMILIES Task 3.2 CWD
Execution Qualities
© Telvent; Miguel A. Oltra, © ICT-Norway; I. Sandstad

© ICT-Norway; T. Faegri, S. Halssteinsen, J. Glattetre
Eureka Σ! 2023 Programme, ITEA project ip02009, FAMILIES

17/10/2005
78

We have presented a reference architecture for security in system families. It has enabled us to show that it is feasible to provide useful decision support, based upon architecture design knowledge, for companies developing system families in which security is a quality requirement.

Software architecture design is still a bit of an art form. Although we can provide useful support for the stakeholders, there is a significant amount of manual labour involved in making the appropriate tradeoffs of the design alternatives and subsequently determining the final detailed design.

Reference architectures play two roles. One is to generalise and provide abstractions useful to a wide range of system. The second role is to be a platform from which specific architectures can be instantiated. The presented reference architecture has been developed with generality as primary objective. Our intention is that companies wanting to use it adopt it to the specific needs of the system families in development. As part of the future activities related to the project is to assist in the specialisation of the reference architecture for a specific system family.



TELVENT

Shaping a World of Convergence

Families Task 3.2 CWD

Chapter 4:
Security Issues in Dynamically Deployable SFs

Distributed Systems

Miguel Ángel Oltra
miguel.oltra@telvent.abengoa.com

TELVENT

Task 3.2
Execution Qualities

Abstract:

One of the most important non-functional requirements nowadays in software systems are those derived from security issues. Specially quality aspects related to security are being considered mandatory in distributed systems environments, where access, trusting and authentication concerns (among others) must be taken into account. New system architecture design patterns based on a component model will facilitate the dynamic and remote deployment of components during execution time; addressing security concerns for the whole architecture. A reference model based on standards that addressed security aspects in distributed systems will be covered in this contribution. The proposed security concerns are independent of the target platform. An scenario for model validation is also presented in the contribution.

Keywords: Security, Distributed Systems, Reference Architecture, Quality Models, Components, Architecture

Relation to previous work in ESAPS & CAFÉ:

•ESAPS:

- Task 1.3 Aspect analysis and modelling
- Task 2.2 Reference architecture
- Task 3.3 System family variant configuration and derivation

•CAFÉ:

- Task 2.3 Design for quality

Relation to other tasks in WP3 or other Work Packages of Families:

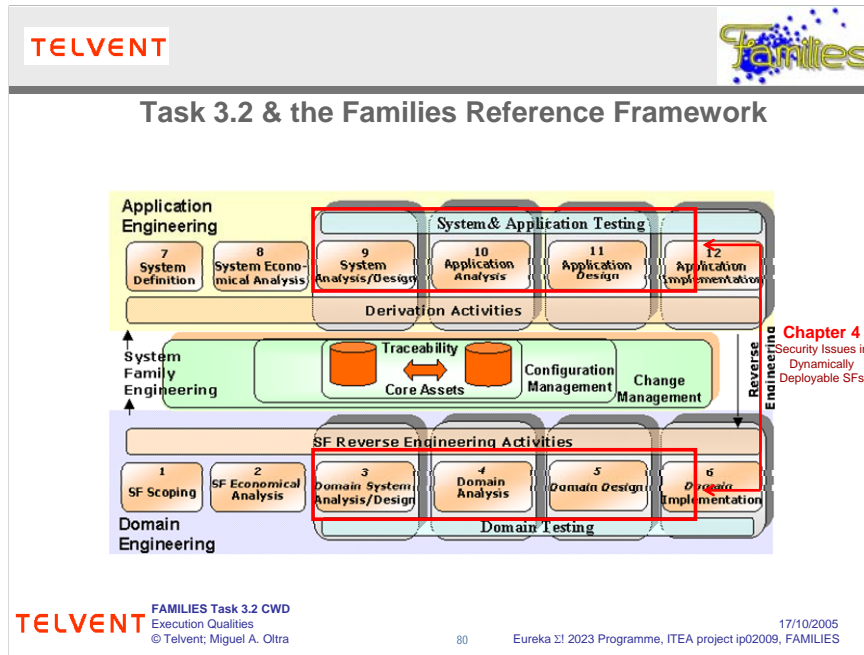
- Task 3.2 Execution Qualities: "System Family Security" (UPM)
- Task 4.1 Domain and application modelling practices and techniques
- Task 5.2 Process and organisation consequences of integration
- Task 5.3 Asset recovery for maintenance, manufacturing and supply

Acronyms used in the contribution & Glossary of unusual Terms used


- AOP: Aspect Oriented Programming
- DMTF: Distribute Management Task Force
- IETF: Internet Engineering Task Force
- OMG: Object Management Group
- OSGi: Open Services Gateway Initiative

References:

- [1] Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and general model, Version 2.1, Section 4. <http://www.commoncriteria.org>
- [2] OSGi Service Platform, Release 3, March 2003 (http://www.osgi.org/resources/spec_download.asp)
- [3] DMTF. Core Specification 2.9 (UML diagram). 2004. <http://www.dmtf.org>
- [4] DMTF. CIM User and Security Model White Paper. 2003
- [5] OMG, Security Service Specification Version 1.8. March 2002
- [6] Web Services and SOA; D.K. Barry Mk, 2004
- [7] M. Graff and K.van Wyk. Secure Coding, Principles and practices. O'reilly.USA. 2003
- [8] Technology Roadmap on Software Intensive Systems, The Vision of ITEA (SOFTEC Project); ITEA Office, March 2001
- [9] Whittaker, J. Why Secure Applications Are Difficult to Write IEEE Security and Privacy. 2003.
- [10] Aplicaciones utilizadas para el ejercicio de Potestades, Criterios de Seguridad; Ministerio de Administraciones Públicas; Febrero 2003
- [11] Controlled Access Protection Profile, Version 1.d, Information Systems Security Organisation; National Security Agency (NSA), 9800 Savage Road, Fort George G. Meade, MD 20755-6000, October 1999
- [12] The Java Security Architecture for JDK 1.2. Version 1.0, Sun Microsystems, October 1998. <http://java.sun.com/products/jdk/1.4/docs/guide/security/spec/securityspec.doc.html>
- [13] Linux Security Administrator's Guide, v0.98, 22 August 1998. <http://www.nic.com/~dave/SecurityAdminGuide/SecurityAdminGuide.html>
- [14] W3C, <http://www.w3.org/Security/>
- [15] Sang Shin. Secure Web services. JavaWorld. 2003
- [16] Apache, <http://www.apache.org>
- [17] Bouncy Castle, <http://www.bouncycastle.org/>
- [18] IETF, <http://www.ietf.org>
- [19] OASIS, <http://www.oasis-open.org>
- [20] JCP, Java Community Process, <http://www.jcp.org>
- [21] ISO/IEC. "Information Technology – Security Techniques – Entity Authentication Mechanisms; Part 1: General Model," Int. Org. Standardization, Genève, Switzerland, Tech. Rep. ISO/IEC 9798–1, 2nd ed., 1991.
- [22] Sovio, S. Asokan, N. and Nyberg, K. Defining Authorization Domains Using Virtual Devices. 2003
- [23] Pras, A. van Beijnum, B. Sprenkels, R. and Parhonyi, R. Internet Accounting. IEEE Communication magazine. May 2001.
- [24] IETF Working Group in AAA, Authentication, Authorization and Accounting. <http://www.ietf.org/html.charters/aaa-charter.html>
- [25] Tor Erlend Fægri, Svein Hallsteinsen. Memo Concerns Security reference model. 2003, SINTEF.



- Chapter 4: Security issues in dynamically deployable SFs (for distributed systems) (Telvent)
Chapter 4 is centred both in Domain and Application Engineering activities, centred in the whole activities, but mainly centred in analysis and design activities.

TELVENT


Introduction & Problem Description

- **Security over Internet must be guaranteed (it is not a safe environment)**
- **Nowadays Internet provides the interconnection of distributed systems (fixed or mobile platforms)**
 - Allowing the offer of services under demand by third parties (service providers)
 - Allowing the management of remote platforms by third parties (operator)
 - Allowing access to services by mobile users (global access)
- **Heterogeneous systems with common parts are considered**
- **Security is a mandatory quality attribute to be considered within the systems architecture**
 - Due to access implications some questions need to be solved (who is logged in, who is remotely managing the system, what is the origin of deployed components, ...)
 - How to guarantee/certify user/client access privileges & permissions has to be solved
- **Component model development allows**
 - Dynamic and remote deployment of components
 - Run-time deployment of components
- **Services are developed as software components by third parties (service providers)**
 - How to deploy these third party services has to be solved

TELVENT FAMILIES Task 3.2 CWD
 Execution Qualities
 © Telvent; Miguel A. Oltra

81


17/10/2005
 Eureka Σ! 2023 Programme, ITEA project ip02009, FAMILIES

Telvent is trying to tackle a security reference model as a solution for dealing with security aspects on a component model architecture for distributed systems. In a distributed environment based on interconnected systems through Internet a high number of considerations have to be taken into account during the analysis and design phases of the global system architecture. This contribution is focused on systems oriented to service platforms, where services are remotely offered by means of service providers. Services are being allowed on demand (e.g. final users, and also remote system operator, can deploy new services over a service platform).

During the design phase, heterogeneous distributed systems with common parts (the considered systems are basically similar platforms with different functionalities provided by deployed components) are going to be considered. Deployed components can be seen as services or as a subset of the functionality of a service (a service can be seen as the addition/composition of the functionalities provided by several components).

The use of a component model based architecture, implies that components can be dynamically and remotely deployed, and that the components can be deployed at run-time over the service platform. Tackling into account all possible threats that can happen in this such hostile context is quite difficult to be envisioned. To avoid the maximum number of threats that can potentially damage the integrity of the system, a well defined security reference model is required. The designed model must be a Domain Engineering oriented model (may be valid for several platforms, implemented with different technologies). The service platform can be seen as a common core, with a set of variants (components). Platform security attributes must be achieved by means of the proposed reference model. Due to this the reference model, has to be quite abstract enough to cover the whole possible threats that can happen on the environment in which the system is going to operate. Moreover, the security reference model, has to be flexible enough to be improved with new security aspects that can appear with the pass of the time (in order to avoid future threats).

The deployed components (providing functionality in terms of services allowed in the system) can be developed by third parties (service providers). How to deal with security quality attributes during deployment of components over a remote system in the previously described environment must be considered in the contribution. The origin of the deployable components must be guaranteed by means of mechanisms (e.g. authentication as proof of origin and integrity of request message), and also user privileges and permissions for the deployment of components need to be guaranteed among other security considerations.

TELVENT


Relevance & Benefits

- **Security design guidelines on System Families for component based distributed systems will be defined**
- **Distributed end-to-end systems security reference model definition (over an Internet infrastructure) for heterogeneous systems**
- **Deployment level security in distributed systems is considered**
 - Deployment level (e.g. Component certification, client certification, user authorisation, ...)
- **Among the expected benefits**
 - Platform independent security reference model
 - Identification and testing of possible threats for the family (that could happen in this context)
 - Identification of good and bad practices

TELVENT FAMILIES Task 3.2 CWD
 Execution Qualities
 © Telvent; Miguel A. Oltra

82

17/10/2005
 Eureka Σ! 2023 Programme, ITEA project ip02009, FAMILIES


The contribution presented in this chapter is focused on the following issues:

- The definition of a set of security design guidelines on System Families for component based distributed systems.
- The definition of a distributed end-to-end systems security reference model for heterogeneous systems. This heterogeneous systems will be interconnected by means of Internet.

The security aspects covered in the contribution are related to two levels of security in distributed systems. One of the considered levels is deployment level, where issues of the following nature should be considered: component certification, client certification, user authorisation, ...) and the other one is at operation level where issues of the following nature should be considered: supervision of deployed components, intrusion notification, client accounting, ... Security aspects will be validated on a proposed scenario.

Among the expected benefits of the work to be done in this contribution are:

- Definition of a platform independent security reference model based on a set of well known standards. In this reference model will be identified the set of security related quality attributes that a system must accomplish in order to guarantee the security for the family on operation
- Identification and testing of possible threats to the family (that could happen in the described context)
- Identification of good and bad practices that final users and system administrators must follow or avoid (depending on if it is a good or a bad practice) when they are working with the system

TELVENT


Approach & Description of Results

- **Study of state-of-the-art**
 - Identification of involved technologies (e.g. for certification, for encryption, for authentication, ...)
 - Identification of current security standards (e.g. CC, OMG, IETF, DMTF, ...)
- **Definition of a security reference architecture based on identified technologies and standards**
 - Platform independent
 - Technology independent for assuring security quality attributes
- **Identification and definition of security scenarios covering several security quality attributes (accounting, availability, confidentiality, integrity)**
 - Scenarios will include remote deployment of components
 - By system operator
 - By authorised user
- **Validation of defined scenarios over an end to end demonstrator**
 - Technology selection to comply with required security quality attributes
 - Definition of mechanisms for remote authentication and authorisation of involved entities

TELVENT FAMILIES Task 3.2 CWD
 Execution Qualities
 © Telvent; Miguel A. Oltra

83

17/10/2005
Eureka Σ! 2023 Programme, ITEA project ip02009, FAMILIES

In this slide is presented the approach being followed to define the previously mentioned security reference model. The first step followed on the approach was the realisation of a study of state-of-the-art with the identification of available technologies that could cover different security quality attributes. The final reference model must guarantee these quality attributes (e.g. for certification, for encryption, for authentication, ...) for the target environment (distributed systems over Internet connections). Also with the inclusion on it of guidelines/recommendations extracted from identified security standards from standardisation bodies that were used during the definition of the security reference model.

Following, the second step of the approach was the definition of a security reference architecture based on identified technologies and standards. The proposed reference architecture, will define a security model independent of the target platform. Also, this reference architecture will take on consideration a set of technologies in order to assure that security quality attributes are covered by the implemented architecture. In next slides will be presented the followed procedure to accomplish with the definition of the security reference model based on standards.

The third step was the identification and definition of some security scenarios covering different security quality attributes (accounting, availability, confidentiality, integrity). The description of this scenarios, includes the potential threats that may tamper the integrity of the system and the proposed countermeasures to guarantee the quality in terms of the security of the system. Moreover, the scenario description will include information about the environment in which potential threats can happen. Among scenarios that are going to be considered are the remote deployment of components, by the system operator (administrator) and by an authorised user with administrator permissions. Mechanisms for dealing with security aspects of the architecture have to be defined, and covered on the scenarios.

The fourth step of the approach is the validation of the proposed security reference model with the defined scenarios over an end to end demonstrator. At this phase of the approach, technologies for implementing the reference architecture and for validating the scenario has to be selected among possible candidates (identified during the study of-state-of-the-art) in order to comply with the required security attributes proposed in scenarios. Also mechanisms for dealing with authentication and authorisation of involved entities in the scenarios are going to be defined.



TELVENT

Approach & Description of Results:
State of the art: Overview

| | |
|-------------------------|----------------------------------------------------------------------------------------------------------|
| Security in DMTF | CIM - UML Core Specification |
| Security in OMG | Security Service Specification |
| Security in WS | Web Services Security |
| Security in W3C | Digital Signatures, HTTP/1.1 protocol, eCommerce and Security in web services |
| Security in IETF | Intrusion Detection Exchange Format, Extended Incident Handling, IP Security Protocol, Kerberos WG, etc. |
| Security in OSGi | OSGi 3.0 Specification |
| Security in Java | Java 2 SDK, v 1.4 Security Documentation |
| Others... | Security supported in AOP |

FAMILIES Task 3.2 CWD
Execution Qualities
© Telvent; Miguel A. Oltra

84

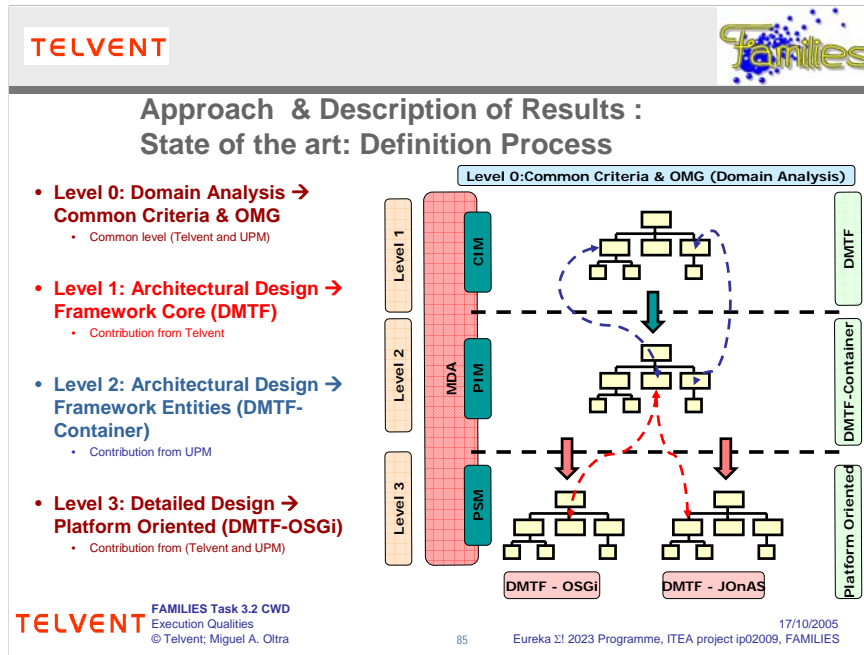
17/10/2005
Eureka Σ! 2023 Programme, ITEA project ip02009, FAMILIES

Our main sources has been in first place standards most used as CIM - UML Core Specification proposed by DMTF, the Security Service Specification proposed by the OMG, Common Criteria ISO/IEC 15408 and Web Services Security Specification proposed by OASIS.

Also other standardisation bodies documentation had been analysed:

- W3C (Network security, authentication services, message validation, personal privacy issues, cryptography, Digital Signatures, HTTP/1.1 protocol, eCommerce and Security in web services)
- IETF (Intrusion Detection Exchange Format, Extended Incident Handling, IP Security Protocol, Kerberos WG, Public-Key Infrastructure (X.509), Securely Available Credentials, Secure Shell, Secure Network Time Protocol, Transport Layer Security, XML Digital Signatures, etc.)
- OSGi (OSGi 3.0 Specification)
- Security in Java (Java 2 SDK, v 1.4 Security Documentation)
- Security supported in AOP and a big amount of paper and books related with aspects security

Another sources analysed are included within the references of this contribution.



In the figure, the followed strategy to define the security reference architecture model is presented. The presented followed approach has been jointly developed among Telvent and UPM (Universidad Politécnica de Madrid). The grade of contribution (per contributor) to each level is indicated on the slide with different colours and notes.

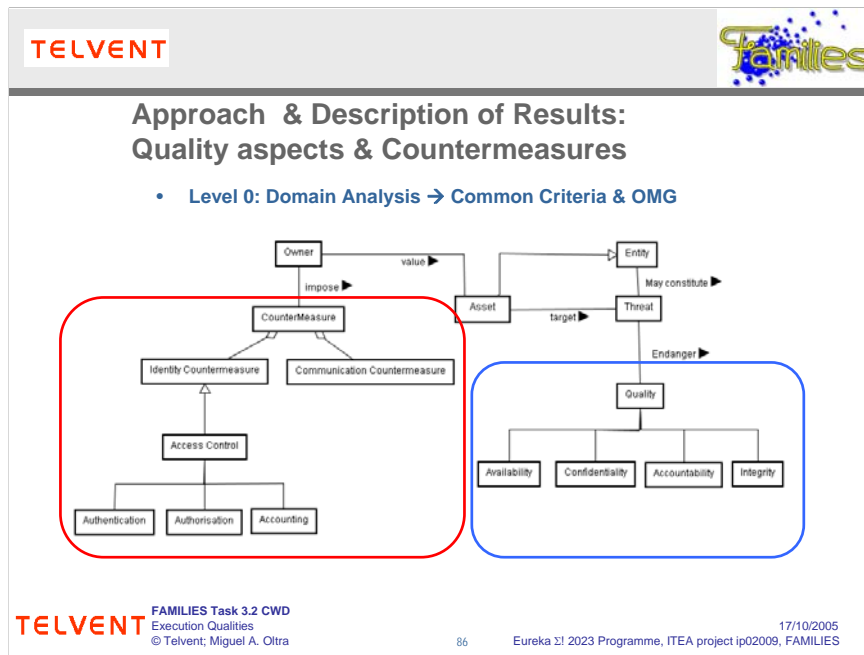
In this strategy, the platform oriented model is achieved by means of the definition of another intermediate models (DMTF-Container and DMTF models). Those models could be seen as the CIM (Computation Independent Model) and the PIM (Platform Independent Model) in an MDA approach. While, the platform oriented model could be seen as the PSM (Platform Specific Model) on a MDA approach. Four levels has been identified in order to cover both domain analysis and design. The idea is that the procedure for model definition follows the well known processes of a software development:

- The starting working point is a security domain analysis (Level 0). This domain analysis is based on the Common Criteria standard, by means of the inclusion of a domain security model based on the work done by Tor E. Faegri [SINTEF, 2003]. This model represents a high detail view of security concepts to keep in mind for designing security architectural aspects of a system.

- Later on an architectural design (Level 1) with the identification of the main framework entities based on inputs from platform oriented vision and DMTF model is provided. A detailed description of these entities in the following head of the document (Level 2) is required for clarification of the proposed model. The proposed model in this Level 1 can be considered an abstraction/particularization of the DMTF model by means of entities mappings. This abstraction model proposed at level 1 can be considered, using a MDA terminology as Computation Independent Model (CIM). The CIM is a Platform Independent Model (PIM) where the problem has not yet been worked out as a solution (as previously mentioned).

The different levels and mappings required for the design of this security framework are included in the figure. In this figure the approach being followed is presented. Due to the fact that the DMTF security model is wide, an intermediate security model is required (DMTF-Container). This one includes only parts of the DMTF model (by means of a mapping) and particular entities included in order to reflect the particularities of a container based system. The so called DMTF-Container model, could be seen using a MDA terminology as a PIM, from where the Platform Specific Model (PSM) should be derived.

-The level 3 of this document represents a mapping from the intermediate security model to a PSM. The considered specific platforms can be OSGi, J2EE (JOnAS), or whatever other specific platform, when a mapping has been done.



Level 0 deals with the domain analysis with regards to the security of a system. The proposed model is based on terminology identified from Common Criteria and the OMG specification previously mentioned. The domain represents what must be guaranteed on the system (quality attributes), and how these attributes are guaranteed (by means of countermeasures).

The figure seeks to illustrate that it is the endangering of qualities such as confidentiality that leads to the imposition of security countermeasures such as authentication.

In addition, the figure shows the relationship between the main objects visible in different views for types of security functionalities, that can be seen as countermeasures to impose in a system in order to guarantee specific quality aspects (see red box in the figure). Countermeasures has been grouped in two kinds:

- Those related with the identity that tries to access (or taken an offered functionality from the system): Identity countermeasures
- Those related with the communication among remote systems: Communication Countermeasures

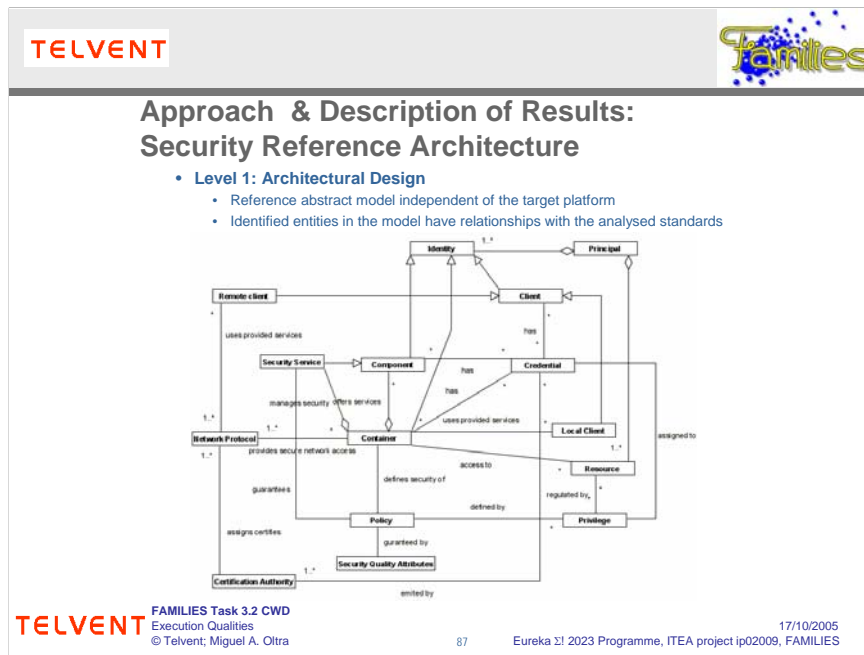
Identity countermeasures must guarantee the access control to certain resources in a system. Three types of security functionality deal with the access control to system resources:

- Authentication of principals and security associations (which includes authentication between clients and targets) and message protection.
- Authorization (i.e., the principal being authorized to have privileges or capabilities and control of access to objects).
- Accountability -- auditing of security-related events and using non-repudiation to generate and check evidence of actions.

The OMG has considered as relevant quality aspects to security the following ones (see blue box on the figure):

- Confidentiality - Information is disclosed only to users authorized to access it.
- Integrity - Information is modified only by users who have the right to do so, and only in authorized ways. It is transferred only between intended users and in intended ways.
- Accountability - Users are accountable for their security-relevant actions. A particular case of this is non-repudiation, where responsibility for an action cannot be denied.
- Availability - Use of the system cannot be maliciously denied to authorized users.

The set of countermeasures identified will be used in order to define the proposed security reference model, personalised on the identified functionalities of a Security Agent. Qualities represent the security aspects that must be guaranteed in the proposed scenario for validation purposes of the security reference model.



Level 1 deals with architectural design concerning to security. Previously to define a reference model for security, an identification of possible entities (in the scope of the target environment: Internet) related to security must be achieved. Also a description of these entities must be given in order to have a common framework of understanding (in other words, call the things with the same name or language).

In the figure are represented the relationships (in UML notation) among identified security entities in a distributed environment for a container oriented architecture.


Following a description of the identified relationships are given. Previously has to be mentioned that the model includes entities related with a distribute environment (e.g. Certification Authority). These entities has been included in order to obtain as most generic as possible model.

In the proposed model a Principal (Principal is a representation of a user of a managed system and its resources. These users may be human users, services, and groups thereof) is the aggregation of an Identity and a Resource. This Identity could be of three types (as reflected in the model): Client (remote or local, depending of how it accesses to the system services or resources), a Component or a Container. The Container could be considered as the central axis of the model. This one offers services by means of an aggregation of components that are increasing its provided functionalities to a final Client. In other words, a Container is a framework or a platform where components can be deployed on a predefined (standard) way.

Security aspects of the system are defined by means of a certain Policy. This Policy tries to guarantee non-functional aspects of the system architecture by means of guaranteeing its required Security Quality Attributes. This is done with the definition of a set of Privileges that are assigned to specific Credentials. Also the Privileges regulate the access to system Resources by Identities. Credentials are emitted by a Certification Authority in order to guarantee the trustworthiness of possible Identities (not all the Credentials must be emitted by a Certification Authority, another kind of credentials can be used, as login/password, shared secret, ...).

Network Protocols are included in the model in order to specify that remote access through network connections must be also secure.

The key entity in the model is the Security Service that will provide a set of functionalities for managing security aspects of the Container by guaranteeing the specified Policy of the whole system. This entity is the more important one within the proposed model, and its functionality and design will be later described in more detail.

TELVENT


Approach & Description of Results: Security Agent & Standards

- **Proposed security model is based on the personalisation of a set of standards**
 - Object Management Group (OMG) (Security Service Specification)
 - Common Criteria (ISO/IEC 15408)
 - Common Information Model / Distribute Management Task Force (CIM / DMTF)
- **The defined security model has a materialisation on the functionality identified in a Security Agent**
 - Required Security Agent functionality has been identified from analysed standards
- **The validation scenario will be based on security aspects to be covered on a specific platform based on OSGi (technology) specification**

TELVENT FAMILIES Task 3.2 CWD
Execution Qualities
© Telvent; Miguel A. Ojtra

88

17/10/2005
Eureka Σ! 2023 Programme, ITEA project ip02009, FAMILIES

In order to model the design a set of standards (as mentioned before) were considered. Following the main standards are detailed, including key concepts obtained from them and that have been included in the proposed model:

1. OMG (Security Service Specification):

This specification defines the term security in the following manner: “Security protects an information system from unauthorized attempts to access information or interfere with its operation.”

Security (as indicate in this specification) may affect to the system quality attributes in terms of:

- **Confidentiality:** Information is disclosed only to users authorized to access it.
- **Integrity:** Information is modified only by users who have the right to do so, and only in authorized ways. It is transferred only between intended users and in intended ways.
- **Accountability:** Users are accountable for their security-relevant actions. A particular case of this is non-repudiation, where responsibility for an action cannot be denied.
- **Availability:** Use of the system cannot be maliciously denied to authorized users.

2. Common Criteria (ISO/IEC 15408)

This specification defines the term security in the following way: “Security is concerned with the protection of assets from threats, where threats are categorised as the potential for abuse of protected assets. All categories of threats should be considered; but in the domain of security greater attention is given to those threats that are related to malicious or other human activities.”

3. CIM / DMTF:

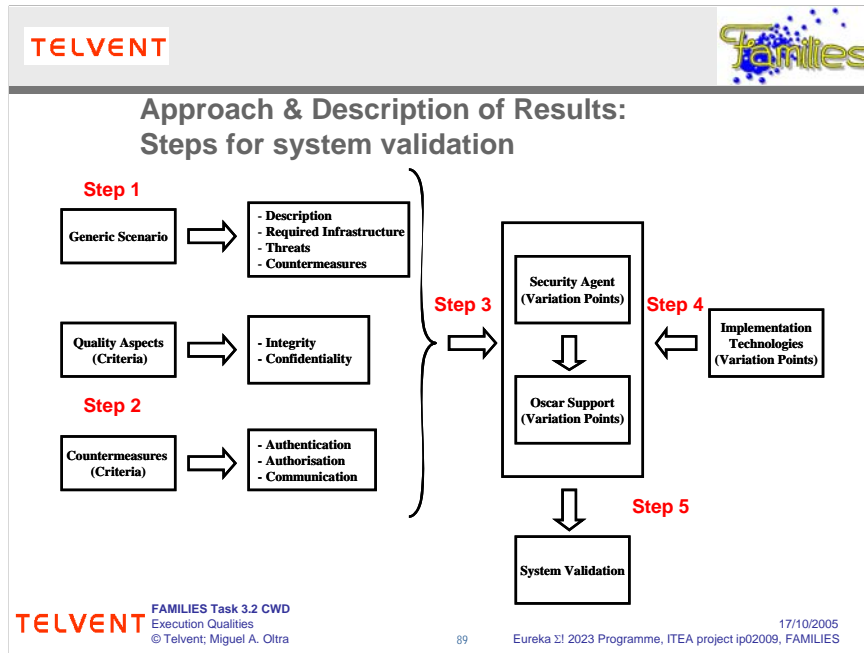
Defines a profile proposal for remote resources management. The CIM Security model is certainly not complete, but it does provide commonly needed classes from which vendor products may derive their specific information models. Future CIM work is expected to continue to expand on the foundation set of classes in this CIM Schema. This models has been complemented with ideas form CC (Common Criteria) and other sources (standards and technologies used).

The main conclusions that can be obtained from this profile are:

- Shows security aspects, supporting access to associated services, components and resources
- Model it not fully completed, but it is well developed enough, providing a set of classes from where specific models can be derived

From the set of profiles defined in CIM specification, has been analysed the user and security model, which main objective is to provide relationships among: User representations, Credentials, managed elements (resources representation) and also resources managers implied on the management of the user system.

Concepts obtained from these standards, has been materialised on the functionality identified in a Security Agent, where the functionality has been grouped in terms of countermeasures. The validation scenario will be based on security attributes to be covered on a specific platform based on OSGi (technology) specification (also these security attributes are concepts identified in standards).

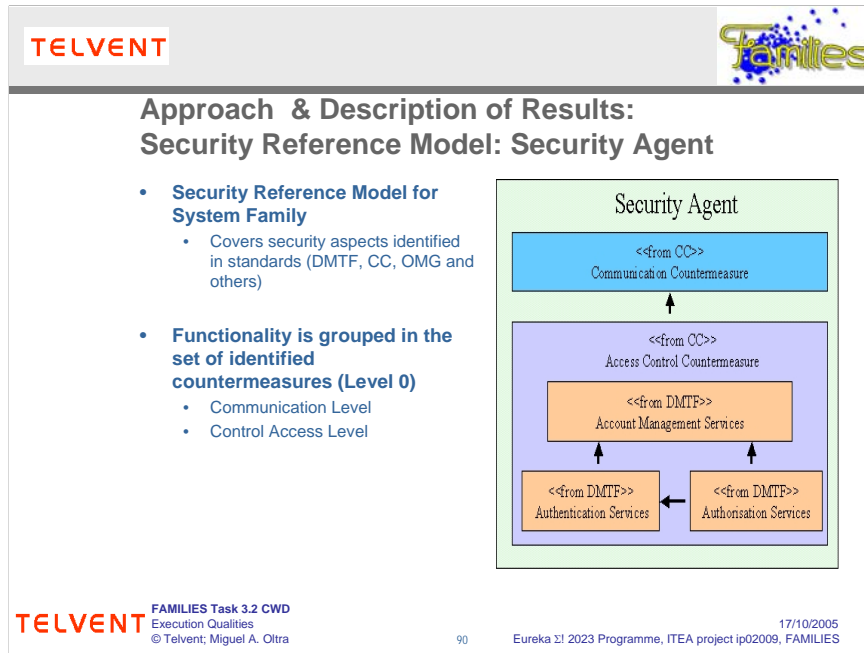


In the figure is presented an overview of the process being followed in order to validate the security reference model (proposed Security Agent). As the proposed model is quite general (and ambitious), a refinement of the scope of validation is required. In summary the figure, gathers the steps that should be followed in the validation process:

- Step 1. Generic scenario, in this step, a description of the scenario is given, joint with information related to: required infrastructure, threats that could happen, and countermeasures in order to avoid the possible threats.
- Step 2. Criteria must be established in order to reduce the scope of the scenario for validation. The proposed criteria are focused on selected quality aspects and countermeasures that must be validated within the scenario.
- Step 3. Inputs from previous steps indicates the variation points of the Security Agent and in consequence the support components of Oscar needed on the whole system for validation.
- Step 4. Also the implementation technologies represent variation points for each selected component (e.g. type of credentials: certificates, name, encryption protocol: RSA, DES, PGP, ..., and so on).
- Step 5. As a result of previous steps, a reduced scope for the system validation is indicated.

Among the possible variation points that can be considered in the distribute system are the following ones:

- Security aspects: Identification, authentication, authorisation, accounting, cryptography, ...
- Alternatives of solution: Standards, technologies, ad-doc solutions
- Levels of security
- Application domain: Telecommunications, domotic systems, internet applications, e-Business, e-Commerce, others...
- Hardware variations
- User variations



The figure represents a security reference model for System Family, that has been based in the proposal UML profile for CIM from DMTF and Common Criteria, where is defined concepts and mapped to UML diagrams (as mentioned before). Here is important to note, secure aspects shown, support access associated with services, components and resources. The figure represents services (that provide security functionality to the system) grouped in countermeasures. In the boxes it is indicated from which standard has been identified the different services.

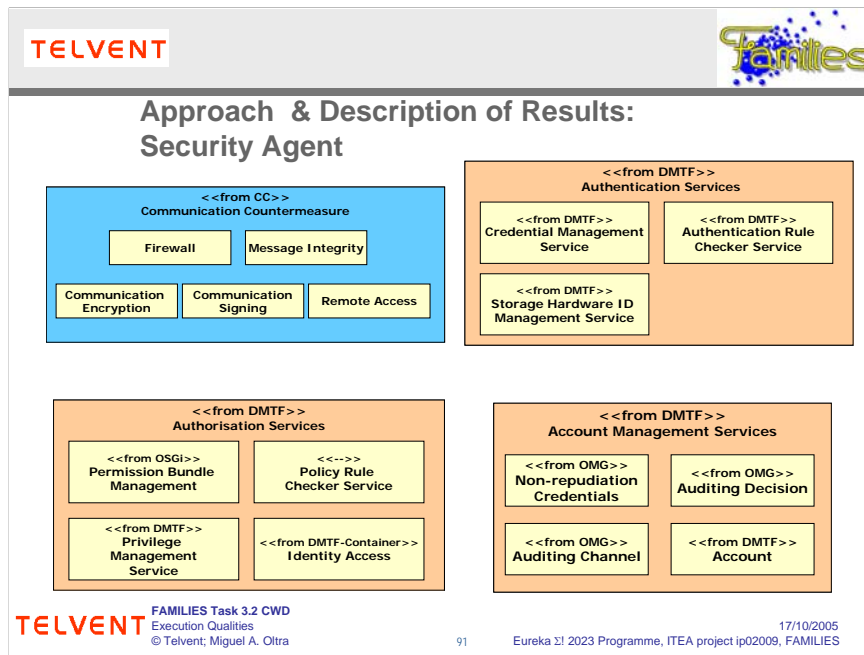
From previous analysis, the security countermeasures were grouped in two main levels or areas (based on concepts identified from the Common Criteria model) (Level 0 of the approach design being followed):

- Access Control Countermeasures
 - Authentication Services
 - Authorization Services
 - Account Management Services
- Communication Countermeasures

Arrows in the figure indicates dependencies among the services grouped by functional blocks. For instance Accounting has dependencies from Authentication and Authorisation services. Following will be mentioned in more detail each of these services grouped by countermeasures areas.

Communication Countermeasures:(see next slide)

- **Firewall:** A common definition of a firewall is a set of related programs, located at a network gateway server, that protects the resources of a private network from users from other networks. In the figure on the next slide, this functionality represents within the so called security agent, that it must provide a firewall functionality. How this firewall functionality is provided, will depend on the implemented solution within the target platform. For instance, this solution could be implemented by means of a firewall provided over the operating system of the platform.
- **Remote access:** represents the functionality of accessing to platform resources remotely. This functionality deals with the management of these kind of accesses.
- **Communication encryption:** under this functionality, are grouped all the related with the confidentiality of remote communications. This functionality deals with data encryption for assuring the mentioned confidentiality among extremes, and also with data de-encryption for presenting non encrypted data to the client.
- **Communication signing:** functionality that deals with the proof of the message origin. Communications can be signed with the credentials of the sender.
- **Message Integrity:** deals with the integrity of a remote received message from a remote client. Integrity message criteria must be defined.



Following are going to be described each of the functionalities grouped by the identified security aspects:

Access control countermeasure:

a) Authentication Services:

- Credential Management Service: deals with the management activities related with the credentials assigned to clients (users or applications) within the system. Among this activities are: validate a credential to a client, credential renewing by means of a management of the relationships with the certification authority, certificate evaluation, among others.
- Authentication Rule Check Service: this functionality deals with the verification of the identity of a client that tries to access or use a resource within the system.
- StorageHardwareIDManagement Service: deals with the management of the identity of hardware devices. This identity must be authenticated in order to guarantee the safety of the platform.

b) Authorization Services:

- Permission bundle management: this functionality deals with the relationships with the OSGi specified service "Permission Admin Service", in terms of accessing to its provided capabilities.
- Identity Access: this functionality is related to the management of identities (e.g. User, Component, ...) that are allowed for accessing to the platform resources. Identity Access is the unique part of the system able to provide interfaces by both UserAdmin and PermissionAdmin services and also validates these permissions. Identity permissions and privileges are defined in permission bundle management and privilege management service respectively.
- Privilege Management Service: this functionality deals with the policy setting for authorisation purposes within the platform. Policy can be defined for identities in terms of privileges in order to grant restricted or not restricted permissions for accessing to available resources on the system.

c) Account management Services:

- Non-repudiation Credentials: provides evidence of application actions in a form that can not be later repudiated.
- Auditing Decision: assists in the detection of current or attempted security violations. This is achieved by recording details of security relevant events on the system.
- Auditing Channel: is used to write audit records on a certain location, where can be checked the evidence of security related events.
- Account: functionality that deals with log service and service tracker by means of its provided interfaces, in order to record same relevant events on the system.

TELVENT

Approach & Description of Results:

Validation Scenario

- A system manager deploys a new service component (bundle)
- Deployment at runtime through Internet connection
- Threats
 - Message spoofing, Identity supersede
 - Message sniffing
 - Platform damage
 - Exploit information from platform
- Countermeasures
 - Admin privileges on the system to allow installation
 - Validation of the integrity of the message
 - System Manager authentication
 - Confidentiality of the message

TELVENT FAMILIES Task 3.2 CWD
Execution Qualities
© Telvent; Miguel A. Oltra

92

Eureka Σ! 2023 Programme, ITEA project ip02009, FAMILIES
17/10/2005

This scenario includes the step 1 (see slide 10) of the approach being followed, where the description of a “Generic Scenario” is provided, indicating information relative to a general description, required infrastructure (high level overview), identified threats and proposed countermeasures.

a) A system manager deploys a new service component (bundle) within the reference architecture of a remote platform (Service Gateway).

b) Component (bundle) deployment in a distributed managed system at runtime through internet connection.

- To provide confidentiality to the communications through Internet is required data encryption at application level.
- To provide authentication and message integrity, message signing is required.

c) Threats include:

- Message spoofing, Identity supersede

Spoofing definition: “Getting one computer on a network to pretend to have the identity of another computer, usually one with special access privileges, so as to obtain access to the other computers on the network”. In this scenario, spoofing can appear, when someone tries to send a request message to the Service Gateway with the credentials of the System Manager, in order to achieve the authentication as System Manager on the Service Gateway.

- Message sniffing

The System Manager credentials, can be obtained from message request sent through Internet. With this credentials, some malicious attack can be done against the Service Gateway, by using these credentials, trying to supersede the System Manager identity.

- Platform damage

A deployment request message is sent to the Service Gateway, containing information for deploying a malicious component over it. The malicious component can be considered a Trojan Horse.

- Exploit information from platform

A malicious component deployed on the platform, can damage/change information stored on the Service Gateway. Also, information can be collected from the Service Platform.

d) Countermeasures:

- System Manager authentication

A proof of the data origin must be provided in the request message. This proof of data origin must include the credentials of the System Manager. This credentials are verified by means of the “Identity Access”. This will allow to proof the identity of the System Manager, and in consequence its authentication on the Service Gateway is validated. The “Remote Access” service must obtain the credentials of the System Manager, and provide them to the “Identity Access”.

- Validation of the integrity of the message

The integrity of the message must be guaranteed in order to avoid the identity supersede of the System Manager on requested messages. The integrity of the message must be achieved by means of the inclusion of the signature of the System Manager and the inclusion of time stamp information in the request message sent to the Service Gateway. The “Message Integrity” must check that both signature and time stamp are valid both together.

- Admin privileges on the system to allow installation

The “Identity Access” must also check that the System Manager has the required privileges (permissions) for achieving the requested deployment service of the Service Gateway. The System Manager privileges are set on the “User Admin Service”. The System Manager requires Admin Permission in order to deploy a component in the Service Gateway.

- Confidentiality of the message

The confidentiality of the message is provided by means of message encryption. The System Manager encrypts the request message with an encryption algorithm. The “Communication Encryption” service must de-encrypt the message. In order to achieve this, the Service Gateway must have the required information for de-encrypt the request message.

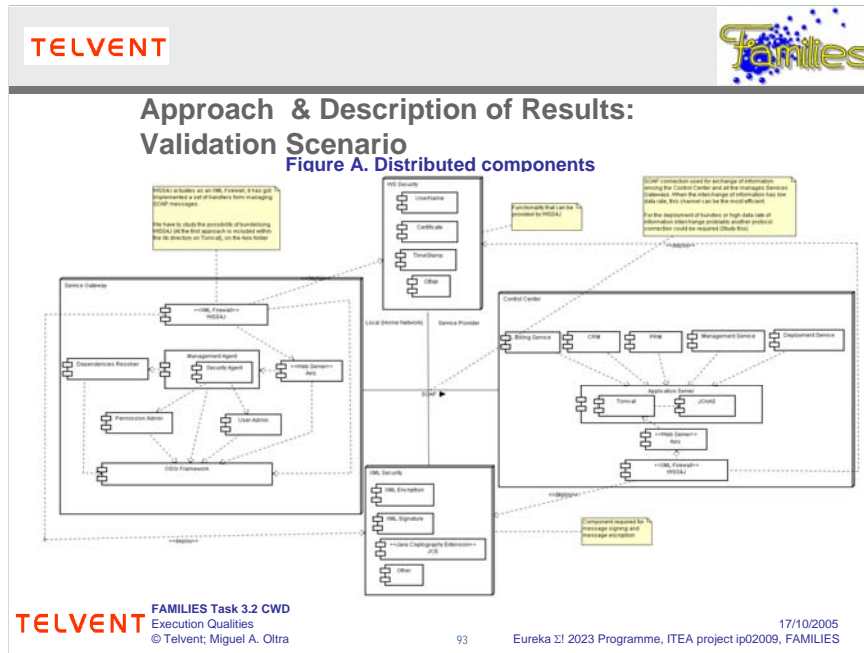


Figure A represents a detailed view of components required on a distributed environment for validating specific functionalities of the Security Agent implemented for an OSGi based platform. Following are indicated in more detail the steps indicated in slide 10 for reducing the scope of the validation scenario

a) Step 2: Criteria definition

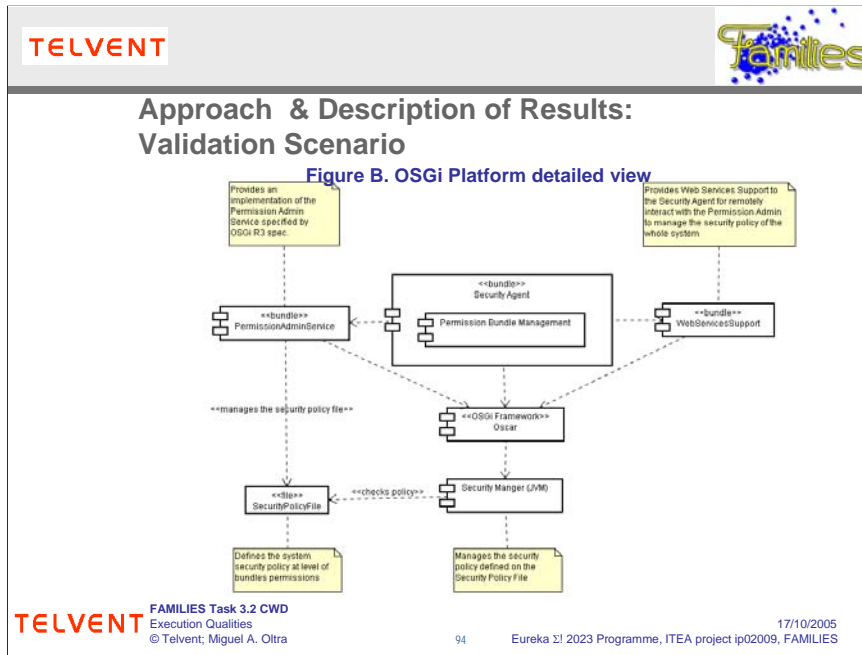
- Quality aspects to be covered: Integrity and Confidentiality
- Countermeasures to be applied are related with Authorisation and Communication

b) Step 3: More refinement at Security Agent level

In concrete the functionalities that are planned to be validated in the Security Agent, once the criteria has been defined from Step 2 are:

- Permission Bundle Management (Authorisation Countermeasure)
- Communication Encryption (Communication Countermeasure)
- Communication Signing (Communication Countermeasure)
- Message Integrity (Communication Countermeasure)

The selection of these functionalities from the Security Agent has implications on the components (variation points) required at OSGi platform level, in order to guarantee the indicated functionalities and quality attributes. Information provided from UPM is required in order to identify the several required components.




c) Step 4: Technologies selection

Remotely bundles permission can be managed through a Web Services Support bundle (Axis + WS-Security). With Axis support, a communication channel can be established between Control Centre and Service Platform (use of SOAP over HTTP). In order to guarantee Integrity and Confidentiality of the communications between extremes over Internet WS-Security is required, and will be implemented by means of previously mentioned Communication Countermeasures on Step 2. A set of required technologies are required for encryption and signing of SOAP messages (XML Encryption, XML Signature implementations from Apache community and JCE implementation from The Legion of the Bouncy Castle community).

Permission Bundle Management interacts with the OSGi specified Permission Admin Service in order to manage bundles permissions. These permissions assigned to bundles are used for authorisation purposes on the Service Platform, for new bundles deployed on the system at run-time. Bundles permissions are stored on a Security Policy File containing information in a format that can be interpreted by the Security Manager included with the Java Virtual Machine who is the entity in charge of checking the policy defined for the system. Figure B represents a detailed view of relationships among components on the Service Platform side.

d) Step 5, the scope of the system for validation has been established with the refinement done in previous steps.

TELVENT


Conclusions & Outlook

- **Requirements identification from standards (OMG, CC, DMTF)**
- **Proposal of a security model for remote management of services platforms**
 - Identification of countermeasures
 - Security Agent
 - Relationship among Security Agent and Services (OSGi)
- **Technologies analysis to comply with security aspects within the services platform**
- **Scenario for validation: model and countermeasures**
- **Validation with Open Source implementations of the technologies**
- **Among the main advantages of the analysis are the categorisation of the security functional aspects in several countermeasures**
 - Incremental evolution of security aspects implemented in the Security Agent
- **This contribution will be a chapter of the research book written in collaboration with UPM contributors**

TELVENT FAMILIES Task 3.2 CWD
 Execution Qualities
 © Telvent; Miguel A. Oltra

95

17/10/2005
 Eureka Σ! 2023 Programme, ITEA project ip02009, FAMILIES

In the following slide are summarised the main conclusions of the work being done under the scope of this contribution. The proposed security reference model is based on a set of well known standards, in order to compile experiences and identify possible lacks of the standards. Also has to be taken into account the target environment, in which the reference security model is going to be personalised. This environment is Internet, and the target platforms are heterogeneous systems, with the same platform architecture but with variability parts (materialised on the deployed services on each system). A proposal of a security agent based on the security reference model has also been proposed by means of a personalisation of the model on the target system, an OSGi based platform. Moreover main countermeasures has been identified both at reference model level and scenario for validation level.

In order to implement a scenario for validation a set of technologies has been identified. Several technologies, can be used for the same purposes during the implementation of the scenario of certain parts of the security agent. Technologies can be seen as variants in the implementation phase. In the approach being followed has been mainly considered Open Source implementations of certain technologies.

As a resume, one of the main advantages of the analysis included in this contribution is the categorisation of the security architectural functional aspects is a set of countermeasures. Also the main objective of the scenario is the establishment of a pre-initial security agent, which will evolve in the future its implemented security aspects gathered on the security reference model.

Indicate that the results of the work done in this contribution will be a chapter of the research book. The chapter will also integrate the work of UPM in task 3.2 and task 5.2 of FAMILIES project.

**Abstract:**

This is the Philips contribution to task 3.2 – Execution Qualities. The title is Improving Security Quality and covers aspects like coping with constantly improved security requirements, embedding security in processes and deployment of security throughout marketing, development and service into maintenance. Security is as good as the weakest part of the chain.

Keywords:

Security, process, procedure, vulnerability, network, internet, regulation, marketing, development, service, Business Unit, PMS

Relation to other tasks in WP3 or other Work Packages of Families:

WP3.2 Resource Usage by M. Weijnenborg. Since both security and resource usage are quality aspects of a system, these two issues were joined in this task.

Acronyms used in the contribution & Glossary of unusual Terms used

HIMSS: Healthcare Information and Management Systems Society

HIPAA: Health Insurance Portability and Accountability Act

KPI: Key Process Indicator

NESSUS: a tool for automated testing and discovery of known security problems

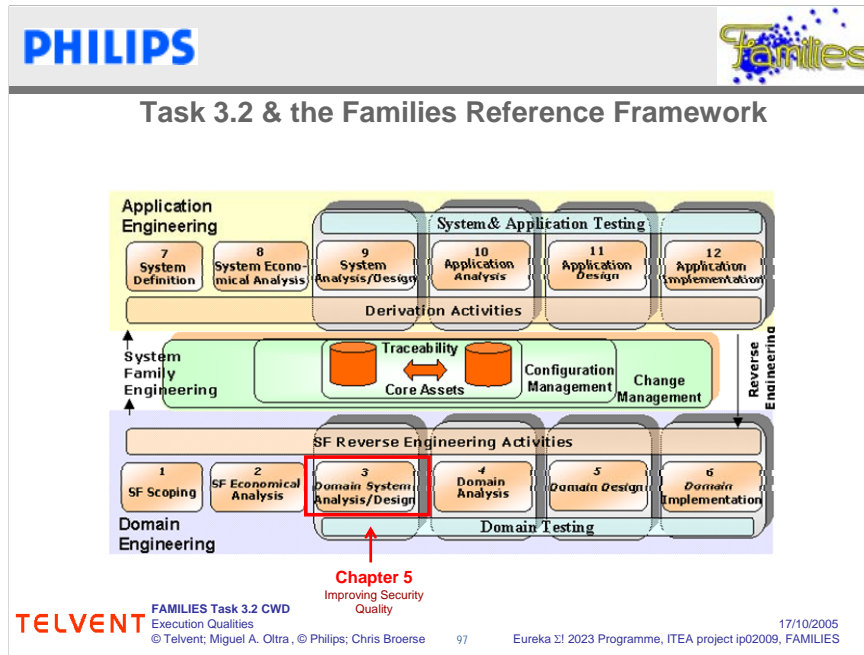
PMS: Philips Medical Systems

Links to additional documents, demonstrators, web pages, etc

HIPAA homepage: <http://www.hipaa.org>

HIMSS homepage: <http://www.himss.org>

Security related information: <http://www.cert.org>



•Chapter 5: Improving security quality (Philips)

Chapter 5 is focused on activities dealing with Domain Engineering, basically with problems that can be considered under the scope of Domain System Analysis/Design activity.

Introduction & Problem Description (1)

- More and more, medical devices are connected to hospital networks which in the end also connect to the internet. Due to the nature of Operating Systems and the applications and services that run on it, security is a quality attribute that may not be neglected.
- Security must be deployed along two axis, across the Business Units which are part of PMS and inside the Business Unit. Goals may not always be the same and conflicts in priorities may arise.
- Within the Business Unit we have to cope with an installed base. The OS and the applications and services may not always be subject to changes. So we have to deal with this in another way.



Security Problem:

Everyone being connected to the internet (whether it be privately or professionally) knows that viruses, worms and hackers form a constant threat to the computer community. This holds for medical equipment too. In fact medical equipment depends on all sorts of network connections: patient information, printers, storage and viewers are all connected to medical devices. Security of medical devices heavily depends upon configuration of assets, processes and ports by the manufacturer and vulnerability of the used platform.

This should not only be once in a lifetime task but must be embedded in the way we work and think. So security should be embedded in our processes and way of working. To make sure we do the right thing we involved Q(uality) A(ssurance).

Although this “security thing” started off in development, we realize that participation of marketing and service is very important. Both are in close contact with the customer, pre sales and after sales and know best the needs and wishes of the customers.

In addition to security, federal issues like HIMMS and the US HIPAA act have positively influenced the security related work.



Introduction & Problem Description (2)

- Due to acquisitions in the past, the installed base includes a variety of Operating Systems, services and applications with which each BU has to deal.
- The same acquisitions lead to a number of different development centers. All development centers must evaluate their processes and align them to optimally support security.


TELVENT FAMILIES Task 3.2 CWD
Execution Qualities
© Telvent; Miguel A. Oltra, © Philips; Chris Broerse 99 Eureka Σ! 2023 Programme, ITEA project ip02009, FAMILIES 17/10/2005

Acquisitions of other companies in the past make the installed base of BU's even more complex and diverse to cope with. There is a wide variety of platforms to support, not to mention the number of assets and services. Apart from the technical challenge there is an organizational issue as well.

Acquisitions from the past has lead to a number of development centres. They all must be connected and aligned to initiatives taken by the security board and realize the importance of being secure and staying secure.

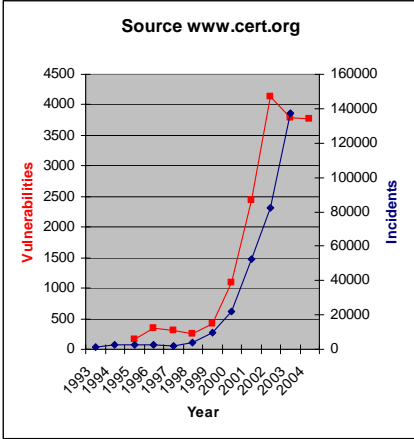
Because in the end this should not only be once in a lifetime task but must be embedded in the way we work and think. So security should be embedded in our processes and way of working. And this all started off in development, we realize that participation of marketing and service is very important. Both are in close contact with the customer, pre sales and after sales and know best the needs and wishes of the customers.

Finally federal issues like HIMMS and the US HIPAA act have positively influenced the security related work. Both issues worked as a "market pull" on various departments with the Business Unit and even more awareness was created within these departments.

PHILIPS


Relevance & Benefits

- Vulnerabilities and Incidents show an ever growing number since 1995.
- Diagnostic equipment is connected to the hospital network and sometimes to the internet. The latter is not safe and due to laptops and USB sticks, internal networks aren't safe either.



Source www.cert.org

TELVENT FAMILIES Task 3.2 CWD
Execution Qualities
© Telvent; Miguel A. Oltra, © Philips; Chris Broerse

17/10/2005
Eureka Σ! 2023 Programme, ITEA project ip02009, FAMILIES

Security quality:



Does the need for security need any relevance? Viruses, worms and hackers are well known to the internet community and so out medical modalities might suffer from these attacks as well. Attached figure shows the number of vulnerabilities (left side) and the number of incidents (right side) as it has grown over the last decade. The CERT organization has even stopped counting incidents due to the automated incident reporting tools that currently exist.

It is enough to say that hospitals and medical centres should simply “keep their front door locked”. By nature, hospitals and medical centres are quite open: people come and go.

But that is not all. Some modalities are portable and are sometimes “left alone” in halls or even waiting rooms. Protection against unwanted “guests” is then an absolute must.

But even the larger modalities (Xray, CT, MR) which have their own operating room and doors that me closed and locked can be subject to attacks. People connect their laptop, USB device (or whatever) and use it in the, so carefully from the outside world protected, hospital information network and then problems arise.

The device must therefore be made secure to be protected not only from the outside but also from the inside.



Relevance & Benefits (2)

- **Among the benefits**
 - Awareness across PMS as a whole and the departments within the BU
 - Relevant departments connected
 - Support of IT infrastructure
 - Process improvements
 - Adapting process descriptions throughout the business
 - Quick reaction to exploit of a vulnerability
 - Work as a team and re-use best practices
 - Bi-weekly Telcon and trimester Face 2 Face meetings
 - Proactive feedback on security related issues

TELVENT FAMILIES Task 3.2 CWD
Execution Qualities
© Telvent; Miguel A. Oltra, © Philips; Chris Broerse 101 Eureka Σ! 2023 Programme, ITEA project ip02009, FAMILIES 17/10/2005

Benefits of the approach can be found in various areas.


The PMS initiative has led to the fact that relevant departments became connected with the security initiative. Especially the HIPAA act (relevant only for the US) mobilized many people in development (in preparing new systems to become HIPAA compliant) as well as marketing and service for customer feedback.

The initiative has led to extra attention towards the IT infrastructure, security at last is more than just the security of a modality; the entire infrastructure contributes to the security of the product. IT department has taken initiatives to improve the infrastructure and supports in regular scans of modalities.

From the beginning the PMS Security group has pushed towards process improvement. Security must be embedded and supported by processes. Processes subject to change were (amongst others): processes that contribute in product creation, processes being part of service and processes describing risk analysis to be regularly performed on modalities.

Last but not least the process improvements have led to processes that quickly act upon exploits in the field. Although the aim is to prevent this at last, it obviously is the ultimate test.

The PMS security initiative consists of a variety of people with various backgrounds. All however represent their Business Unit and represent the various departments active in this field.

PHILIPS


Approach & Results

- **Security approach at BU**
 - Business Unit is responsible for the security of its modalities
 - Business Unit representatives define their own areas of interest
 - Might be derived from TelCon or Face-2-Face meeting
 - Focus on the modality and the BU processes and procedures

- **Security approach at PMS**
 - Focus on managing business risks and business opportunities
 - At this level an inventory is made of common aspects of security
 - Projects are defined with members of the security board to implement these aspects
 - Priorities define the order in which projects are executed
 - Bi-weekly TelCon and trimester Face-2-Face meetings for reporting

TELVENT FAMILIES Task 3.2 CWD
 Execution Qualities
 © Telvent; Miguel A. Oltra, © Philips; Chris Broerse

17/10/2005
 Eureka Σ! 2023 Programme, ITEA project ip02009, FAMILIES

The approach to security is twofold: at BU level and at PMS level.

The BU is bottom line responsible for the modality it makes and thus for the entire security of it. The team at BU level therefore takes the lead in initiatives for securing the product. In the end it is the BU that knows the modality the best.

All BU's have representatives in PMS security board and meeting and telephone conferences are held regularly. Telephone conference are held every two weeks and face 2 face meeting are held every few months. Also at PMS level initiatives are taken and BU may takes these into account at local level.


Next to focussing on the security of the modality, the BU focuses on its processes and procedures. These must support security in all its aspects and must make sure that securing the modality is embedded and supported in all processes throughout the organization. To achieve that the modality security group initiates process improvements and takes initiatives in reviewing.

At PMS level the approach is primarily at managing business risk and opportunities. In this team, which consists of members of the BU, overall issues are handled and taken care of. Amongst others this group has taken care of the following issues:

- Training for service engineers
- Define a security risk assessment
- Start a project on customer feedback
- Take a leading role in HIPAA feedback to customers. Also input from HIMSS was taken into account.
- Define KPI's by which security efficiency can be measured.


We have set priorities for all these activities.

At the PMS level we learn, help, improve and stimulate each other. In the end: it is a teamwork.

PHILIPS

Approach & Results (2)

- **At PMS level we have (amongst others) indicated and started the following projects:**
 - Customer WEB site.
Goal: improve feedback to the customer on security issues and proceedings of vulnerabilities and exploits
 - Emergency Operation Plan
Goal: describe the emergency operation process and indicate the responsible persons
 - Product Risks assessment
Goal: describe a risk assessment process which will be applied to all BU's
 - Education
Goal: prioritize on **who** should attend **what** kind of training with respect to security

 **FAMILIES Task 3.2 CWD**
Execution Qualities
© Telvent; Miguel A. Oltra, © Philips; Chris Broerse

103

17/10/2005
Eureka Σ! 2023 Programme, ITEA project ip02009, FAMILIES



At PMS level we aim at a wider scope and at broader goals than just modality level. So here I would like to give an overview of the initiatives started.

First of all it is all about customer satisfaction. We recognized from the beginning that it would be important to give proper feedback to customers. They often use computers at home which get updated one in a while and may obviously be surprised when their computer in the office does not get this update. Since this office computer may also be a computer with which the modality can be used it needs updates one in a while.

We definitely need an emergency operation plan. In case somewhere an emergency happens proper people need to be informed and proper actions need to be taken.

At PMS level we created a product risk assessment process. It was created by a smaller group of people of the PMS security group and reviewed in the large group to be adopted and used by all BU's.

Education at last has also been tackled by the PMS group. This subject too has been executed by a smaller group of people. Marketing and service were here the primary departments which took the lead in creating an education plan for service engineers to start with. The entire program has been reviewed by the large PMS group.



Approach & Results (3)

- **At BU level we have indicated and started the following activities:**
 - Analyze our installed base and make a risk analysis. Advise and implement minimum security baseline.
 - Make security a way of working within the product creation process.
 - Regular NESSUS scans are executed.
 - Product security documentation has become standard project documentation
 - Subscribe to vulnerability reporting: keep up-to-date. Implement and roll out patches for applicable vulnerabilities
 - Optimize the way of working in case of a security incident and describe this in a process. Preferably align this process with already existing ones.
 - Introduce a way to keep track of Key Process Indicators and measure these regularly.

TELVENT FAMILIES Task 3.2 CWD
Execution Qualities
© Telvent; Miguel A. Oltra, © Philips; Chris Broerse 104 17/10/2005
Eureka Σ! 2023 Programme, ITEA project ip02009, FAMILIES

At BU level activities were carried out as result of PMS activities or just because we found these to be very useful.


A risk analysis of the entire installed base was carried out. This already has lead to improvements of the installed base.

Next we made security a way of working in the development department. During normal product development a NESSUS scan has become regular practice as well as documentation on security. NESSUS is an open source vulnerability scanner.

There are various security related sites at which it is possible to subscribe to security related information. Next to that it is possible to subscribe to alerts which usually indicate effected platform, assets or platform services. By doing this we stay up-to-date to things going on the platform we use and its associated assets.

Last but not least, we carefully looked at and described our process in case of an incident. Since we already have a well described process for field problem reports, we aligned this with security incidents.

As a result of the activities at PMS level we started measuring relevant KPI's.

PHILIPS


Conclusion & Outlook

- **At PMS level**
 - we have created a team of people going strong for security. This team initiates new initiatives to be worked out.
 - we have indicated our major points of interest. We have prioritized these and created projects to implement.
 - we have indicated Key Process Indicators. We are in the process of installing and measuring.
 - we have broadened our scope by including marketing, service and quality departments at corporate level
 - we created and reviewed a security site assessment for all BU's.
- **At BU level**
 - we assessed risks of the entire installed base based on an externally available process description.
 - we improved (and still are improving) our processes

TELVENT FAMILIES Task 3.2 CWD
 Execution Qualities
 © Telvent; Miguel A. Oltra, © Philips; Chris Broerse

17/10/2005
 Eureka Σ! 2023 Programme, ITEA project ip02009, FAMILIES


Conclusion:

Both teams are fully operational and have achieved a lot and are still working on numerous things.

Although it all started strictly from a technical point of view, now other departments are involved and this has had impact on the amount of projects and the scope of the projects. We created a number of projects and prioritized these.

Since it comes down to security of all modalities we have created a security risks assessment to be used in all BU's.

At BU level we have made minor and major steps in improving modality security, both from technical point of view and non-technical. Among the non-technical is at least the processes of which some have changed and some are still in the process of changing.

PHILIPS


Conclusion & Outlook (2)

- **For the Outlook**
 - Improving customer feedback (various ideas)
 - Emergency Operation
 - Virus scanner policy
 - Use KPI's to steer the weak spots

- Remaining risks:
 - Lack of interest for security
 - It is non functional
 - Still heard: "Isn't this a hospital responsibility?"
 - Prioritization
 - At both levels (PMS and BU) aspects of security will have to be prioritized which may lead to conflicts
 - HIPAA consequences may be pulled into the security teams

TELVENT FAMILIES Task 3.2 CWD
Execution Qualities
© Telvent; Miguel A. Oltra, © Philips; Chris Broerse 106

17/10/2005
Eureka Σ! 2023 Programme, ITEA project ip02009, FAMILIES

At both levels it is nice to see that initiatives are taken and worked out.

So for the future we are still looking out for existing and new ideas to be worked out. Among these ideas:

- Improving customer feedback (web site etc.)
- Emergency operation: describe what, when and who.
- Define a virus scanner policy.
- Use KPI's to actually steer on weak spots.

Although all this is achieved and people are enthusiastic to continue some risks remain.

- Still security lacks interest (should something really awful happen first).
- On the prioritization we see that both activities on PMS level and BU level take there time and the right balance must be found. We also see that HIPAA is pulled into the security area.



Shaping a World
of Convergence



Families Task 3.2 CWD

Execution Qualities

Section II: Other Run-Time Quality Attributes

Miguel A. Oltra
miguel.oltra@telvent.abengoa.com



Task 3.2
Execution Qualities



Chapters: Section II

- Chapter 6. Quality of Service for Real-time and Embedded Systems (Thales & CEA)
- Chapter 7. Resource usage (Philips)
- Chapter 8. Predicting Reliability and Availability at the Architectural Level (VTT)



TELVENT
Shaping a World of Convergence

Families Task 3.2 CWD

**Chapter 6:
Quality of Service for Real-time
and Embedded Systems**

Laurent Rioux
Laurent.Rioux@thalesgroup.com

Sébastien Gérard
Sebastien.Gerard@cea.fr

THALES

cea list

**Task 3.2
Execution Qualities**

Abstract:

The purpose of the CEA and THALES in this task was to provide a methodology to validate real-time QoS of RTE systems. In this context, the CEA-List provided a method and a support tool to perform schedulability analysis on UML models adorned with RT QoS. The proposed methodology is based on the Accord/UML methodology [2,4], which is dedicated to modelling embedded real-time systems, so that it can also serve as a scheduling analysis support for a better modelling of real time applications. Based on the schedulability analysis profile, we propose a framework for the construction of a schedulability model [3]. This is the model used later to verify the schedulability of the application and this can be used after for performance evaluation for instance [1]. Then, we describe the architecture and basic principles of the tool used to perform actual schedulability analysis. Then, we explain how to interpret and apply the results provided by the schedulability analyser.

Consideration of RT QoS is important when developing family of products. More particularly, in the RTES that we considered here, real-time features validation is a crucial point in the development. But when should we do this verification if we consider product lines? What can be validated if we consider a product family and not a product? These general questions doest not find here direct answers, but we interested us in this work in the expression of these RT QoS features, in the definition of a general methodology that helps users in validating its systems from a general application model until a final prototyped product. The classification of what kind of property can be validated at the product family level and at the final product level is not addressed here because we first want to be as precise as possible in the model definition to guarantee a correct product validation.

Keywords: Real-Time, Embedded, QoS, Schedulability analysis, real-time validation.

Relation to previous work in ESAPS & CAFÉ

CEA was not a partner of ESAPS & CAFÉ.

THALES took the work done in CAFÉ and ESAPS as input to this work. Especially modelling basic concepts of QoS and the specificity for QoS product line. Because the QoS needs to support variability for product lines, the QoS for Real-time and Embedded systems must be generic (not specific to products)

Relation to other tasks in WP3 or other Work Packages of Families

Relations with WP4, Task 4.1 (Domain and application modelling practices / Guidelines for Variability Modelling)

Glossary:

Component, Real-Time features, Domain Design, Real-Time requirements, use Case, Scenario.

Acronyms:

MDA: Model-Driven Architecture

QoS: Quality of Service

RT : Real-Time

RTE: Real-Time Embedded

RTES: Real-Time Embedded Systems

RTS Real-Time Systems

UML: Unified Modelling Language

References:

[1] N. Torrecillas, H. Dubois, and S. Gérard, "Performance Evaluation of Real-Time Embedded Systems with the Accord/UML Methodology," presented at 20th Annual UK Performance Engineering Workshop, 2004.

[2] S. Gérard, C. Mraidha, F. Terrier, and B. Baudry, "A UML-based Concept for High Concurrency: the Real-Time Object," presented at The 7th IEEE International Symposium on Object-oriented Real-time distributed Computing (ISORC'2004), J. Gustafsson, T. Aoki, and I. Lee, IEEE, pp, Vienna, Austria, 12-14 May 2004 2004.

[3] P. T. Hieu, S. Gérard, F. Terrier, and D. Lugato, "Scheduling Validation for UML-modeled real-time systems," presented at Workshop WiP (EuroMicro'03), Portugal, Porto2003.

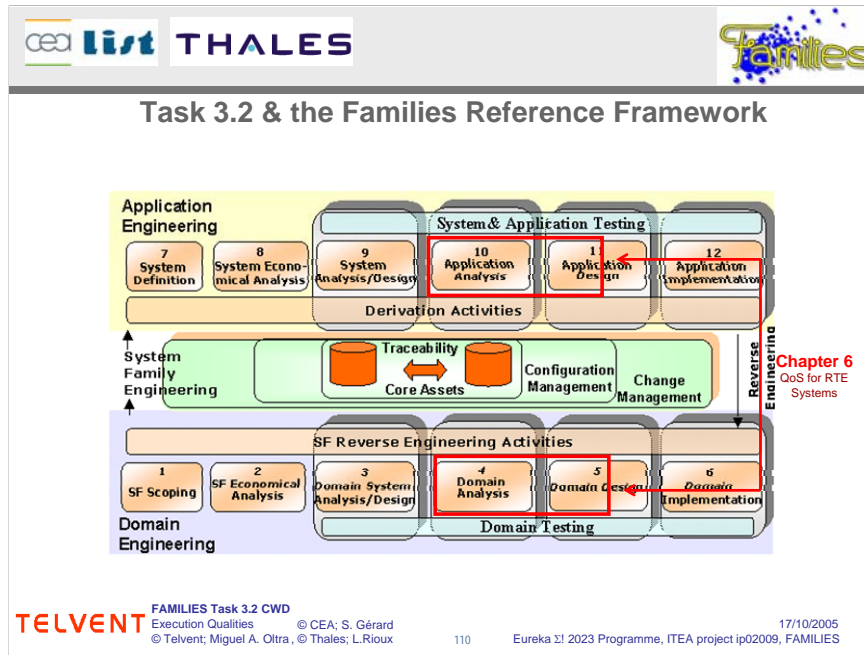
[4] S. Gérard, "The ACCORD/UML methodology," CEA-LIST, Paris, internal report DTSI/SLA/02-326, 2003.

Links to additional documents, demonstrators, web pages, etc

Two documents written by the CEA-List as deliverables for Families WP3:

"Introduction of RT-QoS in component interfaces"

•"User guide to validate RT-QoS in UML models"




•Chapter 6: Quality of Service for Real-time and Embedded Systems (Thales & CEA)


Chapter 6 (CEA and THALES contribution) is more concentrate on Application and Domain Engineering parts.

For the Application Engineering part, it is proposed a way to design and to analyse systems, more specifically real-time embedded applications, and more particularly in the schedulability analysis: a general approach is defined and a connection to a specific analysis tool is performed. Application design follows the proposed design process.

As a methodology is defined for analysis and design of systems, a generic approach related to the specific domain of real-time embedded systems is developed. Previous contributions followed the SPT and QoS & FT UML profiles and now, a new UML profile for MARTE (Modelling and Analysis of Real-Time Embedded systems) is under definition and CEA and THALES are two of the initiators of this RFP. Thus, the specific domain of RTEs is here considered and solutions will be developed in the next months towards a standardisation of them.




THALES



Introduction & Problem Description

- **Real-Time (RT) systems are complex to model**
 - Quantitative & qualitative RT features to be taken into account within models
- **Embedded systems require specific quality of services**
 - Notion of processors, memory size, bandwidth, power consumption...
- **Real-Time and embedded systems require validation and simulation at high abstraction level**
 - To ensure that allocated resources may satisfy required QoS
- **Problems**
 - **How to develop RTE systems by using standard solutions such as:**
 - UML modelling with RTE and QoS extensions
 - component-based software architecture
 - **How to validate and simulate UML models with RTE and QoS extensions (more specifically schedulability analysis)**



FAMILIES Task 3.2 CWD
Execution Qualities
© Telvent; Miguel A. Oltra, © Thales; L.Rioux

© CEA; S. Gérard
111

17/10/2005
Eureka Σ! 2023 Programme, ITEA project ip02009, FAMILIES

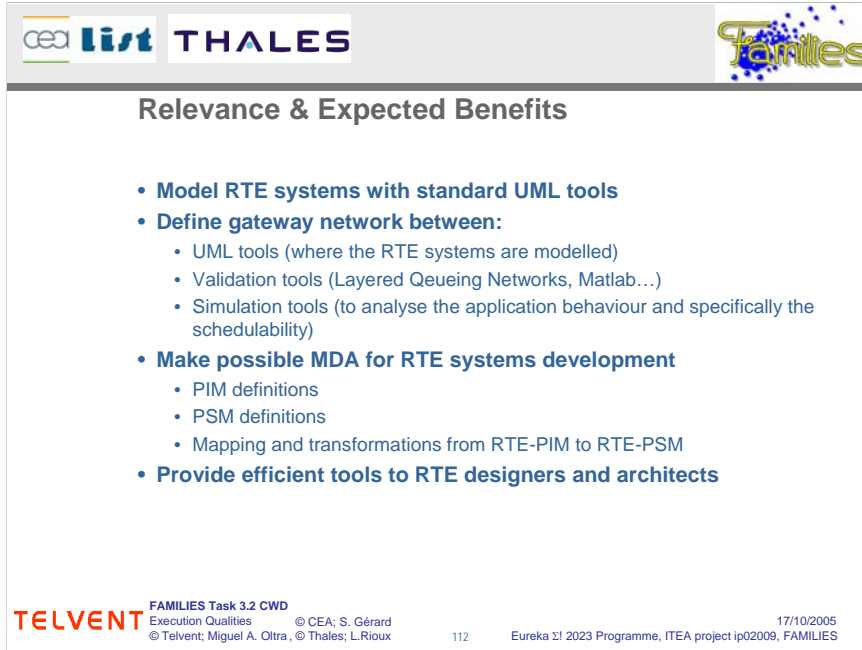
In comparison with usual application domains, real-time systems (in short RTS) development requires on the one hand the possibility of modelling quantitative features such as deadline, period, etc. And on the other hand, RTS design needs the modelling language to support RT qualitative features.

Moreover, when RTS are also embedded systems (in short RTES), it is very important to ensure modelling of hardware aspects and also their integration with the software aspects of the system (co-design problematic with hardware and software considerations).

RTES are usually very critical systems that need to be validated with a high level of confidence before being distributed.

Finally, all of these activities have to be done in the context of the standard in order to improve usability and future of the proposals.

In this standardisation work, THALES and CEA-List are both involved in the OMG consortium. Furthermore, a current French project involving THALES, CEA-List and also INRIA, which is called CARROLL-PROTES, has started in 2004 with the objective to define a new UML profile dedicated to modelling and analysing of real-time embedded systems. The THALES and CEA-List actions in Families project are clearly a first step in this direction.



Relevance & Expected Benefits



- **Model RTE systems with standard UML tools**
- **Define gateway network between:**
 - UML tools (where the RTE systems are modelled)
 - Validation tools (Layered Queueing Networks, Matlab...)
 - Simulation tools (to analyse the application behaviour and specifically the schedulability)
- **Make possible MDA for RTE systems development**
 - PIM definitions
 - PSM definitions
 - Mapping and transformations from RTE-PIM to RTE-PSM
- **Provide efficient tools to RTE designers and architects**


TELVENT FAMILIES Task 3.2 CWD
Execution Qualities © CEA; S. Gérard
© Telvent; Miguel A. Oltra, © Thales; L.Rioux 112 Eureka Σ! 2023 Programme, ITEA project ip02009, FAMILIES 17/10/2005

The most important benefit that can be expected if the profile will be to have a standard way to model real-time and embedded systems. This profile will bring a gateway between UML tools, validation tools and simulations tools. Validations tools are generally formal or mathematical tools (like Layered Queueing Networks approach for performance analysis, Agatha test generator for test generation to validate UML models for instance, connection to MAST tools, etc...). Simulation tools will be able to analyse the application behaviour.

Another objective is to permit to use MDA approach for developing RTE systems by defining Platform Independent Model (PIM) and Platform Specific Model (PSM) and mapping between these two models.


And finally provide efficient tools for RTE engineers and architects.



Approach & Expected Results

- **Specifying a UML profile for RTE systems defining:**
 - A RTE-PIM metamodel
 - A RTE-PSM metamodel
 - Mappings from PIM to PSM
 - A RT-component concept
 - Based on UML2 component concept
 - Based on RT-CCM
 - Based on both standard UML profiles: SPT and QoS
 - A sub-profile for schedulability analysis of RT-Components
 - A tool to validate RT QoS for schedulability
- **Try to standardise the defined profile:**
 - To save investment
 - To improve the profile with the IT communities and other RTE experts
 - To get tools available on the market
 - To get experts / consultants on the RTE domains



FAMILIES Task 3.2 CWD
Execution Qualities © CEA; S. Gérard
© Telvent; Miguel A. Oltra, © Thales; L.Rioux

17/10/2005
Eureka Σ! 2023 Programme, ITEA project ip02009, FAMILIES

To realise the profile for RTE, we have specified a platform Independent Metamodel (PIM), a Platform Specific Metamodel (PSM) and mapping rules between these two metamodels.

We also have introduced the notion of Real-time component based on the notion of UML 2.0 component.

This notion of Real-time component is described in the document “Introduction of RT-QoS in component interfaces” done by the CEA-List.

In this document, we have described in detail our proposal to integrate real-time QoS (RT-QoS) in the context of the component model of the UML2. To achieve this goal, the document is organised as follow. We have firstly described respectively an extension of QoS to support the specification of real-time features of applications, and the UML2 concept of component. Then, we specify how the previously defined RT-QoS may be integrated in the UML2 component model defining also a RT-component model in the context of the UML2.

In the document “User guide to validate RT-QoS in UML models” we have also defined a process with appropriate tools to more particularly validate schedulability RT QoS.



The purpose of this work is to add a further dimension to the $Accord_{UML}$ methodology (to remain, $Accord_{UML}$ is a methodology used to design real-time embedded systems based on UML and appropriate extensions presented below), which is dedicated to modelling embedded real-time systems, so that it can also serve as a scheduling analysis support for a better modellisation of real time applications.


We add to the initial $Accord_{UML}$ framework facilities to extract dedicated information useful to analysis of the models, extensions to transform models and to integrate in them useful data for simulation and therefore analysis; analysis results can then be investigate to correct initial models.

Based on the schedulability analysis profile we propose a set of modelling rules for construction of a schedulability model. This is the model used later to verify the schedulability of the application. Then, we describe the architecture and basic principles of the tool used to perform actual schedulability analysis. Then, we explain how to interpret and apply the results provided by the schedulability analyser. Finally, we give a few recipes for correcting analysed models that contain schedulability errors. This is based on the $Accord_{UML}$ methodology with a connection to the AGATHA tool usually used for execution paths generation and here used to validate scheduling requirements.

We have also studied the opportunity to use the concept of Real-time CORBA component (RT-CCM) in order to be compliant with the current standard of schedulability analysis.

We want to standardise this profile to save investment, to insure interchangeable models between different partners. Also, to get tools on the market which will support an unified way to model real-time systems. This standardisation actions were presented in the previous slide and the actual state of the OMG consortium standardisation action is given in conclusion.







A sub-profile for schedulability analysis of RT-Components

- **The starting model: an application model**
 - Structure of the application
 - Internal behaviour of the system

⇒ lacks in this model to perform a RT QoS analysis
- **The target model: an analysis model for RT QoS analysis**
 - Data extraction from the application model to handle RT QoS analysis
 - Enrichment of the application model with appropriate values
- **Implementation within the Accord_{UML} methodology that supports a connection to the Agatha analysis tool**



FAMILIES Task 3.2 CWD
Execution Qualities © CEA; S. Gérard
© Telvent; Miguel A. Oltra, © Thales; L.Rioux

114

17/10/2005
Eureka Σ! 2023 Programme, ITEA project ip02009, FAMILIES

As we want to consider schedulability analysis for RT-component, we start from an application model where the structure is defined as well as the behaviour. Usually, such models are not expressive enough to be directly linked to QoS or more specifically schedulability analysis tools: temporal information (deadline, ready time, periodicity, etc...), assumptions on the system in order to have an analysis compliant with the real execution among others are missing.

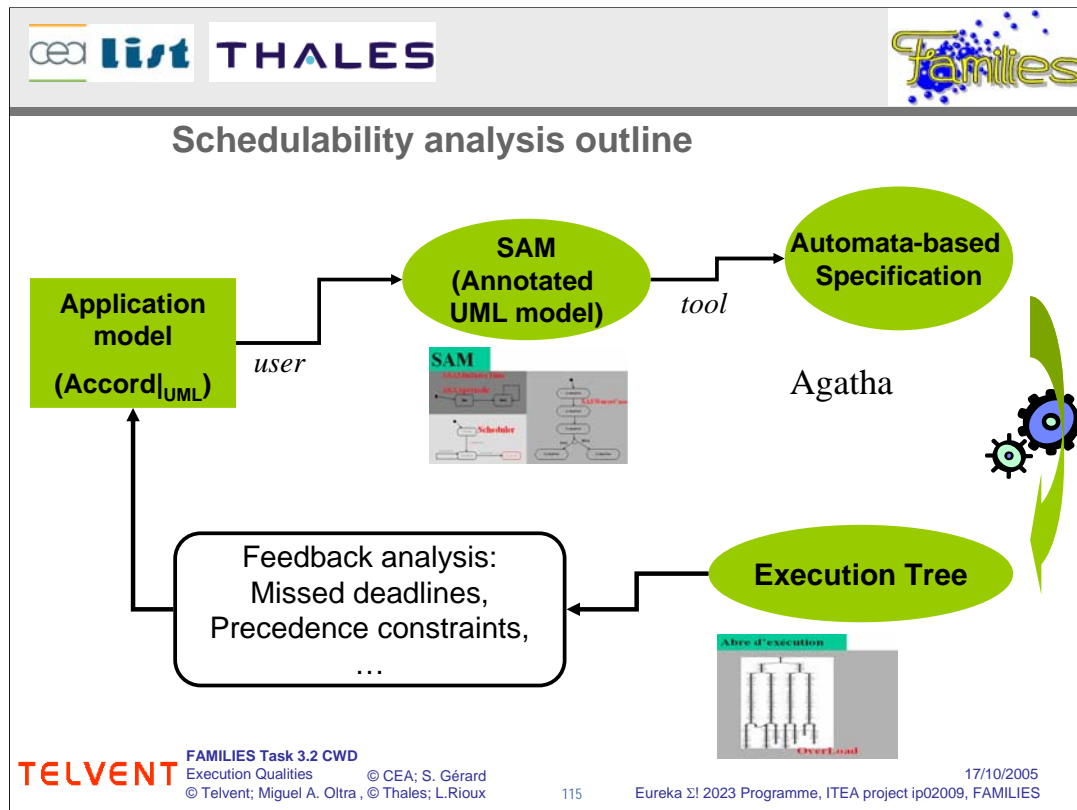
The way this is managed is the following: we consider first an application model and we extract from it the data which is needed for RT QoS analysis: we use model transformation techniques to manage this extraction and to select views of the input model to obtain a target model. Then, this model is enriched with appropriate values to be analysed.

Certain solutions are provided by commercial UML-based and sometimes SPT-based tools and methods such as Rose-RT (IBM Rational Software) or Rhapsody (I-Logix). However, specification of real-time features frequently relies on low-level implementation methods; and these tools/methods do not provide the level of abstraction needed for specifying an application's real-time characteristics. In other words, they still require developers to have thorough knowledge of real time-related development and analysis techniques.

An implementation of this has been done within the Accord_{UML} methodology developed in CEA-List. This implementation manages the defined sub-profile, the model transformation, the connection to the Agatha tool that is used in schedulability analysis of the model and feedback of the results can be obtained and then analysed into the initial model.

Accord_{UML} is a modelling methodology developed by the CEA-List [1,4]. It is dedicated to model-driven development of embedded and distributed real-time systems. While originally defined for an auto industry application, it is currently being assessed for use in other sectors such as telecommunications. This method allows engineers who are not real time specialists to construct complete, unambiguous embedded system specifications and leads them in the applications development process, wherever possible also automating it. Accord_{UML} was therefore given a model-driven approach, making maximum use of model transformation techniques to take users through the step-by-step modelling process and obtain an optimized final code. The purpose of the present study is to add a further dimension to Accord_{UML} so that it can also serve as a scheduling analysis support for modelled real-time applications.

The AGATHA tool proposed by the CEA-List was initially developed to check the consistency of system implementation with respect to project specifications. It is simulation-driven and relies on use of symbolic execution techniques. Its basic approach is to reformulate specifications in a format that is both compact and exhibits an easy-to-read set of basic behaviours inferred from the specification. If these behaviours are inconsistent with the specification, the latter is modified and the simulation process repeated.



This picture illustrates the process that is used here: we can see the starting model on the left, then, the production of an intermediate model that can be connected to the verification tool (here, the Agatha tool); then, the results are given as execution trees which can be analysed for feedback (corrections are made until we obtain a final correct model).

Accord_{UML} methodology relies on use of two analysis (or "specification") models with different but complementary levels of abstraction, followed by a prototyping model for validation. Preliminary analysis identifies the domain concepts and system requirements in the form of use cases and high level scenarios. Detailed analysis, which corresponds to the Application Model in the figure, then specifies in detail all system internal functions required to accomplish the previously identified functionalities (i.e. use cases). This includes, for example, detailed descriptions of the system's structure and its behaviour. Once the Application Model is completed, we extract of the Application Model parts of the model that provide the data needed to construct the schedulability analysis model (this model is noted the SAM). This extraction is provided by a classical model transformation program (for the prototype here presented, it is written in J language; language used in Objecteering. This can be done in any other transformation language such as Java, or dedicated model transformation languages like ATL (INRIA/Nantes language), MTL (INRIA/Rennes) or others...).

The SAM consists of three main packages that are composed of 1) what we extract from the application model and 2) what the user will have to complete.



To use Agatha's symbolic execution engine, it is necessary to translate the UML model analysed (here the SAM) into the Agatha input formalism. The following paragraphs describe the interface devised to achieve this, which relies mainly on model transformation techniques. The transition from a UML model (SAM) to the symbolic execution engine formalism is also explained; and this is followed by a detailed description of the main transformation modules.


Agatha's original vocation was to produce automated functional tests for an application based on concurrent, communicating automata. Agatha thus provides a symbolic execution engine and a set of ad hoc tools required for analysis and instantiation of the functional tests. To adapt Agatha capabilities to real time behavioural analysis (based on symbolic execution), some changes were required to credit time, limit combinatorial explosion and simulate system execution with a given scheduling algorithm.

Symbolic execution is performed as a simulation for all system behaviour cases, according to the values of system variables and the different conditions for external event occurrence.

Symbolic execution by an Agatha execution engine extended to the real time domain generates a set of scenarios that describe all the application's symbolic execution paths. Each of these scenarios represents a scheduling plan corresponding to the different symbolic values of variables or to those fireable transitions that were fired from a control automaton state.


Results can be then analysed in order to correct the input model and then, the cycle can be redone as long as a model with a correct analysis is obtained. Mistakes and results are detailed in next slides.

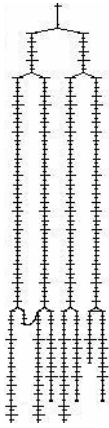






Results and feedback to initial model

- Missed deadlines with deadlocks: automatic result and direct feedback to the model





- Precedence constraints can be improved from analysis of the tree: result by scenarios and by hand



FAMILIES Task 3.2 CWD
Execution Qualities © CEA; S. Gérard
© Telvent; Miguel A. Oltra, © Thales; L.Rioux

116

17/10/2005
Eureka Σ! 2023 Programme, ITEA project ip02009, FAMILIES

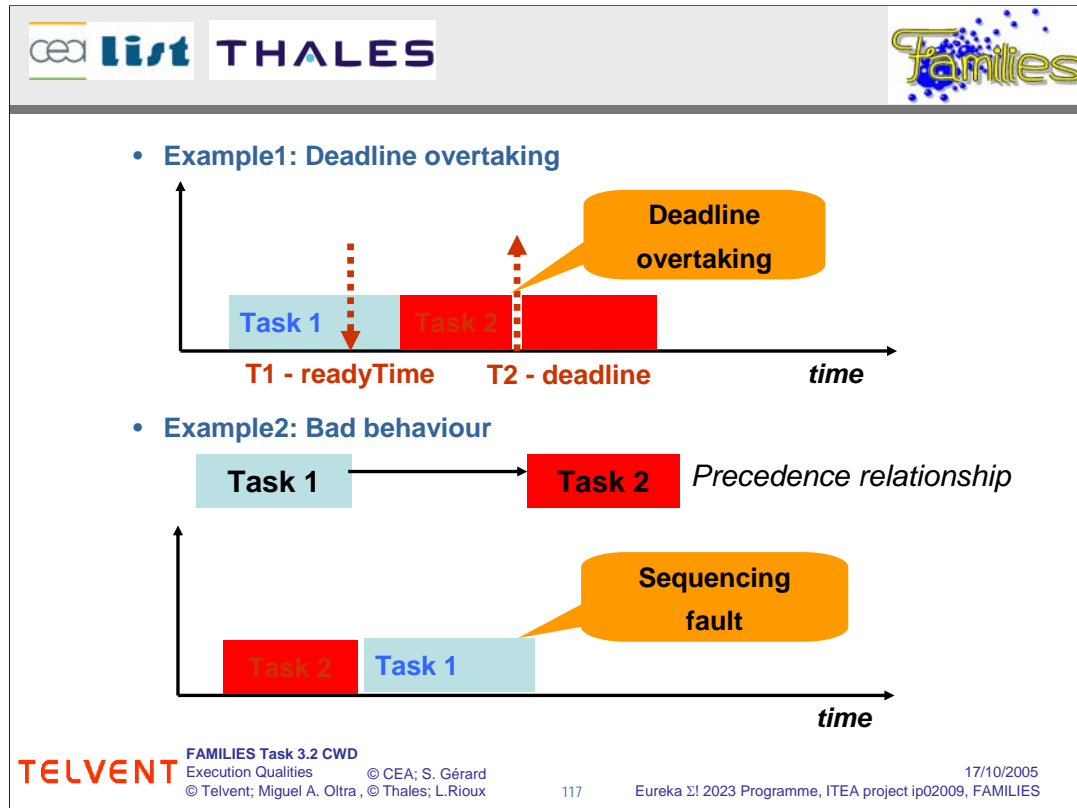
Here are two kinds of results: one with a deadlock (a correction has to be done to avoid this; this corresponds to a mistake in the requirements) on the left; and a complete execution tree on the right (this corresponds to a correct analysis of the requirements). In this last case, the behaviour of the system do not include a critical section with respect to temporal requirements and this kind of results validate the model with the RT QoS requirements (RT QoS are valid requirements). Each choice point in the system execution leads to another branch in the execution graph.

Symbolic execution results in a graph of achievable states like the one shown in second Figure on the right. Each node of this graph correspond to a reachable state of the system during its execution. A constraint is associated to each node and a path constraint is then constructed step by step, which corresponds to the possible values for each variable of the system considered in the given execution path.

A system state is represented by the set of parameters, that is:

- a set of current control states for the modules making up the system.
- the symbolic values of system variables.
- states of control object mailboxes containing applicative messages, and the scheduler queue *exeQueue* containing all tasks triggered or interrupted or waiting for execution.
- current time and system load.

A *deadlock* is defined as a special system state in which the system cannot evolve from its current state because of a modelling defect. Deadlocks can also occur on detection of a system overload or functional defect. A deadlock state is shown below as illustrated in the left Figure in red (on the symbolic execution graph).



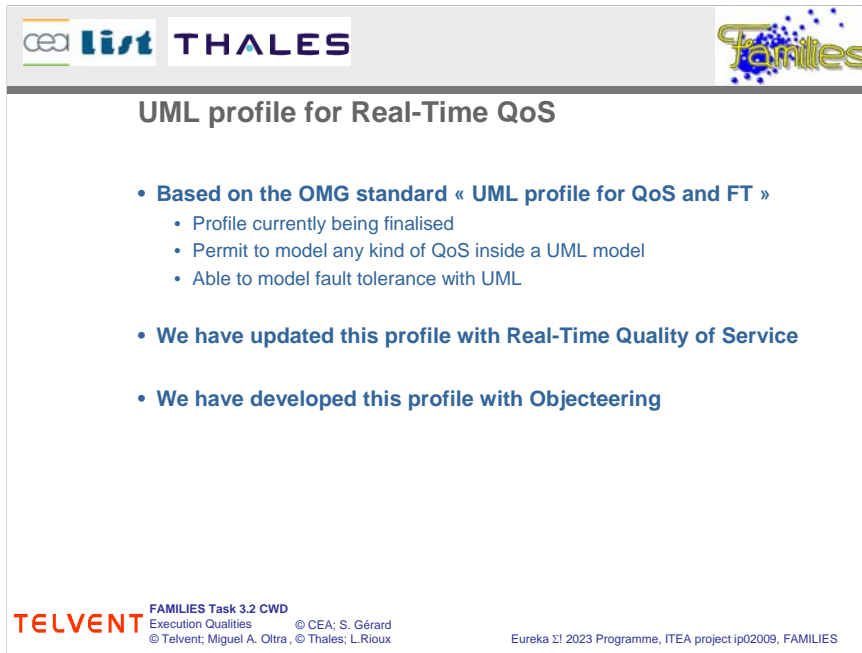
Here are two kinds of mistakes that can lead to a deadlock result in Agatha. This has to be corrected in the temporal requirements to lead to a success result from the validation tool. If path constraint cannot be solved (which corresponds to an unreachable state), this also lead to another kind of error (not represented by a deadlock) in the execution graph.

The first example here presented illustrates the end of the second task after the supposed deadline (the ReadyTime value corresponds to the moment from which the second task can start its execution). This kind of mistake leads to a deadlock in the verification tool because temporal requirements are not respected.

The second example illustrates an inversion in the sequencing and then, when considering such execution path, the precedence relationship is not respected and this leads again to a deadlock in the Agatha verification tool.

The two examples shown here are not the only cases where the Agatha verification tool detects a deadlock or an error: when a timing constraint is not respected (wrong synchronisation, a waiting time higher than the difference between deadline and ReadyTime, a general timing constraints not respected, a deadline not respected because an asynchronous message call answer is still missing, etc...). The Agatha verification system does not give a feedback on the time available if the system do not uses all the resources it can.

Scenario analysis by symbolic execution is important in that; it detects system action schedulability, sequencing errors, timing errors, conception errors. This meets the schedulability analysis goal. Schedulability errors are detected automatically by analyzing deadlock states in the different branches of the symbolic execution graph. Such states namely symbolize system overload problems. As previously, sequencing errors are also automatically detected. They can be corrected either by changing parameter real-time values or modifying the architecture of the model itself.



The slide features a header with logos for 'cea list THALES' and 'Families'. The main content is titled 'UML profile for Real-Time QoS' and contains three bullet points. The footer includes the 'TELVENT' logo, project information for 'FAMILIES Task 3.2 CWD', and copyright notices for CEA, S. Gérard, Telvent, Miguel A. Oltra, Thales, and L. Rioux. It also references the 'Eureka Σ! 2023 Programme, ITEA project ip02009, FAMILIES'.

UML profile for Real-Time QoS



- **Based on the OMG standard « UML profile for QoS and FT »**
 - Profile currently being finalised
 - Permit to model any kind of QoS inside a UML model
 - Able to model fault tolerance with UML
- **We have updated this profile with Real-Time Quality of Service**
- **We have developed this profile with Objecteering**


TELVENT FAMILIES Task 3.2 CWD
Execution Qualities © CEA; S. Gérard
© Telvent; Miguel A. Oltra, © Thales; L. Rioux

Eureka Σ! 2023 Programme, ITEA project ip02009, FAMILIES

We have decided to reuse the standard of UML profile for QoS and FT for our profile dedicated to real-time quality of service for system families. We have worked on the finalisation task force of this standard to ensure it will meet our requirements for the real-time extensions.

After while, we have implemented the UML profile for QOS dedicated to real-time system families inside Objecteering (a SOFTEAM tool) and we have implemented our real-time extension.







RTE-QoS Concepts

- **Permit to define real-time and embedded QoS**
 - RTE-QoSCategory,
 - RTE-QoSCharacteristic,
 - RTE-QoSDimension

- **Permit to use these characteristics defined inside the UML model and to apply constraints on it**
 - RTE-QoSConstraint (and RTE-QoSContext)
 - RTE-QoSValue,
 - RTE-QoSDimensionSlot



FAMILIES Task 3.2 CWD
 Execution Qualities © CEA; S. Gérard
 © Telvent; Miguel A. Oltra, © Thales; L.Rioux

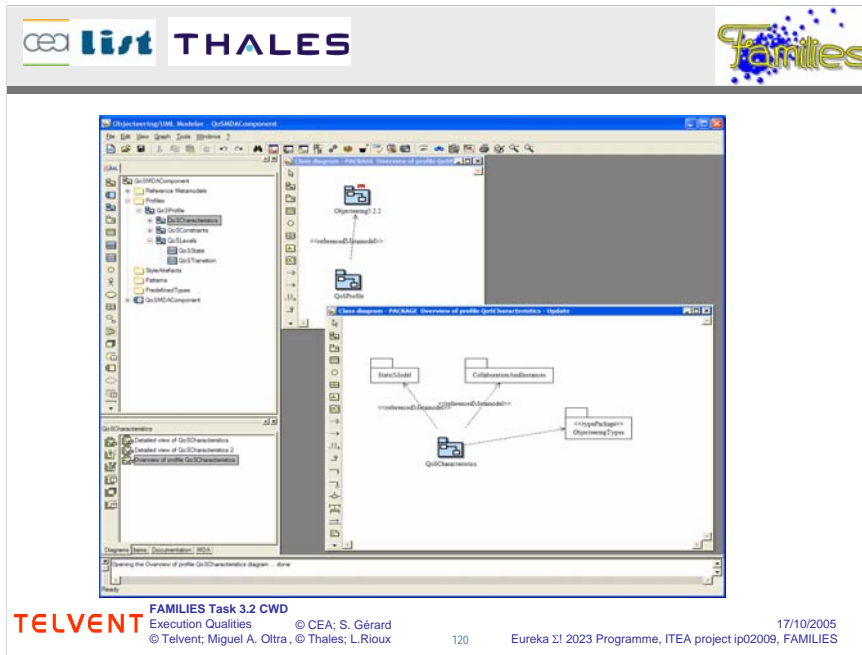
17/10/2005
 Eureka Σ! 2023 Programme, ITEA project ip02009, FAMILIES

RTEQoS Characteristics represents quantifiable characteristics of realtime services. The *RTEQoS Characteristics* are specified independently of the elements that they qualify. *RTEQoS Characteristic* is the constructor for the description of non-functional aspects like: latency, throughput, capacity, scalability, availability, reliability, safety, confidentiality, integrity, error probability, accuracy, ...

RTEQoS Dimension: RTEQoS Dimensions are dimensions for the quantification of *RTEQoS Characteristics*. We can quantify a *RTEQoS Characteristic* in different ways (e.g. absolute values, maximum and minimum values, statistical values). For example, we can quantify the latency of a system function as the end-to-end delay of that function, the mean time of all executions, or the variance of time delay. A *QoS Characteristic* can require more than one type of value for its quantification.

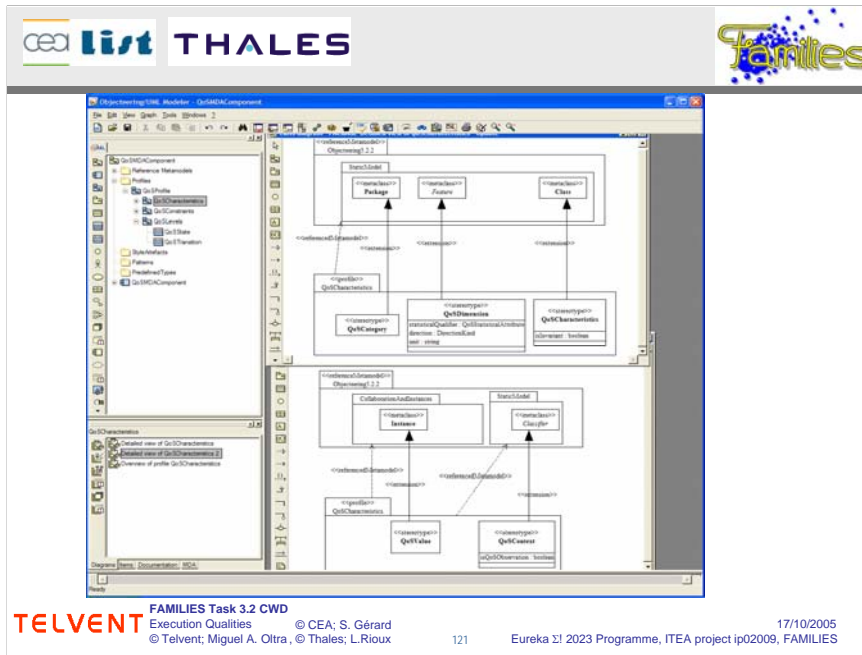
RTEQoS Category: When the number of *RTEQoS Characteristics* is large, or they are especially complex, some mechanisms for grouping are required. Some examples of general groupings of quality attributes are: i) *Performance*: Performance makes reference to the timeliness aspects of how software systems behave. ii) *Dependability*: Dependability is the property of computer systems such that reliance can justifiably be placed on the service it delivers. iii) *Security*: this capability covers different subjects such as the protection of entities, and access to resources.

The main difference between a *RTEQoS Category* and a *RTEQoS Characteristic* is that *RTEQoS Characteristics* are directly quantifiable, and *RTE QoS Categories* does not provide a direct framework for the evaluation of non-functional attributes; it requires a more detailed level of specification to establish constraints or comparisons.



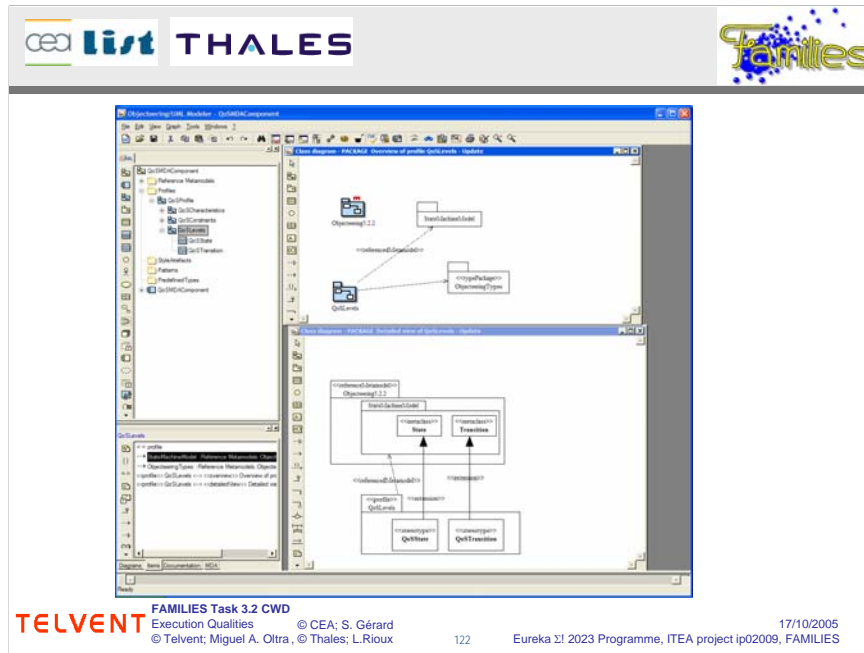
Here it is a snapshot of the definition of the profile base on the standard QoS profile for RTE system families.

The RTEQoS framework supports a general categorization of different kinds of RTEQoS; including RTEQoS that are fixed at design time as well as ones that are managed dynamically. Furthermore it supports the integration of different categories of RTE-QoS for the purpose of modelling RTE-QoS of system aspects. This section includes the metamodells that describe the main packages of this UML extension. The different metamodells establish the elements used to model QoS systems.



The image represents another view of the profile which we have implemented in Objectteering.


QoS Value: *QoS Characteristic* and *QoS Context* provide support for the description of quantifiable QoS values. However, often there are some QoS specific values identifiable at modelling time (e.g. limits of characteristics, or specific QoS values). *QoS Value* instantiate *QoS Characteristic* and fixes it with specific values of its value definitions (*QoS DimensionSlot*). When we attach a *QoS Value* to a model element, we are characterizing the element with quality values.




Here the extension for QoSLevel for RTE systems is presented.

QoS Level: *QoS Level* represents the different modes of QoS that a subsystem can support. Depending on the algorithms or the configurations of the systems, the component can support different working modes, and these working modes provide different qualities for the same services. For each working mode, we specify a *QoS Level*. The *QoS Levels* represent states in the system from a quality point of view. The current *QoS Level* depends on the current resources available, the quality required, and functional parameters such as state variables that identify the current configuration. For each *QoS Level* the resources required are different. In general, the resources offer different quality depending on the load that they have. *Allowed Space* describes the conditions that a software element and the system must achieve to state in a specific *QoS Level*. When a *QoS Level* has more than one *Allowed Space*, the system continues in the *QoS Level* if all *Allowed Space* expressions are true. A *QoS Level Change* occurs when the *Allowed Space* of the current *QoS Level* becomes false, and a transition fires. This change must have one enabled transition from the current *QoS Level* to another that is going to be fired. If there is not an enabled transition, the system is in a state where it cannot achieve its QoS requirements, it will continue in the current state, but it cannot support its contracts. An example of change is when the resources cannot support their contracts (they have received new requests and the resource has a different load level), and we must change the *QoS Level* of some elements. If we cannot change the level, the component will not fulfil its QoS requirements. This change initiates a process of adaptation in the quality of the component. Another source of *QoS Level Change* is the reconfiguration of the component. The component must provide different levels of quality because the user requires a different level of quality at the components with external interfaces. Examples of these changes occur in multimedia systems, when the user resizes some video windows or changes the quality levels of audio and video.

QoS Transition: *QoS Transition* models the allowed transitions between *QoS Levels*. *QoS Level Change* is a type of events that can fire a *QoS Transition*. The architecture and implementation of software elements take into account these states and transitions.



THALES




Conclusion & Outlook

- RTE QoS can be specified within UML models

- Validation of RT systems is then also possible regarding to their RT features
 - Proposition of a method and tool support for performing schedulability analysis
 - Updates of this profile in an Eclipse/Poseidon environment

- Find a way to standardise a profile for real-time and embedded quality of services: UML profile for MARTE
 - UML profile for MARTE RFP at OMG (Modelling and Analysis of Real-Time and Embedded systems) : real-time/05-02-06
 - Initial submission plan by 14 November 2005
 - Final submission plan by June 2006.



FAMILIES Task 3.2 CWD
Execution Qualities
© Telvent; Miguel A. Oltra, © Thales; L.Rioux

123

17/10/2005
Eureka Σ! 2023 Programme, ITEA project ip02009, FAMILIES

Two documents were defined for Families WP3:

- “Introduction of RT-QoS in component interfaces”
- “User guide to validate RT-QoS in UML models”

The first document describes how RTE QoS can be specified in UML models and this is based on the UML2.0 norm.

Validation of RT systems is then also possible regarding to their RT features. The second document presents how the proposed method and tool can be used for performing schedulability analysis of RTES. These profiles, models and tools are actually updated in an open source Eclipse/Poseidon environment.

Last but not least, to standardise a profile for real-time and embedded quality of services within the OMG, a UML profile for MARTE (Modelling and Analysis of Real-Time and Embedded systems) has been proposed as a new RFP at OMG: real-time/05-02-06. The proposed schedule for this RFP is the following:

- Initial submission plan by 14 November 2005.
- Final submission plan by June 2006.



The cover page features the TELVENT logo in orange at the top left, with the tagline 'Shaping a World of Convergence' in orange text to its right. Below this, the text 'Families Task 3.2 CWD' and 'Chapter 7: Resource Usage : dynamically learn & adapt' is displayed. On the right side, there is a graphic of a globe with three circular icons: a factory, a bus, and a server rack. The PHILIPS logo is positioned in the bottom right of the white section. The bottom half of the page has a grey background with the author's name 'M.H.M. Weijenborg' and email 'Marcel.Weijenborg@philips.com' on the left, and the title 'Task 3.2 Execution Qualities' centered. A red horizontal bar is at the very bottom.

Abstract:

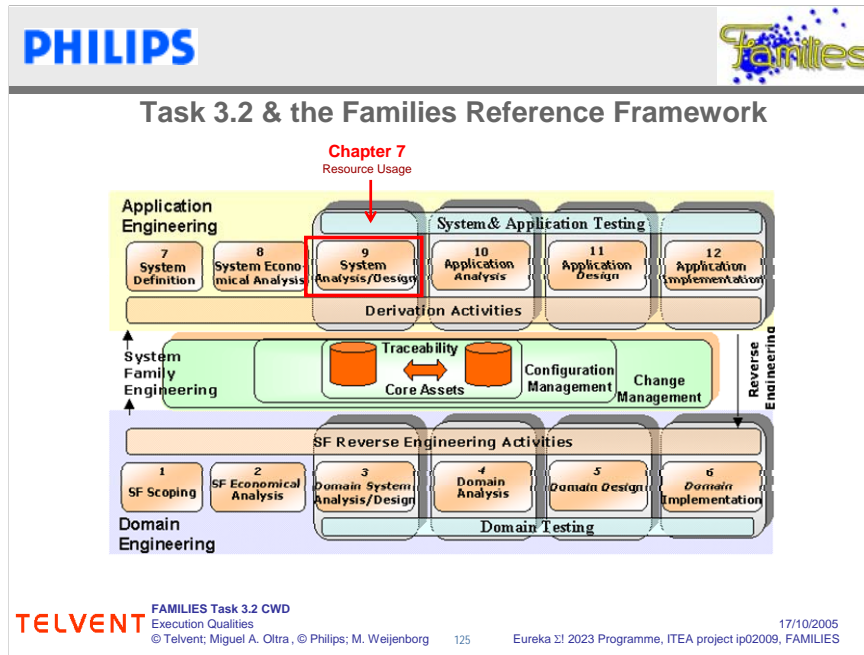
In environments where requirements on resource usage are very relevant, the re-use of components require that those components are aware of their environment and adapt their behaviour accordingly.

We show an infrastructural and architectural approach where layering and combining flexible components can adapt to varying requirements on resource usage, depending on the context where those components are used. The described approach is successfully applied in Philips Medical system families.

Keywords: Resource Usage, Performance, Flexible Architecture, Components, Brokering, Re-use, Execution Qualities


Position in the Families Reference Framework:

This contribution strongly relates to box 9 "System Analysis/Design"



- Chapter 7: Resource usage (Philips)

Chapter 7 is strongly related to problems that can be mapped within the Application Engineering centred on the System Analysis/Design activity.

PHILIPS

Introduction & Problem Description

- **Resource Usage (memory, CPU)**
 - Software components are used in various system constellations
 - Varying resource requirements/restrictions (memory, CPU)
 - Different stages of technology use and mixing thereof
 - We use and mix Java, C++, COM, .NET technologies
 - Integration and wrapping tend to introduce overhead
 - Different technologies have different scaling properties
 - Garbage collection can have nasty dynamic behavior
 - 'new' use of existing components
 - 'new' resource usage profile, not yet anticipated
- **Without proper design attention, reuse of software components can be hindered by improper resource usage in a different context**

TELVENT FAMILIES Task 3.2 CWD
Execution Qualities
© Telvent; Miguel A. Oltra, © Philips; M. Weijnenborg 126



17/10/2005
Eureka Σ! 2023 Programme, ITEA project ip02009, FAMILIES

We make components that are building blocks for large medical systems.

We do not know exactly the context in which these components are used but we know that there are different contexts and these contexts have different requirements both on CPU and memory usage.

The components must try to anticipate such varying usage.

Part of different context is different technologies that contribute to behaviour in terms of CPU and memory usage.



Relevance & Benefits


- **Different use**
 - System integrators have different use cases
 - Different use cases imply different resource restrictions
- **Reuse**
 - Software components that adapt to their context are better and more often reusable
 - More reuse implies leverage of existing knowledge base

TELVENT FAMILIES Task 3.2 CWD
Execution Qualities
© Telvent; Miguel A. Oltra, © Philips; M. Weijnenborg 127

17/10/2005
Eureka Σ! 2023 Programme, ITEA project ip02009, FAMILIES

We aim to design and build components that anticipate their usage context and try to adapt CPU and memory usage accordingly.

When doing this, these components have a more broader usage.

PHILIPS


Approach & Description of Results

The major steps are in determining contexts and use/interact.
Context awareness is handled in three phases:

- **Prepare (static)**
 - Know about various contexts and prepare components for it
- **Learn (dynamic)**
 - Let the components and the context interact and learn
- **Adapt and behave**
 - Components must behave as expected based on the learned context

It turns out that the context awareness concept itself is an iterative cycle where various dimensions are involved:

- **New system integrators**
- **New use cases from existing integrators**
- **New/updated behaviour insights on components**

TELVENT FAMILIES Task 3.2 CWD
 Execution Qualities
 © Telvent; Miguel A. Oltra, © Philips; M. Weijnenborg 128

17/10/2005
 Eureka Σ! 2023 Programme, ITEA project ip02009, FAMILIES

Phase 1-a consist in knowing the kind-of use cases.


Phase 1-b consist in preparing and adapting the components for such use.

Phase 2 consist in determining how to recognise from the context what the intended use is.

Phase 3 consist in to act accordingly to the intended use.

A practical additional result is that the approach nicely fits in a way of incremental design and development.

- While we are busy with design and development, new system integrators join because they see the proof of a working concept. And they bring in new concepts and context.
- Existing system integrators push their limits on use cases and bring in new ones.
- In our model of incremental development, when time progresses, the components get smarter behaviour and everybody benefits from that.

PHILIPS


Approach & Description of Results

- **Context awareness: prepare (static)**
 - Know about various system integrators
 - Know about various resource constraints
 - Know about various use cases
 - Prepare components for it
- **Results:**
 - Overview of various contexts that must be accommodated
 - Components can interact in various low/mid/high-end contexts
 - (low/mid/high contexts typically vary in cpu power, memory size, disk speed, hardware acceleration facilities and database sizes)
 - Different component implementations for equivalent functionality are available for different contexts

TELVENT FAMILIES Task 3.2 CWD
 Execution Qualities
 © Telvent; Miguel A. Oltra, © Philips; M. Weijenborg 129

17/10/2005
 Eureka Σ! 2023 Programme, ITEA project ip02009, FAMILIES


The static part of context awareness is knowing the kind of use cases.

Part is from good thinking (be generic and future proof), part is from asking the customers that are going to use it and part is from inventarisation of typical use cases.

With the information above, the components must be prepared and adapted for such use.

With the known variations in mind, the component design typically can go two ways:

- a) have a component that can accommodate multiple contexts, in a dynamic way (i.e. the if-then-else approach)
- b) for the same functionality, have different component implementations, each having good performance for a typical context. The environment ensures that the proper implementation is used.

PHILIPS


Approach & Description of Results

- **Context awareness: learn (dynamic)**
 - Have initialisation possibilities
 - Probe the environment for optional interfaces
- **Results:**
 - Component layering:
 - Small dedicated components (that expect initialisation)
 - Components that probe the environment and, based on that environment, wire together a network of smaller dedicated components.
 - Collection of components that are flexible and re-usable
 - Not everything can be dynamically recognised by components
 - There is a leftover portion of (static) initialisation information that has to be provided by the system integrator

TELVENT FAMILIES Task 3.2 CWD
 Execution Qualities
 © Telvent; Miguel A. Oltra, © Philips; M. Weijnenborg 130

17/10/2005
 Eureka Σ! 2023 Programme, ITEA project ip02009, FAMILIES

The dynamic part of context awareness is runtime learning.

A component that is prepared for use in various contexts has to learn the context. Two basic learning options are available:

- 1) outside-in: the environment of the component actively provides the component with the required contextual information.
- 2) inside-out: the component itself is learning the context by consulting its environment such as probing for optional interfaces.

In practice it shows that a combination of both is the best choice:

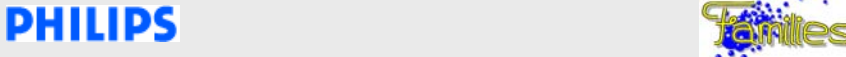
On one side there are various smaller and dedicated components that mostly depend on outside-in learning.

On the other side there are 'smart' (inside-out) higher level components that learn most of their context by themselves and based on that information, select suitable components from the smaller and dedicated components. These selected components are initialised and wired together for the whole functional behaviour with a good resource usage.

An important aspect of learning the context is probing for optional interfaces. Presence/Absence of interfaces can effectively be used for speculating on the actual context.

We cannot always end up with top-level components that are only based on the inside-out approach.

There is a leftover portion of information that has to be provided via the outside-in approach. Most of the time this is more-or-less static data that can be configured by the system integrator in the configuration database.



Approach & Description of Results

- **Context awareness: adapt and behave**
 - Components handle resources differently in different contexts
 - Components can be reused in multiple contexts
- **Results:**
 - Collection of components
 - Flexible components
 - Flexible architecture
 - Easy reuse
 - Easy extend

TELVENT FAMILIES Task 3.2 CWD
Execution Qualities
© Telvent; Miguel A. Oltra, © Philips; M. Weijnenborg 131 Eureka Σ! 2023 Programme, ITEA project ip02009, FAMILIES 17/10/2005


Once a component has obtained its context knowledge, it is simply a matter of recognising the intended use and act accordingly.

We now have a collection of low-level components that are suited for particular tasks with a particular resource profile.

We have higher-level components that learn from their context and dynamically select suitable components from the low-level components and wire them together for the intended total-result.

The approach not only gives us a lot of flexible components but also a flexible architecture that allows for easy prototyping and implementing alternative approaches.


This is because of the existing variation points where easily new variations can be added or swapped.

PHILIPS


Approach & Description of Results

Component Broker

- **Why:**
 - Have runtime flexibility of obtaining function-for-context
 - Avoid static design (if-then-else)
 - Avoid compile time dependency
 - Allow additional flexibility by system integrators
 - In a layered component design, the middle layers need not be aware of the details
- **How:**
 - Use the runtime 'reflection' features in java and .NET
 - Broker configuration via flexible text file
 - ContextID + FunctionID get translated by broker into actual class instantiation
 - Need standardised interfaces (abstract functionality)



FAMILIES Task 3.2 CWD
Execution Qualities
© Telvent; Miguel A. Oltra, © Philips; M. Weijenborg

17/10/2005
Eureka Σ! 2023 Programme, ITEA project ip02009, FAMILIES

For those high-level components that select and wire together various low-level components based on contextual information, you want to avoid too much hard coding of that knowledge:

- the if-then-else approach statically limits your flexibility
- hard coding implies compile time dependency

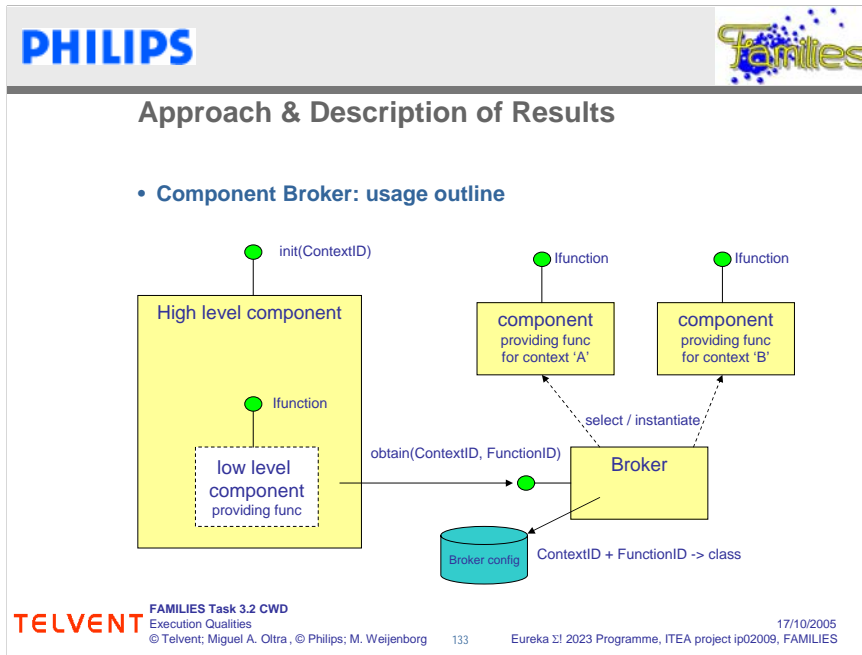
Instead we choose the concept of (ContextID + FunctionID) to identify a piece of predefined functionality. The functionality is predefined via mutually agreed interfaces.

The actual implementation class is not relevant for the caller. The broker instantiates a class that offers the required interfaces.

We use the runtime 'reflection' capabilities of Java and .NET for choosing the actual implementation class (the broker is doing that).


The broker configuration is a straightforward text file that can easily be adapted.

This approach gives flexibility and opportunity of choice very late: even the system integrator can replace low-level components on his own without any recompilation of the high-level components.



The high-level component either implies a context or finds out its context from its environment.

When the high-level component needs functionality, via a predefined interface, from a low-level component that is suitable for the context, then the broker is used for obtaining a class instance that offers that functionality suitable for the context.

PHILIPS

Approach & Description of Results

- **Pitfalls**
 - Sometimes the generics of components hinder in squeezing performance and resource usage
 - Testing needs special attention: each variation point inside a component adds to the overall count of system operating configurations
 - Documentation is important: without proper documentation of the variation points, the oversight can be lost fairly quickly

TELVENT FAMILIES Task 3.2 CWD
Execution Qualities
© Telvent; Miguel A. Oltra, © Philips; M. Weijenborg

134


17/10/2005
Eureka Σ! 2023 Programme, ITEA project ip02009, FAMILIES

As it is often the case: each advantage has its downside.

Sometimes there are specific situations where there exists an opportunity for additional performance gain. But sometimes the generic and flexible approach of the component implementation prevents quick implementation of that opportunity.


Each variation point inside a component represents an alternative code path. In a total system configuration this potentially multiplies the number of ways in which components interact. If we are not careful, many combined code paths go untested. Therefore extra attention is needed for testing, such that there will be good confidence in the proper overall operation of various system constellations with interacting components.

When a component grows from the initial one-problem->one-solution in a flexible self-adapting entity, documentation of the variation points is very important. Failing to do so can easily result in loss of overview. This hinders both regular maintenance and later adding of more flexibility.

PHILIPS

Conclusion & Outlook

- The concept of adaptive behaviour works
- The described approach is successfully applied in Philips Medical product families
- New system integrators start integrating the components
- Adding support for new use cases and new contexts have a cross-usage benefit by others
- Sometimes the generics of components hinder in squeezing performance and resource usage
- With various regular cases in place, now the more extreme use cases need to be accommodated

 FAMILIES Task 3.2 CWD
Execution Qualities
© Telvent; Miguel A. Oltra, © Philips; M. Weijnenborg

17/10/2005

Eureka Σ! 2023 Programme, ITEA project ip02009, FAMILIES

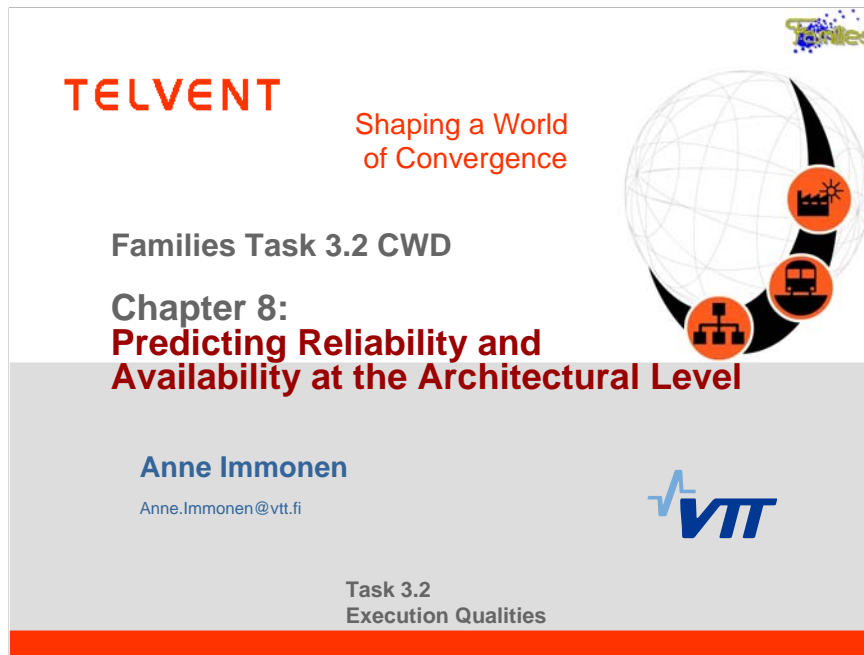
Various system integrators did apply the software components. The components work satisfactory in various contexts.

Because actual results can be shown, new system integrators are easier to convince in using the components.

New system integrators add new types of usage. When components are upgraded to suit those needs then other system integrators automatically benefit from this.

Because the software components are used in various contexts, some degree of generics is unavoidable (in fact most of the times desired). However sometimes the generics hinder in getting full optimal performance and resource usage profiles.

Many regular use cases from various system integrators are now taken into account. Now comes the challenge to accommodate the more extreme use cases.

**Abstract:**

The contribution of this work is the RAP (Reliability and Availability Prediction) method that aims to improve reliability and availability of a system family. The RAP method is an integral part of Quality-driven Architecture Design and quality Analysis method (QADA[®]), extending it by providing needed activities to support reliability and availability. The RAP method provides guidelines on how the reliability and availability requirements should be mapped to architecture, how they should be represented in architecture, and how the architecture should be analyzed to validate whether or not the requirements are met.

Keywords: Reliability, availability, predictive analysis, execution quality, evaluation

Relation to other WPs and tasks:

Closely related with work done in tasks 3.3 and 4.3 by VTT.

Acronyms:

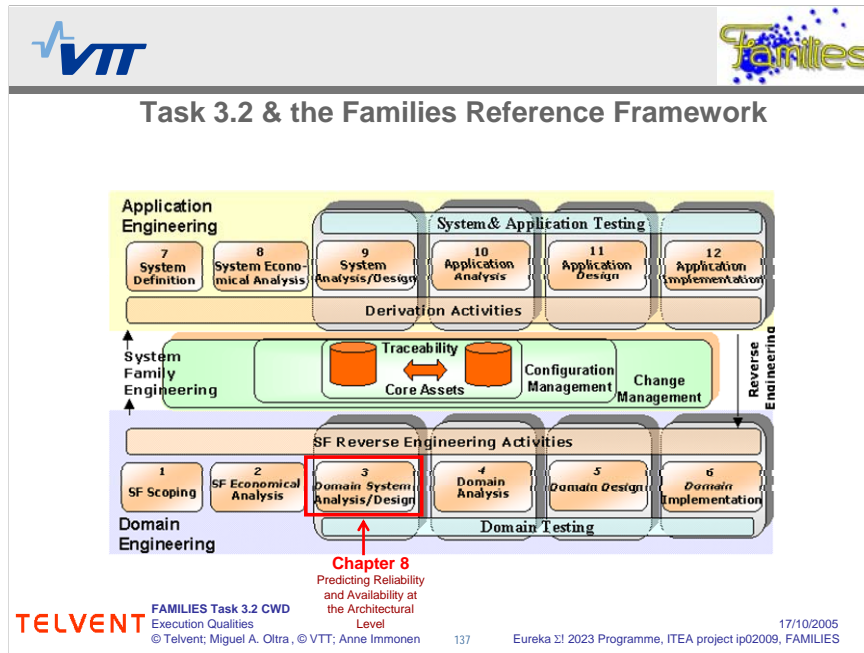
| | |
|------|---------------------------------------------------------|
| QADA | Quality-driven Architecture Design and quality Analysis |
| R&A | Reliability and Availability |
| RAP | Reliability and Availability Prediction |

References:

- [1] Matinlassi, M., Niemelä, E. and Dobrica, L. (2002). Quality-driven architecture design and quality analysis method, A revolutionary initiation approach to a product line architecture. Espoo, VTT Technical Research Centre of Finland: 129 p.
- [2] Purhonen, A., Niemelä, E. and Matinlassi, M. (2004). "Viewpoints of DSP Software and Service Architectures." Journal of Systems and Software **Vol. 69**(No. 1-2): Pp. 57-73.
- [3] Immonen, A. and Niskanen, A. 2005. A Tool for Reliability and Availability Prediction. Submitted to Euromicro 2005.
- [4] Immonen, A. 2005. A method for predicting reliability and availability at the architectural level. Submitted to the Families technical book, Timo Käkölä and Juan Carlos Dueñas (Eds.) (In progress)

Links:



<http://www.vtt.fi/ele/research/soh/projects/families/index.htm>
<http://www.vtt.fi/ele/research/soh/projects/qada/index.htm>



•Chapter 8: Predicting Reliability and Availability at the Architectural Level (VTT)

The chapter is related to activity 3 of the Families Reference Framework: Domain System Analysis/Design. The contribution of the chapter is RAP (Reliability and Availability Prediction) method which objective is to predict reliability and availability of a *system family* from the architectural models. The RAP method fills the gap from requirements negotiation to R&A analysis, providing required tool and notation extensions, techniques and guidelines for R&A prediction at the architectural level. The method consists of three phases: 1) Defining reliability and availability goals, 2) Representing reliability and availability in architectural models, and 3) Reliability and availability evaluation. Each phase includes a set of steps that further consist of activities.

The RAP method is directed especially to system families, but it can be applied to individual systems as well. The main benefits of the method are that reliability and availability (R&A) of system family and systems increase, the maintenance costs decrease and less resources, modifications and fault repairs are needed. Also, risks are lower as it can be ensured that architecture meets the requirements.





Introduction & Problem Description

- **In the near future, systems will be more complicated and more tightly embedded into our surroundings**
- **These systems have to be reliable and available, in other words, they have to**
 - work as intended
 - be available when needed
- **In the context of system families, reliability and availability are extremely important, because**
 - **faults** cause extensive and long-term problems involving all the family members
 - **choice of architectural style** has a great influence within system families
- **The traditional reliability and availability analysis is not applicable and sufficient for today's complex systems**

TELVENT FAMILIES Task 3.2 CWD
Execution Qualities
© Telvent; Miguel A. Oltra, © VTT; Anne Immonen 138 Eureka Σ! 2023 Programme, ITEA project ip02009, FAMILIES 17/10/2005

Reliability and availability (R&A) are execution qualities that can be observed from the behaviour of the system at run-time. Reliability is the probability of failure-free operation of a software system for a specified period of time in a specified environment. Availability is the probability of a software service or system being available when needed.

Faults and low availability of our surrounding systems can cause serious problems and extensive damage, for example, financial losses and even danger to human life. R&A of the system can be guaranteed if it can be ensured that the system meets its requirements. Traditional R&A analysis commonly requires a lot of time and resources. The complexity and the larger scale requirements of today's systems constrain the use of the traditional reliability and availability analysis.





Relevance & Benefits

- **Architecture is the first asset that describes the system family as a whole**
- **A method is needed to predict reliability and availability from the architectural models**
- **Benefits of the predictive method:**
 - Reliability and availability of system family members increase
 - Quality of components from different component suppliers can be validated and proved in the context of system family
 - Maintenance costs of reliable systems are lower and less modifications and fault repairs are needed
 - Risks are lower as it can be ensured that architecture meets the requirements
 - Reduced amount of resources, such as money, specialists and time, are required

TELVENT FAMILIES Task 3.2 CWD
Execution Qualities
© Telvent; Miguel A. Oltra, © VTT; Anne Immonen 139 Eureka Σ! 2023 Programme, ITEA project ip02009, FAMILIES 17/10/2005

The reliability and availability should be able to be analysed already from the architectural models when the modifications are easier and the architectural decisions can still be affected. The predictive reliability and availability analysis method saves resources, time and money, and decreases risks and maintenance costs. The existing prediction methods have several shortcomings; therefore a new method is required.



Approach & Description of Results

- The result of this task is a RAP (reliability and availability prediction) method
- The RAP method is an integrated part of Quality-driven Architecture Design and quality Analysis (QADA ®) methodology
- The objective of the RAP method is to predict reliability and availability of system family
- The predictive reliability and availability analysis also requires consideration of earlier development phases

RAP method consists of following phases:

1. Defining reliability and availability goals
2. Representing reliability and availability requirements in architectural models
3. Evaluating reliability and availability

© = Registered trademark of VTT Technical Research Centre of Finland.
FAMILIES Task 3.2 CWD
Execution Qualities
© Telvent; Miguel A. Oltra, © VTT; Anne Immonen 140 Eureka Σ! 2023 Programme, ITEA project ip02009, FAMILIES 17/10/2005

The main result of this task is RAP (reliability and availability prediction) method that is an integrated part of the QADA ® methodology [1], [2], and therefore it obtains the same principles as QADA.



QADA bases on the following principles:

- family-oriented product development
- quality-driven architecture design
- reuse of existing artifacts and knowledge
- quality evaluation based on architectural models

The main activities of QADA are:

- requirements engineering
- architectural modeling
- quality analysis

The RAP method extends these activities with reliability and availability related aspects. In summary, the RAP method consists of three phases, which further consist of different steps.





Approach & Description of Results

Phase 1: Defining reliability and availability goals

- **The objective of the first phase is to define how to elicit reliability and availability (R&A) goals and how to bring R&A requirements into design**
- **The phase includes the following steps:**
 - Identifying stakeholders and their concerns
 - Refining quality requirements
 - Mapping R&A requirements to functionality
 - Selecting an architectural style and doing the trade-off analysis
 - Defining criteria for R&A evaluation

TELVENT FAMILIES Task 3.2 CWD
Execution Qualities
© Telvent; Miguel A. Oltra, © VTT; Anne Immonen 141 Eureka Σ! 2023 Programme, ITEA project ip02009, FAMILIES 17/10/2005

The first phase of RAP method is about gathering and negotiating reliability and availability requirements in a way that the best possible requirements set can be identified. The applicable methods and frameworks are used for eliciting quality requirements.



Approach & Description of Results

Phase 2: Representing R&A requirements in architectural models



- **The objective of this phase is to define how to describe architecture in a way that R&A can be analysed from the architectural models**
- **This phase includes the following steps:**
 - Mapping required R&A to conceptual architectural elements
 - Mapping from conceptual to the concrete architecture
 - Mapping provided R&A to concrete architectural elements

TELVENT FAMILIES Task 3.2 CWD
Execution Qualities
© Telvent; Miguel A. Oltra, © VTT; Anne Immonen 142 Eureka Σ! 2023 Programme, ITEA project ip02009, FAMILIES 17/10/2005

Different sets of quality concerns can be transformed by the architectural design process into different architectural decisions. The architecture design is the first phase in the system development where reliability and availability can be analysed. The architectural models must be represented in a way that enables the analysis directly from the models.

The different architectural decisions have a great influence on how the reliability and availability requirements are met. The second phase of the RAP method is about defining and selecting suitable design methods and techniques to ensure that reliability and availability requirements are taken into account in the architecture. Also, the traceability of requirements must be guaranteed.

The guidelines of QADA are used in architecture modelling. QADA describes architecture on two abstraction levels: conceptual and concrete. Both levels consist of four viewpoints: structural, behavioural, deployment and development.



Approach & Description of Results



Phase 3: Evaluating reliability and availability

- **The objective of this phase is to define how to validate that the R&A goals met in the architecture**
- **The phase consists of the following steps:**
 - Quantitative analysis
 - bases on failure behavior, interaction and behavior of components
 - applies computational methods for calculating reliability and availability
 - Qualitative analysis
 - relies on documented design rationale and heuristics
 - examines the architectural decisions
 - Decision making
 - bases on results of quantitative and qualitative analysis

TELVENT FAMILIES Task 3.2 CWD
Execution Qualities
© Telvent; Miguel A. Oltra, © VTT; Anne Immonen 143 Eureka Σ! 2023 Programme, ITEA project ip02009, FAMILIES 17/10/2005

The third phase of the RAP method is about analysing reliability and availability from the architectural models. The analysis consists of both quantitative and qualitative analysis. The quantitative analysis bases on numeric calculations, whereas the qualitative analysis bases mostly on human knowledge.

The tool support for the RAP method has been developed [3]. The tool assists in phases 2 and 3 of the RAP method. In phase 2, the tool enables the attachment of required and provided reliability and availability to the architectural models. In phase 3, the tool assists in quantitative analysis.



Conclusion & Outlook

- The main purpose of the RAP method is to predict reliability and availability from the architectural models, before actual system implementation
- The RAP method fills the gap from requirements negotiation to R&A analysis, providing required tool and notation extensions, techniques and guidelines for R&A prediction at the architectural level
- The RAP method has been validated by applying it to a distribution platform of a system family

TELVENT FAMILIES Task 3.2 CWD
Execution Qualities
© Telvent; Miguel A. Oltra, © VTT; Anne Immonen 144 Eureka Σ! 2023 Programme, ITEA project ip02009, FAMILIES 17/10/2005

The RAP method has been applied to a distribution platform of a system family of software systems that is formed by executing units in a networked environment. The system family includes variants of the platform products for three end user applications; a game, a health care application and a emergency intervention system. There exist variations within the reliability and availability requirements between the different family members and also in the importance of those quality attributes.

Reliability and availability prediction requires refinements to software development phases from requirements gathering to architectural analysis. The objective of this contribution is to define RAP (Reliability and Availability Prediction) method that assists in reliability and availability requirements engineering, architecture modelling and reliability and availability analysis from the architectural models.

The RAP method will be published as a book chapter in the Families Technical Book (eds. Timo Käkölä and Juan Carlos Dueñas), with the title: "A method for predicting reliability and availability at architectural level" [4]. The book is still in progress.



Shaping a World
of Convergence




Families Task 3.2 CWD
Execution Qualities

Conclusion

Miguel A. Oltra
miguel.oltra@telvent.abengoa.com



Task 3.2
Execution Qualities



Conclusion

- **Run-time system quality attributes must be taken into account during system architecture design phases**
 - How to embed QoS on the system family architecture provided documentation
 - How to provide mechanisms and techniques in order to consider system quality attributes on the early stages of the system family design
 - How to present architectural patterns that best reflect different -ilities
 - How to evaluate and measure system quality attributes
- **Two sections have been presented covering and trying to answer some of these topics**
 - System Family security attributes
 - Other run-time quality attributes


TELVENT FAMILIES Task 3.2 CWD
Execution Qualities
© Telvent; Miguel A. Oltra

146 Eureka Σ! 2023 Programme, ITEA project ip02009, FAMILIES 17/10/2005

When dealing with system families architectures considerations about non-functional aspects have to be considered. These non-functional aspects, must be satisfied during system operation (on run-time). So it is of mainly importance to consider these run-time system quality attributes during system architecture design phases. Contributions included in this CWD had deal with some of the high amount of -ilities that may restrict the design of a system, by trying to response to one or several of the following questions:

- How to embed QoS on the system family architecture provided documentation?
- How to provide mechanisms and techniques in order to consider system quality attributes on the early stages of the system family design?
- How to present architectural patterns that best reflect different -ilities?
- How to evaluate and measure system quality attributes?

The task 3.2 CWD has been divided in two main sections organised in chapters. These sections has tried to answer some of the previously mentioned topics. Section I deals with system family security attributes (this is a big topic, and the several chapters covers only a small part or subset of it). Section II deals with other run-time quality attributes; such as availability, reliability and performance.



Conclusion (Cont.)

- **System design taken into account security aspects concerns may become a big and difficult problem to afford**
- **Chapters of section I present different mature approaches covering security aspects focused on a subset of topics related to architectural system design**
 - Modelling (by means of documentation) and evaluation techniques of quality attributes
 - Security life cycle development based on standards (variation points, elements, scenarios, security model, ... are identified and presented)
 - A decision support framework forming a reference architecture for security. Decisions based on architectural tactics and patterns will support the design of the system architecture
 - Security reference models for architecting based on the mapping of security countermeasures identified from standards
 - A process for response improvement to incident reports that may occur in a product family engineering organisation related to security aspects by performing assessments both at PMS level and Business Units level

TELVENT FAMILIES Task 3.2 CWD
Execution Qualities
© Telvent; Miguel A. Ojtra

17/10/2005
Eureka Σ! 2023 Programme, ITEA project ip02009, FAMILIES

In this conclusion slide is presented a resume of the main conclusions achieved in each chapter.


Chapter 1 main results may be summarised in the following sentence: the modelling and evaluation techniques of quality attributes must be embedded in the design phases by means of provided documentation. The proposed techniques have been employed for represent security aspects of a system architecture.

In chapter 2 as a main result, has been provided a security life cycle development based on standards. In this chapter, variation points, elements, scenarios, a security model, ... have been identified and presented. All these topics must be considered during the development life cycle, specially when the life cycle has to consider security aspects that the final system must guarantee.

Chapter 3 provides a decision support framework forming a reference architecture for security. Decisions based on architectural tactics and patterns will be a support in the design activities of the system architecture for system architects.

Moreover, in Chapter 4 has been defined a security reference model for architecting based on the mapping of security countermeasures identified from well known standards.

Finally in Chapter 5, a process for response improvement to incident reports that may occur in a product family engineering organisation related to security aspects by performing assessments both at PMS level and Business Units level has been developed.



Conclusion (Cont.)

- **Not only security has been considered in task 3.2 but also, other quality attributes are covered in Section II**
- **RTE QoS can be specified within UML models**
 - Validation of RT systems can be realised by a method and tool support for performing schedulability analysis of its features
- **Flexible architecture based on software components that adapt to multiple contexts are developed by system integrators**
 - Generic components hinder resource usage and performance improvement
- **A method (RAP) for predicting reliability and availability based on pre-defined goals and architectural models by means of techniques for analysing the family architecture before implementation**
 - Fills the gap from requirements negotiation to reliability and availability analysis, providing tools, notation extensions, techniques and guidelines for their prediction at architectural level

TELVENT FAMILIES Task 3.2 CWD
Execution Qualities
© Telvent; Miguel A. Oltra

17/10/2005
Eureka Σ! 2023 Programme, ITEA project ip02009, FAMILIES

To finalise, the main results from section II chapters are summarised in this slide. Among the problems and objectives covered in section II chapters are issues related to run-time systems quality aspects like availability, performance and reliability. In concrete in chapter 6 the problem of how to specified run-time embedded QoS by means of UML models has been accomplished. The use of an UML profile (standardised by the OMG) embedded systems in order to represent RTE QoS is a possible solution to this problem. Moreover a validation of the approach being followed in this chapter is of great importance. The validation has been realised by a method and tool support for performing schedulability analysis of run-time systems features.

Flexible architectures are required for assets reuse when deploying components in similar environments. Generic components hinder quality aspects of the system architecture like resource usage and performance. How to deal with this problem is the purpose of chapter 7; where a flexible architecture based on software components that adapt to multiple contexts in order to improve performance and resource usage has been proposed.

Finally in chapter 8, a method (called RAP) for predicting reliability and availability based on a set of pre-defined goals and architectural models and by means of techniques for analysing the family architecture before implementation has been proposed. This method tries to fill the existing gap that usually appears in the early system design activities. This gap appears from requirements negotiation to reliability and availability analysis, and by providing tools, notation extensions, techniques and guidelines for their prediction at architectural level, where these system quality attributes might be taken into account in the early phases of the system design.



Shaping a World
of Convergence



Families Task 3.2 CWD

Execution Qualities

The Authors' Picture Gallery

Miguel A. Oltra
miguel.oltra@telvent.abengoa.com



Task 3.2
Execution Qualities



The Authors' Picture Gallery

Task Leader



Miguel A. Oltra
Partner: Telvent
E-mail: miguel.oltra@telvent.abengoa.com

Section I Editor



Tor Faegri
Partner: ICT-Norway
E-mail: tor.e.fagri@sintef.no

Section II Editor



Anne Immonen
Partner: VTT
E-mail: anne.immonen@vtt.fi

The Authors' Picture Gallery (cont.)

| | | | |
|------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------|
|  |  |  | |
| <p>Ivar Sandstad Partner: ICT-Norway E-mail: ivar@superoffice.com</p> | <p>Jens Glattetre Partner: ICT-Norway E-mail: jens.glattetre@superoffice.com</p> | <p>Svein Hallsteinsen Partner: ICT-Norway E-mail: Svein.Hallsteinsen@sintef.no</p> | |
|  |  |  |  |
| <p>Juha Savolainen Partner: Nokia E-mail: juha.savolainen@nokia.com</p> | <p>Laurent Rioux Partner: Thales E-mail: laurent.rioux@thalesgroup.com</p> | <p>Sebastien Gerard Partner: CEA E-mail: sebastien.gerard@cea.fr</p> | <p>Hubert Dubois Partner: CEA E-mail: hubert.dubois@cea.fr</p> |

TELVENT FAMILIES Task 3.2 CWD
Execution Qualities
© Telvent; Miguel A. Oltra

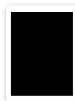
17/10/2005
Eureka Σ! 2023 Programme, ITEA project ip02009, FAMILIES



The Authors' Picture Gallery (cont.)



José Luis Arciniegas
Partner: UPM
E-mail:
jlarci@dit.upm.es



José Luis Ruiz
Partner: UPM
E-mail:
jlruiz@dit.upm.es



Juan Carlos Dueñas
Partner: UPM
E-mail:
jcduenas@dit.upm.es



Rodrigo Cerón
Partner: UPM
E-mail:
rceron@dit.upm.es



Chris Broerse
Partner: Philips
E-mail:
chris.broerse@philips.com



Marcel Weijenborg
Partner: Philips
E-mail:
marcel.weijenborg@philips.com