Requirements Engineering for Trusted Embedded Systems

José L. Fernández
Industrial Engineering School
Madrid Technical University (UPM)
Jose Gutierrez Abascal 2, 28006 Madrid (Spain)
+34 913363144
jose.fernandez@incose.org

Rubén Alonso and Francisco Gómez
Visual Tools
Isla Graciosa 1, 28034 Madrid (Spain)
+34 91 7294844
ralonso@visual-tools.com; fgomez@visual-tools.com

Christophe Jouvray
Trialog
25 rue du Général Foy, 75008 Paris (France)
+33 144 706 106
christophe.jouvray@trialog.com

Yann Rouxel
EADS DS
1 boulevard Jean Moulin, 78990 Elancourt (France)
+33 1 6138-5316
yann.rouxel@eads.com

Antonio Pérez
Ikerlan
Pº J.M. Arizmendiarrieta 2, 20500 Arrasate-Mondragon, Guipuzkoa (Spain)
+34 943712400
APerez@ikerlan.es

Abstract. Trusted computing is a complex and multidimensional issue, and its application to embedded systems has some particularities not found in business and desktop computing. Trust

for embedded systems differs from traditional trust because these systems have physical accessibility, resource constraints and real-time requirements.

In this paper, we describe the approach followed to specify trust requirements for the ITEA2 project TECOM (Trusted Embedded Computing)[1]. Using these requirements, the project partners developed embedded systems in diverse areas such as home control, video surveillance, digital content delivery, automotive, and communications gateways for remote maintenance.

The approach for the engineering of trust requirements presented here is based on two principles: a common conceptual model to identify the trust concepts for embedded systems and their relationships, and a well-defined process to elicit and specify embedded systems requirements.

Trusted Embedded Systems

Trusted computing is a complex issue and its application to embedded systems has some particularities not found in other application domains.

The ITEA Trust4All project describes the properties of trust (Muskens 2005):

- Trust is directed. It is an oriented relationship between the trustor and the trustee.
- Trust is subjective. It is based on a person's confidence and opinions.
- Trust is context-dependent. It is related to a system in a given context.
- Trust is measurable. Levels or values of trust can be computed.
- Trust is not necessarily symmetrical. A may trust B, but this does not necessarily imply that B will trust A.
- Trust can evolve. It is not a permanent quality, so its value can change over time.
- Trust can be history-dependent. The present level of trust may be affected by previous experience.
- Trust is required when there is a lack of information. If we know everything, there is no need for trust, but trust is necessary when we cannot obtain complete information about the trustee.
- Trust is transferable. Information about trust can be communicated among entities.
- Trust can be a composite property. In TECOM, integrity, availability, safety, reliability, security and survivability are considered as trust dimensions.
- Trust is conditional transitive. Trust is not perfectly transitive in the mathematical sense.

Frequently embedded systems are easily accessible at physical layer. In recent years, this accessibility has led to new types of security attacks in the areas of physical tampering and side-channel analysis. A key issue related to physical accessibility and portability is privacy, particularly in networked embedded systems.

Storing data on an embedded system creates privacy concerns. On the one hand, storing sensitive information on a single embedded system rather than on multiple servers minimizes the number

[1] TECOM is the ITEA2 project 06038 (2007-2010), www.tecom-itea.org

of locations where an attack can occur. On the other hand, small embedded systems can be easily lost or stolen, so they must have extra security measures built-in to ensure that private data cannot be compromised (Hwang 2006).

Embedded systems have resource constraints in terms of memory, computational capacity and energy, a situation which can pose several risks. In terms of memory, sophisticated public key cryptography techniques might be infeasible for embedded systems. An energy intensive trust mechanism can cause an embedded system to perish from battery exhaustion before it can perform useful work. Thus by seeking to drain the embedded system battery, an attacker can cause system failure even if breaking into system is impossible.

Unlike transaction-oriented business computing, embedded systems often perform periodic computations with real-time deadlines. When a delay of only milliseconds can cause system failure, the embedded system becomes vulnerable to attacks designed to disrupt system timing. It is important to ensure that real-time deadlines will be met, even in the face of “Denial of Service” attacks or compromised system parts.

In this paper, we describe the approach followed to engineer trust requirements for the ITEA2 project TECOM (Trusted Embedded Computing). Using these requirements, the project partners developed embedded systems in diverse areas such as home control, video surveillance, digital content delivery, automotive, and communications gateways for remote maintenance.

The approach for the engineering of trust requirements presented here is based on two principles: a common conceptual model to identify the trust concepts for embedded systems and their relationships, and a well-defined process to elicit and specify embedded systems requirements.

Trust Conceptual Model

The trust conceptual model describes the concepts underpinning trust in embedded systems by means of a UML class diagram. The conceptual model presented here is the basis for the specification of requirements for TECOM systems. The conceptual model represents the trust concepts and their relationships while avoiding the ambiguities relative to similar concepts.

The conceptual model is represented in Figure 1. It uses the fault⇒error⇒failure paradigm, frequently called AVI fault model (Avizienis 2001), and extends it with new concepts, e.g. trust, trust dimension, requirement, design selection, and system.

The concepts and relationships represented in the conceptual model class diagram of Figure 1 are:

- **Fault:** The adjudged or hypothesized cause of an error. Faults may be classified according to several criteria. Here they are classified as accidental or intentional. Accidental faults can arise during either system development or operation. During development accidental faults result from a bad design. During operation they may be produced by the violation of an operating or maintenance procedure. Intentional faults fall into three classes: malware, physical attacks, and intrusions. Typically an attack exploits vulnerabilities (faults that can be exploited) of an asset to cause an intrusion. Attacks may use physical means to exploit vulnerabilities, such as power fluctuations, radiation, and wire-tapping. (Powell 2003).
- **Error:** The part of the system state that may lead to a failure (Avizienis 2001). An error is detected if its presence in the system is indicated by an error message or an error signal that

originates within the system. Detected errors contribute to trust risk. Errors that are present but not detected are latent errors.

- **Failure:** The inability of the system or some of its parts to meet its specifications (functional and non-functional requirements). Failures considered in this conceptual model are breakdown, denial of service and malfunction. Failures are related to the system.
- **Asset:** Anything of value that should be protected from malicious harm. An asset is a specialization of a system part. With regard to TECOM, an asset requires protection because it is the potential target of attack. In security engineering, the emphasis tends to be on data assets (e.g., integrity and privacy), but trust also includes software assets (e.g., integrity) and services (e.g., theft and denial of services). Physical security also deals with protecting people and property, including hardware and facilities. Physical security is a very important issue for embedded systems.
- **Trust:** The trust concept has several meanings in the diverse projects dealing with trust. Here the definition of the ITEA Trust4all project (Muskens 2005) is used: "Trust is the degree to which a trustor has a justifiable belief that the trustee will provide the expected function or service." Trust may have a trust level.
- **Trust Dimensions:** Several dimensions of competence trust in the context of Internet can be identified (Schneider 1998). These dimensions can be aggregated in the so-called dependability quality factor (Avizienis 2001). From the conceptual model point of view, the result is very similar. The following trust dimensions are taken into consideration:
    - **Integrity:** The absence of improper, intentional, or accidental system alterations. Integrity here is a more general concept that the integrity concept of security that considers only intentional faults.
    - **Availability:** Ensures that the system continues to operate in the face of certain anticipated failures.
    - **Reliability:** The capability of a system to perform consistently and precisely what it is expected to do.
    - **Security:** Ensures that the system resists intentional faults. Security is a complex and important dimension also contains diverse factors, for example access control, attack detection, security integrity, freshness, physical protection, privacy and other factors explained below.
    - **Survivability:** The capability to provide a level of functionality or service in adverse or hostile conditions.
    - **Safety:** The absence of catastrophic consequences on the user(s) and the environment.
- **Trust Policy:** A quality policy that mandates a system-specific quality criterion for trust or one of its dimensions. System-specific quality criteria can also involve the system's environment, the infrastructure in which it exists, and any assumptions about the system.
- **Trust Risk:** The potential risk of system failure due to errors. Trust risk is the sum (over all relevant errors) of the negative impact of the failure (i.e. its criticality) multiplied by the likelihood of the failure occurring.

- **Requirement:** The specification of a required amount of trust (i.e. a dimension of trust) in terms of a system-specific criterion and a minimum level of an associated quality metrics that is necessary to meet one or more trust policies.
- **Design selection:** A design decision that helps fulfill one or more trust requirements and/or reduces one or more system faults. Trust can be implemented as some combination of hardware or software parts, manual procedures, and services provided by either the application or the execution platforms. Design selection may be at different levels: protocol level which includes the design of protocols to be performed on embedded devices; algorithm level consisting of the design of cryptographic primitives and application-specific algorithms; architecture level consisting of hardware/software partitioning, execution platform features and architecture tactics to prevent, tolerate or remove faults; hardware element level dealing with the hardware design of the modules (the processors and coprocessors) required; and circuit level implementing transistor level and package-level techniques to thwart various physical layer attacks.
- **System:** A combination of interacting parts organized to achieve one or more stated purposes.
- **Part:** A building element of the system. Parts may be composite parts (e.g. a subsystem) containing other parts. Parts depend on other parts. High levels of dependency, also described as high coupling between parts, are error propagation enablers.

The conceptual model also represents some important relationships between the concepts by various lines and symbols. The following symbols represent relationships in the UML notation. Associations that signify the relationship between two concepts that need to know each other: a line. Composition that is the relationship between an element and its parts: a black diamond. Specialization that is used when a concept is specialized in more specific ones: a hollow triangle.

The design selections to be considered in trusted systems are not represented in the conceptual model. This is part of the architecture phase of embedded system development, and for this reason the design selection concept is not expanded in the model represented in Figure 1.

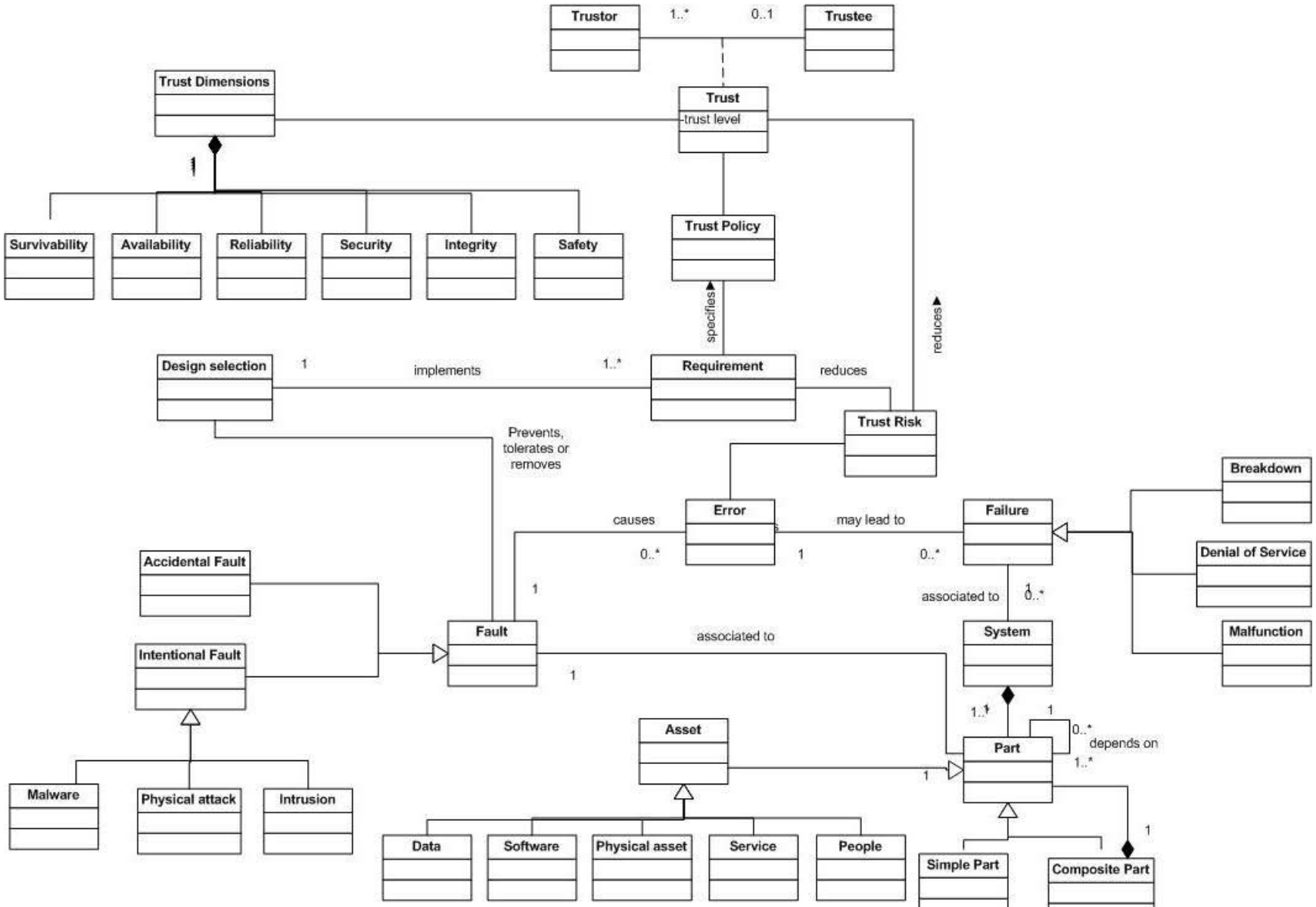

Figure 1. Conceptual model for trust in embedded systems

Requirements Engineering Process

In this section, the requirements engineering process defined for the TECOM trusted systems specification is described, highlighting the identification and classification of the non-functional requirements related to trust and its dimensions. This process, seen in Figure 2, encompasses requirements derived from real-life scenarios, as well as requirements based on the trust conceptual model presented in Figure 1.

The requirements capture technique used is based on the “scenario” paradigm. The capture and specification of scenario-based requirements in applications related to trust is a recommended practice since this method is currently applied in some aerospace, automotive, and train signaling applications.

The word scenario refers to the response of the system to a domain event. A domain event is something that happens within the realm of interest, to which there is a possible response that may produce a system malfunction in some cases.

Scenarios may be of different classes: normal case scenarios, alternative case scenarios, exception case scenarios, and what-if scenarios.

Based on these assumptions, TECOM partners followed detailed requirements specification techniques for writing use cases and misuse cases. This process included a template for describing the requirements as well as guidelines for writing related case scenarios.

Misuse cases are a relatively new technique for describing a negative form of a use case. Misuse cases are useful in systems where there are concerns about security and/or safety (Sindre 2001). For these concerns, a misuse case describes active opposition to proper functioning in the form of a hostile agent's goal, which is detrimental to some of the system capabilities.

A mis-actor is the opposite of a normal actor, in that the mis-actor intentionally or accidentally initiates a misuse case scenario, which the system should not support. A mis-actors can be more abstract than ordinary actor in use cases, for instance “bad luck”, “human error,” “electromagnetic interference” or “bad weather”. For example, the weather can be anthropomorphized as a mis-actor when it causes a car to skid.

Typical goals of hostile mis-actors are:

- to harm the system, environment or people
- to achieve goals that are incompatible with the system’s purpose.

Misuse cases are created for describing the fault–error scenarios by pairing system threats with system parts faults or vulnerabilities. This subprocess is represented as the loop in Figure 2. The subprocess of threats-vulnerabilities pairing ends when a complete enough criteria that is all system assets are considered, is met. Step 4 consists of the specification of requirements that reduce the risk of each threat-vulnerability pairing. In step 5 these requirements are classified according the trust dimensions defined in the conceptual model of Figure 1.

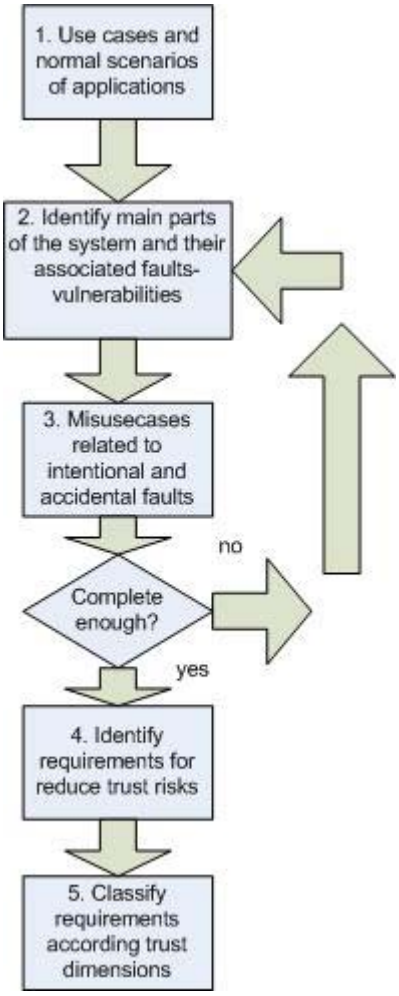

Figure 2: Process for specifying trust requirements.

Requirements Commonalities and Variabilities

Although the embedded systems developed by TECOM partners had a wide range of scopes and functionalities, from the point of view of trust they were treated as a product line. Since the goal was to discover commonalities on trust requirements, an approach based on the identification of commonalities and variabilities was used (Savolainen 2007). The trust requirements of the TECOM systems are based on the quality factors and trust dimensions defined above in the conceptual model.

A quality model decomposes each trust dimension into quality factors (i.e. aspects, attributes or characteristics) and subfactors. Typically quality factor names end with “ility”. Quality criteria are descriptions of something that provide evidence either for or against the existence of a specific quality factor or subfactor.

Quality criteria are used when quality factors and subfactors become very specific, and in particular application specific. There are often many possible quality criteria that can be chosen for any given quality factor or subfactor, and these criteria can often be parameterized using a standard template with variable parts.

To illustrate this, the requirements of the trust-security dimension for each TECOM system are grouped

using the factors defined by Firesmith (Firesmith 2004):

- **Access Control** is the degree to which the system limits access to its resources only to its authorized externals (e.g., human users, programs, processes, devices, or other systems). The following are quality subfactors of the access-control quality factor:
  - **Identification** is the degree to which the system identifies (i.e., recognizes) its externals before interacting with them.
  - **Authentication** is the degree to which the system verifies the claimed identities of its externals before interacting with them. Thus, authentication verifies that the claimed identity is legitimate and belongs to the claimant.
  - **Authorization** is the degree to which access and usage privileges of authenticated externals are properly granted and enforced.
- **Attack/Harm Detection** is the degree to which attempted or successful attacks (or their resulting harm) are detected, recorded, and notified.
- **Availability Protection** is the degree to which various types of Denial of Service attacks are prevented from decreasing the operational availability of the system. This is quite different from the traditional availability quality factor, which deals with the operational availability of the system when it is not under attack.
- **Freshness** is the degree to which information is not a copy of another one received in the past. This factor prevents replay attacks.
- **Security Integrity** is the degree to which system assets are protected from intentional and unauthorized corruption. These system assets could be divided in several groups (data, equipment, people, software and service) as seen in Figure 1.
- **Nonrepudiation** is the degree to which a party to an interaction (e.g., message, transaction, transmission of data) is prevented from successfully repudiating (i.e., denying) any aspect of the interaction.
- **Physical Protection** is the degree to which the system protects itself and its assets from physical attack.
- **Privacy** is the degree to which unauthorized parties are prevented from obtaining sensitive information. Privacy includes the following subfactors:
  - **Anonymity** is the degree to which users' identities are hidden away from unauthorized parties.
  - **Confidentiality** is the degree to which sensitive information is not disclosed to unauthorized parties.
- **Prosecution** is the degree to which the system supports the prosecution of attackers.
- **Recovery** is the degree to which the system recovers after a successful attack. Recovery is also a factor of survivability.
- **Security Auditing** is the degree to which security personnel are enabled to audit the status and use of security mechanisms by analyzing security-related events.
- **System Adaptation** is the degree to which the system learns from attacks in order to adapt its security

countermeasures to protect itself from similar attacks in the future.

Once the requirements were grouped, they were arranged in a table, marking which TECOM system had a requirement related to each of the factors or subfactors. With these inputs, the TECOM systems requirements could be divided into three classes:

- **Common** trust requirements which are common to all TECOM systems
- **Partial** trust requirements which are specified in more than one TECOM system
- **Unique** trust requirements where are specified in only one TECOM system.

For example, take a group of 3 systems, each in a different application domain (i.e. home control, video surveillance and automotive) and these systems have the following trust-security requirements:

- **Sys1:**
  - S1-R1: User authentication/identification
  - S1-R2: Boot time integrity check
  - S1-R3: Data replication in different locations
  - Others not described for the sake of brevity
- **Sys2**
  - S2-R1: Data integrity check
  - S2-R2: Backups of software and data
  - Others not described for the sake of brevity
- **Sys3**
  - S3-R3: Integrity check of stored snapshots
  - Others not described for the sake of brevity

The quality attributes of these requirements are organized in Table 1.

Table 1: Quality attributes

ID	(Sub) Factors	Sys1	Sys2	Sys3	Property
1	Access control	√			Unique
1.1	Identification	√			Unique
1.2	Authentication	√			Unique
…	Others				
3	Integrity	√	√	√	Common
4	Availability	√	√		Partial

…	Others				

After the quality attributes are organized in the table, we know what (sub) factors have in common the different systems. We could go one step further by trying to find similar requirements among these systems. In this step we could use a requirements table for the traceability of specified requirements and TECOM systems implementing them.

Table 2: System requirements commonalities

ID	Requirement	Sys1	Sys2	Sys3	Property
1	User authentication/identification	**S1-R1**			Unique
2	Integrity checks	**S1-R2**	**S2-R1**	**S3-R1**	Common
3	Backups and data replication	**S1-R3**	**S2-R2**		Partial
4	Others…				

The system requirements table allows us to classify system requirements according the trust dimensions and quality factors. With this information we can prioritize requirements and find the best approach to implement the most critical ones, taking into account the possibility of reusing the same design selection across the different TECOM application domains.

To Conclude

Frequently requirements specification is not seen as a complex issue, so the effort spent by the embedded systems industry in specifying good requirements is insufficient for a successful project.

This paper presents an approach for the specification of non-functional requirements, mainly for those related to trust in embedded systems. The trust dimensions include reliability, integrity, availability, survivability, safety and security. The approach has two principles: a well established conceptual model of the terms used to build the requirements specification, and a well defined process to elicit and specify trust requirements for embedded systems and to identify their commonalities.

This approach was used in the ITEA2 TECOM project which has the goal of developing solutions for trust in embedded systems. Since diverse systems were built in the project, the requirements specification approach had to be easy to adopt and use, and rigorous enough to meet the project partners' goals. Finding commonalities among the trust requirements specified for the different embedded systems was one of the main project issues.

In TECOM, two reference architectures were built. One is based on virtualization technologies and uses Xtratum hypervisor developed by Universidad Politécnica de Valencia, and the other is based on a Java middleware framework developed by Atego using the OSGi standard. These architectures offer mechanisms and services that allow designers to enforce the trust requirements.

Acknowledgements

The collaboration and input provided by the TECOM project partners was critical for the success of the systems requirements specification. The partners are: Atego (France), EADS DS (France), Fagor Electrodomesticos (Spain), Ikerlan (Spain) Technikon (Austria), Thomson (France), Trialog (France), Universidad Politécnica de Madrid (Spain) Universidad Politécnica de Valencia (Spain), and Visual Tools (Spain).

We wish to express our thanks to the English style reviewers Barbara Raither and Maria Tellechea for their contribution.

TECOM-ITEA is part of ITEA 2, a EUREKA Programme running under the European cluster programme. The work of the Spanish TECOM partners is partially funded by the MITYC, the Spanish Ministry of Industry, Tourism and Trade, under contract FIT-340005-2007-40. The work of the French TECOM partners is partially funded by the French Ministry of Economy, Finance & Employment.

References

Avizienis A., Laprie J.C. and B. Randell. 2001. Fundamental Concepts of Dependability. Research Report No 1145, LAAS-CNRS, April 2001.

Firesmith D.2004. Specifying Reusable Security Requirements, in Journal of Object Technology, vol. 3, no. 1, January-February 2004, pp. 61-75.

Hwang, D.D. Schaumont, P. Tiri K. and I. Verbauwhede.2006. Securing Embedded Systems. IEEE Security & Privacy, March-April 2006, pages 40-49.

Muskens, J. et al. 2005. Definitions of Trust. ITEA Trust4All.

Powell D.and R. Stroud (Eds). 2003. Conceptual Model and Architecture of MAFTIA. Project IST-1999-11583. Deliverable D 21.

Savolainen, J.. Kauppinen M and T. Männistö. 2007. Identifying key requirements for a new product line. 14th Asia-Pacific Software Engineering conference. IEEE.

Schneider F.B. (Ed).1998. Trust in Cyberspace. National Academy Press. Washington DC.

Sindre G.and A.L. Opdahl. 2001. Templates for Misuse Case Description. Proceedings 7th International Workshop on Requirements Engineering: Foundation for Software Quality. REFSQ 2001.

Biographies

Jose L. Fernández has a PhD in Computer Science and an Engineering Degree in Aeronautical Engineering, both from Madrid Technical University (UPM). He has 27 years of experience in industry as system engineer, project leader, researcher and department manager. He was involved in projects dealing with software development and maintenance of large systems, specifically real-time systems for air traffic control, avionics, power plants Supervisory Control and Data Acquisition (SCADA) and cellular phone applications. Currently, he is consultant, researcher and part time associate professor at the Industrial Engineering School of the Madrid Technical University (UPM).

Rubén Alonso is a PhD student in the University of the Basque Country and he holds a Computer

Engineering degree from University of Deusto. He has been involved in industrial and research projects since 2004, mainly related in trusted computing, security and embedded systems. Currently, he is responsible for trust and embedded systems research projects in the Technology and Innovation Department of Visual Tools, S.A.

Francisco Gómez holds a MS. Degree from the Technical University of Madrid. He works at Visual Tools as Director of Technology and Innovation and he was one of the company founders. He has academic and industrial background with thirty years experience. Prior to his current job, he was responsible for On-Board Software at the On-Board Data Division of the European Space Agency (ESTEC, The Netherlands). He has been teaching at University Carlos III of Madrid and involved in R&D projects for the last fifteen years. His main interests are embedded real-time systems, user-computer interaction and video processing systems.

Christophe Jouvray holds a PhD from University of Orsay. His doctoral thesis was on the support of model driven engineering for intelligent sensors. He has more than 10 years of experience in embedded systems and the engineering process for embedded systems. He is currently involved in the development of security, trust and privacy metamodels for embedded systems. Prior to joining Trialog, he was with CEA List (a French laboratory specializing in real-time embedded system modeling) and at the IEF (Fundamental Electronic Institute at Orsay University) where he worked on real-time embedded modeling, software component approaches, and middleware and execution platforms.

Yann Rouxel has a PhD in Physics from University Joseph Fourier in Grenoble, and an engineering Degree in Telecommunications from ENST Bretagne. He has been involved as a developer, architect and technical leader in IT software projects since 1996, mainly in the defense field. He currently works at EADS DS in Elancourt, France.

Antonio Pérez is an Electronic Engineer (1996, E.N.S.E.R.B. Burdeos) with one-year experience at IXL (France) and over 10 years experience at Ikerlan-K4 in Spain. He has been a project leader in high reliability embedded systems, PLCs development, CNC development, development of high performance human machine interfaces, distributed control systems for vertical transportation, and oven control systems. He has worked for IKERLAN-K4 as a scientist researcher on real-time and safety-related critical embedded systems. Currently he is the head of the Electronics department and the coordinator of the embedded systems research line. He has lead more than 10 R&D projects.