

Applying Security Standards to Multi Agent Systems

Tim Geissler, Olaf Kroll-Peters

DAI-Labor, TU Berlin, Germany
{Tim.Geissler, Olaf.Kroll-Peters}@DAI-Labor.de

Abstract. There is a growing need for the evaluation and certification of distributed systems regarding security standards. Almost all existing multi-agent systems that have been developed in the context of the telecommunications market neglect strong security evaluations. In particular, an evaluation of a multi-agent system based on the widely accepted and well known Common Criteria standard has not been carried out successfully yet. A certified agent environment must at least support the areas: Protection of user data, strength of cryptography, and agent communication. Furthermore, for a successful evaluation, it is required to investigate the environmental aspects, such as configuration management, lifecycle support, product delivery and guidance of administration, testing, functional and high level specifications, and a vulnerability analysis concluding the evaluation. This paper describes the experiences of certifying the Serviceware Framework JIAC IV¹, a multi-agent system in compliance with the Common Criteria security standard.

1 Introduction

Today's trust of consumers in security aspects and measures is supported by security certificates that are provided by an independent security institution, e.g. a federal office for security. These certificates establish a security standard, because they are bound to a specified and evaluated process. Such a process is called a "product evaluation" and involves three parties: the developer, the evaluator and the certifier. Firstly the product developer describes the product after a given guideline, such as the Common Criteria (CC) standard. Secondly these documents and the product itself are analyzed by an evaluation facility. Finally, if the evaluation process was successful a certificate is issued by the accredited certifier.

Typical evaluations are carried out on products that require a high security assurance such as smart card readers, firewalls, PKI infrastructures, etc. To improve secure development and interoperability between related products it is required that the certification process is expanded to further categories. This assumption can be intensified by the fact that the level of service interaction and heterogeneous infrastructures is increasing. This is a problem that distributed systems, such as multi agent systems (MAS), can partially solve. Apart from the evaluation described in this paper, no evaluation of an agent system was successfully accomplished yet. The certification of JIAC-IV was carried out to address this problem.

¹ JIAC-IV project is funded by T-Systems Nova GmbH

The remaining of the document is organized as follows: Section Two explains the history and terminology of the CC standard. Section Three comprises a summary on the security solutions and measures in MAS and clarifies the concrete demands in respect to the CC standard. Section Four introduces the Serviceware Framework JIAC IV in accordance to the previous sections. The focus of section Five is the knowledge expertise and experience of the JIAC IV certification process. Finally the summary will conclude the paper and give a prospective on how certification of distributed systems can extend trust in future Information and Communication Technology (ICT).

2 The Common Criteria Standard

The CC is an accepted world-wide standard to define, assess, and measure the security aspects of an ICT product. The CC provides thereby an understanding of what the product realizes on security functionality and secondly assures this functionality by an independent evaluation. This typically allows a high degree of comparability between different evaluations of a product category. Also any consumer can review the criteria standard and the evaluation methodology.

2.1 History and Basics of the CC Standard

Being a relatively new certification scheme, the Common Criteria is gradually replacing the older European ITSEC and the U.S. TCSEC standards because it utilizes more elaborate assurance methods. Historically in October 1998, the CC was created as a mutual agreement between government organizations of Canada, France, Germany, the United Kingdom and the United States to recognize security evaluation standards and methods internationally. The origins of the CC can be traced back to initial developments by the U.S. federal government in 1985 as the TCSEC or the Orange Book. Since then, several criteria versions for security evaluation have been formulated. Today the CC signatories also include Australia, Finland, Greece, Italy, New Zealand, Norway and Spain. For further information and explanation on the CC standard see [1].

A certification is a technical audit for a product in accordance with generally accepted security criteria. In a CC evaluation a product is evaluated in a defined process against a set of criteria. The product itself is evaluated and the developing process, as well as the company that developed the product, and the development environment.

This means that the following two aspects have to be regarded: On the one hand there are the functional requirements of a security product, which include topics such as cryptographic support, communication, protection of user data, and identification and authentication etc. These are requirements for the desired security product (see [2]). On the other hand there are the assurance requirements that confirm the effectiveness of the implementation by analyzing the product's functional specification, the design process itself, product testing, as well as the post-production aspects, such as secure deployment, installation, configuration, and a guidance to assure secure administration. Finally all provided documents and the product itself is subject of a vulnerability analysis. This procedure is specified in more detail in [3].

2.2. The Certification Process

The CC has a defined set of Evaluation Assurance Levels (EALs) that measure the criteria of evaluation of the security product and test it to verify that it meets its security claims stated by the ICT product vendor. The EALs offer a comparative platform to the consumer for selecting a product, because they can only be fulfilled when following the standards of a CC certification ([1]). In EAL 3, which is the specified level of evaluation assurance for the security product introduced in this document, the evaluation includes aspects of testing, environmental development, and a vulnerability analysis. Previous and further levels do also exist but are not considered here either.

The process of certification starts with a definition of the “Target of Evaluation” (TOE) i.e. the specific ICT product or system that is subject to evaluation. The first document written according to the CC standard is the Security Target (ST). This document contains the security objectives and requirements by concentrating on a specific TOE and defining its functional and assurance measures.

As already mentioned the ST also specifies the evaluation assurance requirements, which are, for an EAL 3 certification: the evaluation of the configuration management (ACM), the deployment process of delivery and operation of the security product (ADO), the product specifications in functional and high-level specifications (ADV), the administrator and user guidance documents (AGD), the life cycle support (ALC), testing of the product (ATE), and finally a vulnerability analysis (AVA), on the base of the provided document specifications, concludes the certification.

The following figure clarifies how this process, starting with the TOE definition and the initial security target document, covers the specified security requirements for the TOE including both the functional and the assurance families of the CC standard.

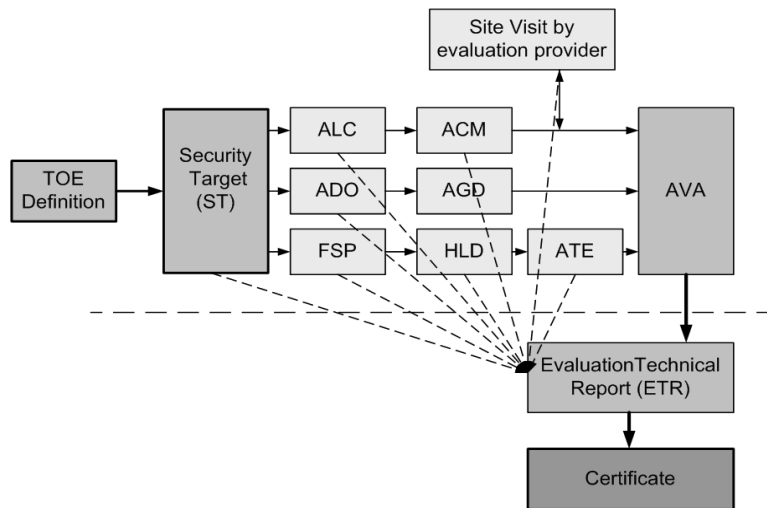


Figure 1: Timeline in the certification process

Once the ST is accepted by the evaluation facility the certification process continues with the outlined documents (see figure 1) within the development environment.

This particular evaluation covers about 800 pages of extensive documentation in a dozen documents plus secondary documentation. Each document needs to be adjusted during multiple iteration steps. Testing covers 250 tests that all need to be capillary described and accomplished on various operating systems, to assure platform independency as provided by Java. There was even more distinction on the day of the site-visit. Evaluator and certifier required proof for the existence of sentries or guards during night time, shatter-proof glass, and resistance of all door locks and so on.

3. Security in Multi Agent Systems

Intelligent software agents are a basic technology for the construction of distributed applications. Agents fulfilling their tasks in an autonomous manner are well suited for being used in a distributed environment. In addition to the concepts of object-oriented programming, software agents also incorporate certain aspects of artificial intelligence (e.g. learning and adaptation) and therefore enhance the possibilities of the software communication process. Software agents provide a wide range of approaches to satisfy the growing requirements of today's enterprise. For further definitions see Maes [4], Ferber [5], Wooldridge and Jennings [6], and Albayrak [7].

When introducing data security to agent systems, the security goals of ICT systems need to be used as a basis and are specified in compliance with RFC 1825 [8]:

- Data integrity means to guarantee that data cannot be changed without notice.
- Confidentiality guarantees that sensitive data arrives undiscovered at a place of destination and can only be identified and decoded by correct entities.
- Non Repudiation proves that the maintained sender is also the actual sender.
- Authorization, guarantees that each entity is responsible for its actions.

Agents are typically used to interact in distributed systems that can lead to a possible loss of control on the agent owner's side. Therefore the security context of agent systems is not only a new subject for the certification discussion but much more a strongly relevant one for the ICT, especially the telecommunications market. This requires that measures against various security challenges and threats must be taken. For known approaches that consider security in MAS see [9] and [10]. The next paragraph lists threats that exist in relation to the life cycle of an agent or agent platform.

3.1. Threats in Multi Agent Systems

The following concretizes the views of security challenges in agent systems:

1. Man in the middle attacks are characterized by an attacker eavesdropping on data sent between two agent platforms, migrating agent data, and during the exchange of user data, comprising identification and authentication data, or application data.
2. Eavesdropped data can lead to modification of data, where an attacker manages to modify data regarding to his own goals and needs to gain unauthorized access.
3. In a replay attack gained knowledge of data content is used to modify data content that has been transported before to acquire unauthorized information. Such a “spoofing and masquerading attack” threatens the integrity of the platform.

4. Derivation of private key data, from publicly known data, such as the public key. This threatens confidentiality of the TOE key-pair and of all security functionality.
5. The fifth threat exists in the refusal of the receipt or dispatching of a message. Beyond that, an agent could refuse its agreement to a contract or deny its actions in the past, which would break the security goal of non-repudiation.
6. Finally the possibility of flooding offered services of a target platform with service requests, also known as a Denial of Service attack would result in an operational overload so that the target system crashes.

The countermeasures to these threats along with the basics and the core aspects of the Serviceware Framework JIAC IV are presented in the next section.

4. The Serviceware Framework JIAC IV

In the following the Java Intelligent Agent Componentware (JIAC IV) is introduced. JIAC IV is a Serviceware Framework with a scalable architecture based on the programming language Java and is intended mainly for developing applications within the telecommunications and telematics market.

The main goal of JIAC IV is to support the design, implementation, and deployment of software agent systems, which also allows the possibility of reusing applications. It includes development methodologies, software development tools, and a runtime environment that can be observed, configured, and controlled by the developer. The core aspects of JIAC IV are: distribution of agent systems (single components), scalability and flexibility of the underlying agent architecture, adaptability and knowledge based interaction via agent services.

The certified JIAC IV product provides the development of secure services oriented towards MAS. Applications can be developed with the help of a library that consists of prepared services, components, and agents, which can be integrated into an application system in order to perform standard tasks. Individual agents are based on a component architecture, which already provides the basic functionality for communication and process management. Only the functionality that is specific to the application must be provided and interactively integrated by the system's engineer.

The runtime environment of an agent is the agent platform, which is represented by an agent management system. Agents may migrate from one platform to another. The communication basis of the agents is formed by Java components, whereby the first communication act of an agent on the platform is accomplished during the logon procedure with the manager. In the further life cycle of the agent, communication acts only start during beginning of any service usage.

The security of the agents and of the agent platform is ensured by the use of safety mechanisms, which were built into the communication and the migration process. It is possible to limit the service ability of offered agent services, by means of a service authorization. For further information see [11].

The following description shall help to understand the JIAC IV Serviceware Framework in its connection to applied security functions (see Section 5.3). For further information on JIAC IV we refer to the published specifications in [7], and [13].

4.1. The Core Concept of a JIAC Agent

Each Agent in JIAC IV consists of knowledge elements and Java components. Knowledge elements are written down and specified with the JIAC Agent Definition Language (JADL) and must be compiled by JIAC IV's Ontology Compiler. The ontology is used to define needed data structures. Services and the appropriate protocols are defined in plan elements. Furthermore Java components serve functionalities that are not knowledge-based as an interface, e.g. accesses to data bases. These elements and the Java components can be added or removed at run-time to or from the agent.

The core of a JIAC agent is designed in accordance with the BDI Model for autonomous agents. For further information on BDI see e.g. [12]. The basic concept of this model is the distinction between knowledge (Beliefs), goals (Desires) and planned actions (Intentions). A brief description of the realization of these concepts in JIAC follows in the next paragraphs.

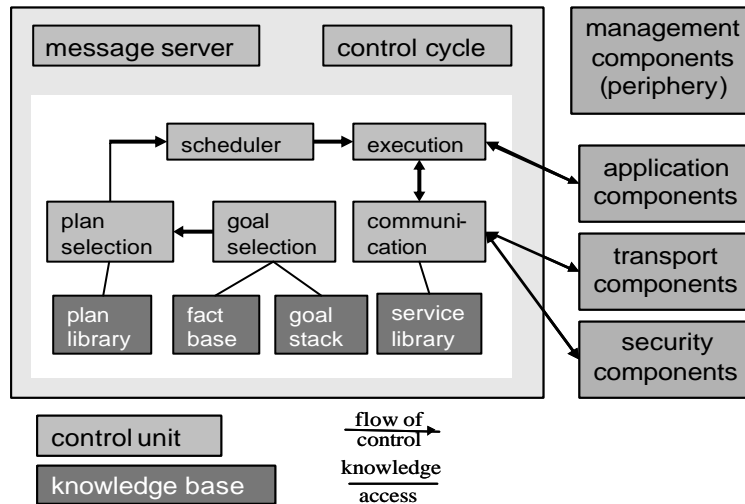


Figure 2: The core of a JIAC agent

The message server and the control cycle are components that belong to the internal infrastructure and are included in every JIAC agent to support communication and the corresponding coordinated execution of the received messages. The message server enables message passing between the peripheral components of any agent.

The internal control units organize an agent's infrastructural basics, such as protocols for communication with other agents by ingoing and outgoing speech-acts (according to [14]) sent and received via the transport component.

To complete an agent's action and interaction the components that belong to the knowledge base include libraries about plans to fulfill provided services that are stored in the fact base, as well as the agent's goal stack.

The behavior of JIAC agents is goal triggered. Whereas the fact base holds the agent's ontologies and objects, the GoalStack stores new goals until they become active, and the GoalSelection determines goals that shall be processed next. When a

goal is selected, it is propagated to the PlanSelection component that searches the PlanLibrary. This library holds a list of all available actions (plan descriptions), to find a service that can fulfill the goal. If no applicable action can be found the agent contacts the DirectoryFacilitator (see [15]) and asks for remote services, which may fulfill the goal. After this process an action becomes an intention and is scheduled for execution. This is supervised by the scheduler- and execution-components.

4.2. Countermeasures to Identified Security Threats

The following table demonstrates how the CC standard was applied within the evaluation of the Serviceware Framework JIAC IV in contrast to stated security aims:

CC Class Name	Taken measures as specified by the CC standard	Threats in MAS					
		1	2	3	4	5	6
Communication	Non-repudiation of origin					x	
Cryptographic support	Cryptographic key management	x	x		x		
	Cryptographic operation	x	x		x		
User data protection	Access control policy	x	x	x	x		
	Import/Export outside of TOE control		x		x	x	
	Stored data and data exchange integrity		x		x		
Identification and authentication	User attribute definition	x				x	
	Specification of secrets	x	x		x		
	User identification and authentication	x	x		x		
	User-subject binding			x		x	
Security management	Management of security attributes	x				x	
	Management of TOE security data		x		x		
	Specification of management functions	x	x	x	x		
	Security management roles	x				x	
Protection of the TOE security functions	Fail secure						x
	TOE - data consistency	x	x		x		x
	TOE - trusted channel	x	x	x			
	Trusted path	x	x		x		

Table 1. The Functional Security Requirements and their correlation to the identified threats in multi-agent systems (see section 3.1)

The sixth threat “flooding of offered services” could only be prevented partially, because “denial of service” attacks are a very common and hard to solve problem when services are provided via a network. Also this certification process is only based on the evaluation of implemented software without requiring additional hardware.

In the following description the certification process of JIAC IV is used to illustrate how the implemented security measures can be classified and revised. The implemented security functionality is encapsulated within special security functions and is explained in the following chapter (see Section 5.3).

5. The Certification Process of JIAC IV

As already mentioned the JIAC-IV certification extends the typical limits of the CC expertise, because MAS differ from the mentioned ICT products. In the next section the security threats and countermeasures that were identified in Sections 3.1 and 4.2 will be presented in comparison with the statements given in the Security Target.

5.1. The TOE and the Security Target

As noted it is essential for the ST to define the TOE (see Section 2.3) as: "...the Java Intelligent Agent Componentware developed by the DAI-Labor...JIAC IV is suitable for business, telematics, and telecommunication services." [17] JIAC IV can be described as an: Agent platform that consists of (i) a local platform (running on a Java Virtual Machine) that constitutes a platform independent runtime environment; (ii) an Agent system responsible to serve management and cryptographic support to (iii) Public services that are carried out by agents.

In the following agents are understood as software systems, that are managed by a specific multi-agent platform. Agents are capable to act autonomously and in a flexible way to realize specified services. The agent platform gains and obtains its resources from the runtime environment that is realized by the JVM. Therefore the runtime environment can be understood as the underlying layer of the agent platform. Furthermore every platform is connected to other agent platforms by a network of trustworthy platforms.

To offer scalability and flexibility the basic concept of JIAC IV is based on knowledge and interaction via service agents. These agents can be separated into infrastructure agents that must remain stationary and manage the support maintenance and security functionalities. Secondly service agents can also be seen as application services, including the mobile agent scenario, which are accessible via the agent platform on behalf of a user and can be handled by a graphical user interface.

Agent platforms can transfer speech acts, as in the meaning of remote platform data, to update information on service application data or simply to use a service. Furthermore this includes mobile agents that are able to migrate from one platform to another on behalf or as part of a service.

5.2. The Target of Evaluation

As already stated the TOE is the central focus of an ICT product certification. Therefore in the following the TOE is described along with the specification for each of the TOE's Security Function (SF). The specifications in the ST require further refinement and lead to the evaluation assurance classes, which can be applied to the TOE in accordance to the specifications provided in [1], [2], and [3].

The local platform (TOE) provides interfaces to a Certification Authority that supports the agent platform with valid certificates (see [16]) over trustworthy remote platforms. These platforms provide additional services to enhance the service diversity of the local platform. An LDAP service provides a yellow page directory to find

remote platforms and registered agent services. The trustworthy user interface is used to establish a trusted path and secure channel to human users. Finally a physical administrator access provides the possibility to install and administer the local platform.

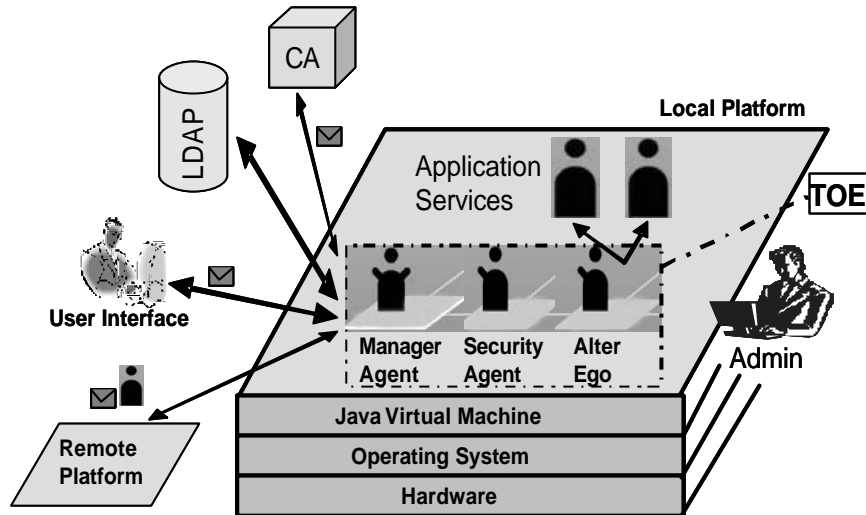


Figure 3: The TOE and its environment

In the following sections we describe for each TOE Security Function (TSF) the behavior of the security function and the purpose and method of use of their interfaces. For further information see the ST ([17]).

5.3. Specification of the Security Functions of the TOE

Before any data exchange takes place (besides X.509 certificates), a secure communication channel (as specified in [18]) between the user interface, remote platform, or CA and the local platform (TOE) is always established. This trusted channel protects data from modification, deletion, insertion, and replay and furthermore protects the transferred content by encryption. This enables a trustworthy connection and has to be understood by the meaning that time-valid certificates, issued and signed by a valid CA, have been verified successfully by both sides of the connection. Only certificates that have passed the time-validity check and are not on the list of revoked certificates (CRL) are referred to as valid. Also any SSL communication is established in accordance with a specified cryptographic algorithm that uses SSL with 3DES and a cryptographic key size of 168 bit. Further cryptographic algorithms are RSA (the used asymmetrical key-pair) with a cryptographic key size of 1024 or 2048 bit and SHA for secure hashing of data. In case the management of the TOE cannot react properly anymore to upcoming exceptions the TOE switches into a secure state by disabling all external communication. Otherwise external communication can only be enabled or disabled by the TOE administrator.

The following five security functions are used to protect all identified interfaces of the TOE: Security function 1 deals with the user interface and the user communication with the TOE. The identification and authentication (I&A) of human users, which want to access the TOE, can only take place before any service usage is instantiated and after the establishment of a secure communication channel (SSL) with the TOE. Pass-phrases need to be alphanumerical with a minimal length of eight characters. The TOE relates the user to the received I&A data and associates the transmitted data to the human user by providing a reference number to the used service. Service results during the communication process are always returned to the identified user interface. Service agents can also migrate to other trustworthy remote platforms and carry out actions on the user's behalf.

Security function 2 deals with the exchange of speech-acts to and from trustworthy remote platforms. These speech-acts are used to exchange data on services and agents. Before any data exchange takes place, a SSL channel between the remote platform and the local platform needs to be established to protect the transported data. Import and export of speech-acts is only accepted when sent over an SSL connection.

Security function 3 assures the exchange of LDAP based data via an establishment of a TCP/IP channel. For this function it is required to know the LDAP's (host) address and to be able to exchange data as specified within the LDAP protocol (further specification can be found in [19]). When data import and export to and from the LDAP takes place it can only be interpreted as agent registration data. This data can either be about registered agents (import) and about new agents (export) that are registered at the LDAP. Very important is that this security function improves the secure behavior of the TOE by only allowing incoming TCP/IP plain-text as LDAP based agent registration data. The TOE has to ensure that these data are exported without any security attributes including sensitive TOE internals.

Security function 4 deals with the transmission of mobile agents that can be considered as a special kind of a speech-act to trustworthy remote platforms. Import and export of mobile agents is only accepted when the data is sent over a trusted SSL connection. Each service agent that is involved with the fulfillment of a user task internally stores a service-ID that is maintained and transmitted within that agent so that the TOE can associate the service-ID with the identified (human) user at any time.

Finally security function 5 deals with the exchange of certificates to identify trustworthy remote user interfaces, and remote platforms as well as a CRL to identify certificates that must be revoked. A trustworthy connection is established if the signature of the provided certificate or CRL is valid when counterchecking its information with the local public key of the CA. This is because the CA is the main authority and there is no method to proof its integrity other than a human administrator who can verify the fingerprint of the certificate with the CA. If this certificate leaks integrity, all external communication will be stopped.

The TOE must always verify the integrity of the certificates and the CRL by comparing the deciphered hash - that was encrypted by the CA using its private key - provided by the CA with the hash that was calculated by the TOE. If an invalid signature is found, the received data will be dismissed.

Furthermore the TOE is responsible for the management of internally stored keys. This especially addresses monitoring the integrity of the TOE (RSA) key-pair and the imported CA's public key. Generation and destruction of the key-pair (zeroisation

key-pair file) belonging to the local platform is realized by the TOE's functionality but must be initialized (console command) by the administrator of that platform. If a loss of integrity is detected the platform will switch to a secure state where all external communication is stopped until the corrupted files have been replaced.

The administrator is the only person that has the ability to modify the behavior of functions/agents by disabling or enabling functional parameters of all agents.

6. Summary

The certification after the CC standard is used to establish a standard for the comparison and estimation of security measures of a product. On the contrary to the standards suggested by FIPA, the CC standard is not applied to agent systems but is accepted world-wide. Whereas the FIPA standard is specifically bound to the development of agent systems, the CC standard is more like a generic catalogue for countermeasures in ICT security and safety. But even though the rule set of the CC covers some 100 pages the evaluation is always subject to strong fluctuations of specifications, since the evaluation facility must concretely hold the frame that the certification possibilities itself allow. In the case of a certification of a multi agent system this is complicated by the fact that no product has been certified, which is working in heterogeneous environments, like MAS yet.

This problem has been addressed by involving numerous iterations on each stage of the certification and development process. We have illustrated and described the experiences that we made during the certification process with the evaluation facility and with the accredited certifier.

Our aim was furthermore to improve the creation of other Serviceware Frameworks by our expertise. This should serve for the fact that any certification of similar products can be accomplished and carried out less complicated in future operations by keeping the area for ambiguities as small as possible. Due to our experience the strength of a certification process is emerging when not concentrating on the whole product from the beginning as much more starting with the smallest frame of product definition and to continue and iterate throughout the whole process from that point. This opens up the possibility to increase and accumulate the immense knowledge that is required for a certification process. Also it simplifies to extend a certificate by a re-certification that can be carried out afterwards.

On the other hand the development of the product and the remaining certification documents and tests can be provided more clearly within the various iteration loops that always focus to remain on a minimum of clearly given specifications.

This clarifies most of the problems during the identification of the product design and specification to the involved instances. For example in order to clearly define the working area our first document was sent to the evaluator more than 100 times. This process of writing just about 80 pages took about half a year. After that a common comprehension on the usage of the CC methodology had been found between us and the evaluator. But still the certifier disapproved our document and the process of reiteration had to be started again. This exemplifies that adaptation of the certification process to security in agent systems needs to be strongly adjusted.

Since the accomplishment of a certification examines almost all aspects of a software development process it thereby allows the possibility to examine all actions that are involved in the considered development process on the one hand. On the other hand tasks that were established in an environment development are examined and counterchecked during the evaluation and therefore lead to firmly defined actions that also improve the level of product development.

A certification therefore stands not only for its use in improving the security functionality of a product itself; it also establishes and checks the efforts for concrete product quality management. Thereby it improves the developer's commitment to security and safety throughout the whole product lifecycle. As the certificate itself is handed out by an independent third party (the accredited certifier) the consumer awareness is based on comparability and offers the possibility to transparently and coherently investigate the security level before making any decision of investment.

7. References

- [1] Common Criteria, Part 1: Introduction and General Model, 1999
- [2] Common Criteria, Part 2: Security Functional Requirements, 1999
- [3] Common Criteria, Part 3: Security Assurance Requirements, 1999
- [4] Life like Autonomous Agents, Pattie Maes, ACM Press Vol.38 No.11, pp. 108, 110, 1995
- [5] Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence, Jacques Ferber, Addison Wesley Longman, 1999
- [6] Agent Theories, Architectures, and Languages: a Survey, in Wooldridge, Michael and Nicholas R. Jennings (Eds.), Intelligent Agents, pp. 1-22, Springer, 1995
- [7] Introduction to Agent Oriented Technology for Telecommunications, Sahin Albayrak (Ed.): Intelligent Agents for Telecommunication Applications, IOS Press, p.1-18, 1998
- [8] Security Architecture for the Internet Protocol - RFC 1825, Robert Atkins, 1995
- [9] Mobile Agents and Security, Giovanni Vigna (Ed.), LNCS Vol. 1419, Springer 1998
- [10] Realization of an Agent-based Certificate Authority and Key Distribution Center, Karsten Bsufka, Stefan Holst, Torge Schmidt, IATA'99, LNAI Vol. 1699, Springer 1999
- [11] A Toolkit for the Realization of Agent-based Telematic Services and Telecommunication Applications, Stefan Fricke, Karsten Bsufka, Jan Keiser, Torge Schmidt, Ralf Sessler, and Sahin Albayrak, 2000
- [12] Intentions, Plans, and Practical Reason, Michael E. Bratman, Cambridge, MA: Harvard U. Press, 1987
- [13] Agent-based Marketplaces for Electronic Commerce, Ralf Sessler and Sahin Albayrak, International Conference on Artificial Intelligence, IC-AI 2001
- [14] FIPA Communicative Act Library Specification, Document nr. SC00037J, Foundation for Intelligent Physical Agents, Geneva, Switzerland 2002
- [15] FIPA 98 Specification, Part 1 Agent Management, Document nr. OC00002A, Foundation for Intelligent Physical Agents, Geneva, Switzerland 1998
- [16] ITU-T Recommendation X.509, The Directory Authentication Framework, 1997
- [17] Security Target (ST), Version 2.0, DAI-Labor, 2004
- [18] TLS Protocol v1.0 - RFC 2246, Network Working Group, T. Dierks, C. Allen, 1999
- [19] RFC 1777 - Lightweight Directory Access Protocol, Network Working Group, W. Yeong, T. Howes, S. Kille, 1995