2004 Spring Simulation Interoperability Workshop
Washington, D.C., April 2004

Composable Mission Spaces and M&S Repositories - Applicability of Open Standards

Dr. Andreas Tolk
Virginia Modeling Analysis & Simulation Center (VMASC)
Old Dominion University
Norfolk, VA 23529
atolk@odu.edu

Keywords:
Web Enabled M&S, Web Services, Composability, WSDL, UDDI, SOAP, XML, UML, MDA

ABSTRACT: *The challenge of generating a federation driven by operational requirements "on the fly" has been formulated in various programs and proposals, but so far, no satisfying solution has been accomplished. The use of open source doesn't satisfy the industry, which has a vital interest to keep their core solutions in-house (protection of intellectual property). The use of interface-driven solutions based on black boxes leads to discrepancies and inconsistent federations.*

The use of open standards to connect components and escorting reference models - allowing a white box view on the components functionality and their interior and exterior interrelations without revealing the implementation details - will fulfill the industry requirement for privacy as well as the user requirement for the necessary transparency required for V&V and credibility of the federated solution.

Using an improved version of the Levels of Conceptual Interoperability Model (LCIM), various open standards will be evaluated concerning their contribution to respective levels. Based on this finding, a possible implementation will be proposed showing how the various open standards can complement each other resulting in an M&S repository comprising heterogeneous solutions that can be federated into a composable mission space driven by user requirements.

Furthermore, it will be argued that aligning these standards within DoD is possible, and this will help to terminate the shortcomings of interface driven solutions as well as integrate components delivering M&S functionality into real world systems, in particular for Command and Control.

1 Introduction

The following vision drives many standardization efforts conducted and supported by the Simulation Interoperability Standards Organization.

Simulation functionality is encapsulated in components. These components are well defined and, together with their documentation, accessible in form of metadata, are stored in a repository. A user, who wants to solve a problem like conducting an analysis or experiment, comes with a set of requirements. Using these requirements, an appropriate set of components is selected from the repository and is checked out. As the components are composable, the new simulation system comprises all functionality needed to fulfill the requirements of the user. The execution is orchestrated ensuring frictionless and consistent execution and evaluation of the simulation. The services are accessible from command and control systems as well. As all project managers are using this repository, it is gradually improved by enhancing the functionality and adding new functionality in form of new components to it with every project.

Unfortunately, this vision is far from reality. Most projects are targeting a project-specific solution. The use of project overarching standards is only seldom found in the requirements of projects. Optimization of the use of resources is conducted – if at all – on project level, which eliminates the project driven development of overarching solutions. No project is willing to invest the initial costs in order to save money in follow-on projects – as they are not within the responsibility of the project manager, who must take care of his budget, not the budget of the "next generation." Therefore, overarching solutions often remain a dream within acquisition, in particular in the area of simulation component development.

Anyhow, the necessity for standards enabling reuse, composability, and orchestration is becoming obvious to an increasing number of experts. Respective requirements are formulated increasingly in new project proposals. Reusability, composability, and migration concepts for legacy solutions are required for projects such as the Joint National Training Capability (JNTC), the Distributed Continuous Experimentation Environment (DCEE), Joint Virtual Battlespace (JVB), Joint Synthetic Battlespace (JSB), and more. Interestingly enough, the common infrastructure groups are often only interested in common infrastructures for a very special subset of users and not really interested in dealing with the applicability of already established solutions from outside their domain. This "not invented here" syndrome can be found in all expert groups and conferences, on the M&S as well as on the command and control side. To solve the problem of interoperability, such preoccupations must be eliminated; the often-described "cultural gaps" have to be bridged. Solutions must comprise hooks in form of metadata to show the *operational capabilities* they support (in other words, having a place in the mission-means-framework) as well as ensure the *integration into tactical systems*. Furthermore, *behavioral representation* and metadata to support *V&V* must be part of the solution.

Up to now, no standard – or group of standards – has emerged that is accepted by all participants and that fulfills all requirements. However, evaluations conducted by the Extensible M&S Framework (XMSF) team for the Defense M&S Office (DMSO) and the U.S. Air Force Joint Synthetic Battlespace (JSB-AF) show that a family of web based open standard has the potential to support many of these requirements [1].

Within the following sections, the standards and how they have to be applied and orchestrated will be presented. The targeted objective is to set up a technical and management framework for an M&S repository being the basis for a common composable mission space. Furthermore, the potential role of SISO in these processes will be identified.

2 Levels of Conceptual Interoperability

During the Fall Simulation Interoperability Workshop 2003, Tolk and Muguira introduced a model dealing with the various levels of interoperability [2]. Using this model, Tolk and Muguira showed that **meaningful interoperability on the implementation level requires composability on the conceptual level.** This circumstance was already pointed out before in slightly different wording, when Zeigler stressed that **meaningful interoperability cannot be achieved by standards targeting the implementation level** [3].

While the view on the various levels of conceptual interoperability given in [2] was very data centric, the author developed the model further in order to cope more efficient with dynamic issues. The improvements have been influenced by ongoing studies at the University of the Federal Armed Forces in Munich, Germany, dealing with the applicability of linguistic research results to cope with issues like ontology driven interoperable solutions. A good overview on these efforts is given in [4]. Hofmann introduced a pragmatic level above the semantic level, meaning that the receiver of the information not only understand its meaning (semantic level), but also knows what to do with it. Together with the findings summarized in [2], this led to the definition of the enhanced version of the Levels of Conceptual Interoperability Model (LCIM).

- On level 0, **no connection** is established at all.
- On level 1, the **technical** level, physical connectivity is established allowing bits and bytes to be exchange.
- On level two, the **syntactical** level, **data** can be exchanged in standardized formats, i.e., the same protocols and formats are supported.
- On level 3, the **semantic** level, not only data but also its contexts, i.e. **information**, can be exchanged. The unambiguous meaning of data is defined by common reference models.
- On level 4, the **pragmatic/dynamical** level, information and its use and applicability, i.e. **knowledge**, can be exchanged. The applicability of information is here defined in an unambiguous form.
- On level 5, the **conceptual** level, a common view of the world is established, i.e. an **epistemology**.[1] This level not only comprises the implemented knowledge, but also the interrelations between these elements.

When setting up a common repository, the escorting management processes must ensure that interoperability is reached on all levels. Which open web-based standards can be applied and how they have to be aligned is topic of the following two sections.

[1] Epistemology in this context deals with the theory of knowledge, especially with reference to its limits and validity. It is a typology of the ontology, i.e., a way to formalize the knowledge about a given domain.

3 Overview of Recommended Web-based Open Standards

Within this section, a family of open web-based open standards following the ideas of the XMSF initiative will be presented to support the implementation of a web-based M&S repository for composable mission spaces on the technical side. However, is already pointed out in earlier papers, a technical solution which is not accompanied by the alignment of the management processes is doomed to failure, as it at best can lead to a point solution in time. As soon as the first updates will be conducted, the former interoperable solution will have to be adapted, and without aligned management processes this cannot succeed.

There are without doubt solutions to cope with these requirements other than the web-based open standards recommended in this paper. However, this work reflects the ideas of the XMSF group, and XMSF is defined as a composable set of standards, profiles and recommended practices for web-based modeling & simulation (M&S).

Furthermore, the author is convinced that web-enabled M&S is one of the most promising futures of distributed simulation systems and offers furthermore tremendous opportunities for integration of M&S functionality into real world systems, in particular command and control systems of the next generation. Some examples are given in [5].

In the following sub-sections, we will not deal explicitly with the technical level. We will much more assume that the components will be implemented on platforms that can be connected via basic net protocols, such as TCP/IP or UDP/IP, which is not seen as a real restriction.

3.1 XML

The first recommended standard is the Extensible Markup Language (XML). In the context of this paper it is sufficient to declare that XML provides the description of the data to be exchanged as well as storage and transmission formats. XML is related to the more general Standard Generalized Markup Language (SGML). XML expanded the browser-oriented use of the Internet, in which services provide information to a user via HTML, by enabling service-to-service communication. Wherever data must be exchanged between two services or applications, XML can be the suitable format for making the data self-describing.

In some applications, XML can be inefficient due to its consequent use of strings based on Unicode for capturing the information, but ongoing standardization efforts on a binary version of XML may solve this problem soon.

Furthermore, for some of these applications, the following argument may be valuable as well: Despite potential efficiency problems when XML is used as the mean of information exchange, XML can in any case be used to describe the interface of a component in a standardized and easy to cope with manner. This means that the description of an interface using XML alone is of value when it comes to identifying of some piece of information is available in some of the obtainable components. An example can be seen in the SEDRIS components. Although in many cases it makes only limited sense to convert the binary interfaces into Unicode XML descriptions, for data visibility purposes it makes perfect sense to describe these interfaces so that they can be discovered by other services using XML. Once the need for information exchange is discovered, the connection specific binary interface for efficient information exchange for the "operational" use can be established.

In section 4.1, we will introduce the namespace management based on XML description of the data. In summary, XML seems a reasonable choice for the standardized description of the structure of data.

3.2 SOAP, WSDL, and UDDI

The next step towards a common infrastructure utilizing web-based open standards is the application of web technology to exchange the data specified before. The recommended standard to do so is the Simple Object Access Protocol (SOAP), as already presented in direct context of XML and simulation systems in earlier SIW papers, in particular [6,7].

SOAP defines a message framework for exchanging data comprised in XML documents. SOAP provides a minimum level of transport using the Hypertext Transfer Protocol (HTTP), Simple Mail Transfer Protocol (SMTP), or the Multiple Internet Messaging Extensions (MIME) multipart. The use of alternative communication protocols is possible, but HTTP and SMTP are actually applied in most circumstances. The underlying principle of SOAP is to define simple one-way mappings for basic functions like *GET* and *POST* to send information and to ask for information. To this end, SOAP defines a mandatory envelope and body that specifies the start, content, and end of the messages, as well as obligatory headers, attachments, encoding, etc.

This simple but powerful principle allows wrapping of existing applications in the following manner: the data to be interchanged is first described using XML. Then, the access to the interfaces is defined using SOAP to deliver the information to a parser that decodes XML and feeds the interface in the required form. This method is applicable to every interface-driven system, such as databases as well as HLA-compliant federates. To exchange the FOM/SOM format using XML is an easy exercise for every XML student, and concerning the author it is only a question of time until FOM/SOM will be replaced by XML specifications.[2] Regarding the additional functionality needed, the reader has to wait until section 3.6, in which we will cope with the necessary web-based open standards to handle these challenges.

So far, XML can be used to describe the data to be interchanged and SOAP can be used to access these interfaces. However, to realize the idea of the repository with reusable components, however, self-description of the services and the possibility to locate the services to make use of them is needed as well.

These ideas are captured by the Web Service Description Language (WSDL), for which the World Wide Web Consortium (W3C) coordinates the related standardization efforts. A very generic definition for Web Services is, that they are a set of operations, modular and independent applications that can be published, discovered, and invoked over the web. Once a web service is written, it can be published and registered by the provider. Interested users can then locate it to use the functionality provided. The necessary information about the functionality of the web service, its location, content, and structure of input and output data, and constraints have to be described. This is done using the Web Service Description Language (WSDL). To describe a web service, it is necessary to describe data types as well as operations and services accessible.

We already have seen that XML can handle the definition of the data types. A very good introduction to web services and how the related standards have to be orchestrated is given by Newcomer in [8], which is the source of the following figure and explanations. Generally speaking, WSDL uses XML with a well defined tag set. To start with, WSDL describes operations and services using XML schemata combining the names and the related input and output parameter definitions. These operations can be accessed via different port types, which are logical groupings of operations, such as an object's interface definition using IDL in CORBA or type libraries in .NET or the SOM for an HLA federate. Furthermore, all this is transport specific, which means that the port type information must be embedded into a net accessibility framework comprising the necessary network address and the binding. Finally, the port definition is the shell comprising all information in the cascade just defined.

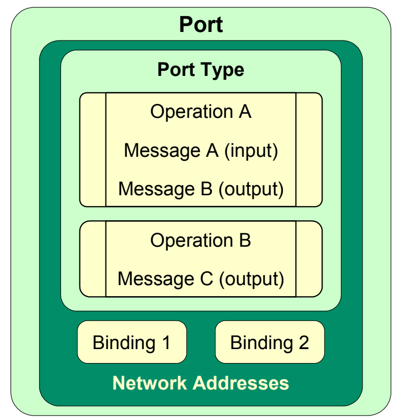

Figure 1: Cascading Elements of the WSDL

The last member of the standard family to cope with in this section is the universal description, discovery, and integration (UDDI) registry. UDDI is not really a standard, but a registry established by an industry consortium to create and implement a directory of web services. As actually implemented, UDDI is more a marketplace for web services. However, the ideas and methods should be used when establishing a common repository as envisioned in the introduction of this paper. The role of UDDI in this family of standards is the following:

As soon as a component has been migrated into a web service, i.e., the role of the operations and escorting data definitions, network addresses, and bindings are defined, this web services has to be published, so that potential users are able to locate the service and use it. To this end, the description of the service is posted to one of the UDDI registries. If a potential user is looking for support of his event, he will go to the UDDI registry as well searching for web services matching his specifications. Several problems related to UDDI are discussed in [8], such as covering the UDDI operator costs, the roles for founders, the UDDI acceptance, the quality of UDDI data, etc.

[2] When applying the IEEE 1516 Standard, this exercise isn't even required any longer. In 1516, the HLA OMT data interchange format (DIF), which is the file exchange format used to store and transfer HLA FOMs and SOMs between FOM/SOM builders, is formally defined in XML (P1516.2, Annex D).

UDDI handles the technical side of the problem quite well, however, in order to ensure usability for a repository supporting the Warfighter with reusable, composable, and orchestrated solutions for training, education, experimentation, and support of operations, reliable components are necessary. To this end, more than technical applicability is needed. The credibility of the composed solution most be ensured by the system, which means the meta-data concerning the reliability of models and data, pedigree, validation and verification data, etc. must be coped with as well as metadata ensuring composability and orchestration. The web service related standard family is a solid core of standards coping with the technical issues, but more than XML, SOAP, WSDL, and UDDI will needed to support the vision of reusable components in a common repository. We will discuss the applicability of related recommended parts of the solution and how to orchestrate them in the following sections and sub-sections.

3.3 DoD Metadata Registry and Clearinghouse

Regarding LCIM, the standards proposed so far can cover the levels 0, 1, and 2. However, in order to make sense of the information to be exchanged, we need semantic consistency based on the syntactic consistency based on XML. To this end, in addition to the DoD XML Registry, where XML tag sets are registered, the U.S. Department of Defense established the "DoD Metadata Registry and Clearinghouse" [9], which objective is given on their website as follows:

[The] Defense Information Systems Agency (DISA) is responsible for data services and other data-related infrastructures that promote interoperability and software reuse in the secure, reliable, and networked environment planned for the DoD's Global Information Grid (GIG). The Metadata Registry and Clearinghouse's primary objective is to provide software developers access to data technologies to support DoD mission applications. Through the Metadata Registry and Clearinghouse, software developers can access registered XML data and metadata components, COE database segments, and reference data tables and related meta-data information such as Country Code and US State Code. These data technologies increase the DoD's core capabilities by integrating common data, packaging database servers, implementing transformation media and using Enterprise data services built from "plug-and-play" components and data access components.

The definition of the DoD Discovery Metadata Specification (DDMS) is part of this plan and a very important step towards data-driven net centric interoperability [10]. The metadata is grouped into four categories, namely security, resource, summary content, and format.

- Security Set elements enable the description of security classification and related fields and provide for the specification of security-related attributes and may be used to support access control.
- The Resource category elements provide a way to describe aspects of a data asset that support maintenance, administration, and pedigree of the data asset.
- The Summary Content categories provide the description of concepts and additional contextual aspects of the data asset being tagged and include such elements as subject, description, and coverage.
- The Format elements provide the description of physical attributes of the asset and include elements such as file size, bit-rate or frame-rate, and mime type.

All these categories are of interest in the practical implementations of a repository, however, we will focus on the Summary Content in particular, as this category is necessary to bridge the gap between syntactic and semantic interoperability.

The actual version of the DDMS [10] provides basic Summary Content elements to capture content metadata. Activities are underway to test additional Summary Content elements that provide a more robust, structured method of describing the contents of a resource. Candidates for addition to the Summary Content Category set follows Person, Place, Organization, Material, and Event elements. It should be pointed out that these structures is very similar to the core element structure of the Command and Control Information Exchange Data Model (C2IEDM), which use as a reference model has been articulated several times in recent papers. For an overview, in particular with interest to joint and combined applications, please refer to [11]. We will also focus on the use of common reference models supporting semantic interoperability in section 4.

In summary, the necessity for additional metadata for the application domain and the use of DDMS in the domain setting up a repository supporting the Warfighter with reusable, composable, and orchestrated solutions for training, education, experimentation, and support of operations, should be the main message of this subsection.

3.4 UML

As already shown in [2], the level of semantic interoperability is necessary but not sufficient for meaningful interoperability of dynamic systems. This level is sufficient for databases and picture-driven solutions, such as the C4ISR systems of the past, but in order to cope with agile components in a dynamic domain such as battle spheres, we need standards to align and harmonize processes using this data as well.

The unified modeling language (UML) is a family of graphical notations backed by a single meta-model. Its use for describing and designing software systems, in particular those using the object-oriented programming paradigm, can be considered to be established. UML is an open standard controlled by the Object Management Group (OMG). Only recently, UML version 2.0 became the actual standards. Beside changes in the meta-model, three new views have been introduced, resulting in 13 diagram types[3], in particular:

- The **class diagram** coping with classes, features, and relationships is considered the most popular diagram of UML.
- To align interactions between objects – emphasizing the sequence – the **sequence diagram** is used. For emphasizing the links to be used by interactions, the **communication diagram** is needed. In addition, UML 2.0 introduces the ***timing diagram*** to cope with the timing aspect of interactions more sophisticated then possible so far. Furthermore, an ***interaction overview diagram*** was introduced.
- The procedural and parallel behavior of classes is dealt with in the **activity diagram**. The **state machine diagram** shows how events change the interior states of an object. The events can be triggered by interactions or by internal changes.
- UML 1.x only dealt with components as a sort of class of classes using the **component diagram**. A very important additional feature of UML 2.0 is the new ***composite structure diagram*** dealing with runtime decomposition of a class. It allows taking a complex object, such as a federate, and breaking it down into parts. They can be used to explore run-time instances of interconnected instances collaborating using the links established before.
- Interactions of users – or other systems using its functionality – with the system are caught using the **use case diagram**.

In summary, the UML standard allows to cope with a dynamical description of a system beyond the semantic level. UML enables us to describe, how a system and its components interact externally as well as internally. As pointed out in [2], to reach meaningful interoperability it is not sufficient to understand the meaning of data to be exchanged, but also to know how to use it. The same piece of data used by two systems to simulate its use may lead to completely different results, based on the use of this information in the systems. UML enables the application of white box techniques to cope with the behavior of the model without having to reveal all the internal details of the implementation.

This is critical for the credibility of the resulting solutions. As shown in many papers dealing with Validation & Verification (V&V) before, a federation of valid components is not necessarily valid itself. In order to find out, which components are composable and can be orchestrated more information than just the interface description is needed.

However, the simulation industry has a commercial interest not to make its implementation openly available. The use of UML as a standard means of documentation can be a viable compromise. In addition, the XML Metadata Interchange (XMI) format enables the transfer of UML models via XML, which allows extending the WSDL data with a description of the necessary internal representation in order to achieve interoperability on the pragmatic/dynamical level of interoperability.

There are discussions going on if UML itself is sufficient to deal with these challenges, or if other standards, such as the Discrete Event System Specification (DEVS) formalism are necessary. The actual state of this discussion is somehow summarized in [12]. The author is convinced that DEVS can be expressed using UML, but to his knowledge, no formal proof of this thesis has been given since the definition of UML 2.0.[4]

To summarize this subsection, UML has the necessary potential to cope with many questions of composability and orchestration of components. The UML

[3] The missing diagram types not dealt with in this context are deployment diagram (how or classes deployed to nodes), object diagrams (configurations of instances of classes) and package diagrams (compile-time structure of classes).

[4] Hild's dissertation is coping with challenges [13], but is not coping with UML in its actual version 2.0, as this wasn't available yet during the time of his analyses. However, most of his observations of his work are still valid.

description can be translated in XML using XMI, so that this description can become metadata on the internal behavior model of a component to be used to extend WSDL.

3.5 MDA

The Model Driven Architecture (MDA) itself is a family of standards and is described in earlier papers in a similar context already [1, 5, and 10].

The main objective of the developers of MDA is the ability to derive code from a stable model as the underlying infrastructure shifts over time. In other words, the model of the application is captured in an implementation and platform independent language. The specification of this core model is based on the established OMG standards UML, the Meta-Object Facility (MOF), and the Common Warehouse Metamodel (CWM). The three standards are tightly connected. Knowing the UML, learning MOF and CWM is relatively easy as the similarities outnumbers the differences of the three approaches. This core model of the application is defined as the *Platform Independent Model* (PIM). It can be interpreted is a general solution model meeting the operationally driven requirements of the final customer of the system.

When applying the MDA to develop software, choosing the target platform is the next step in the process. While the PIM copes with the general components, algorithms, and data to solve a given problem, the next model in the hierarchy is dealing with the problems of the implementation. This model is defined as the *Platform Specific Model* (PSM). Contrary to the unique PIM, several PSM exist to solve a given problem. For the most often used middleware solutions in the domain of the OMG, standards to map a PIM to a respective PSM are already under development. Respective middleware solutions are, e.g., CORBA, XMI/XML, DOTNET, and JAVA. The derivation of source code from the PSM normally can be done tool driven. In general, the PSM will be a UML model which takes the specifics of the chosen middleware solution and the target platform into account. The compilation of the models and the assembling and binding can be done automatically as well. In summary, this application of the MDA mandates the use of well-defined patterns for system design. Such software-design principles hold true both within large applications and across interconnected applications running across the Internet.

The applicability of the MDA to M&S already has been shown and applied in a commercially viable manner by IT partners supporting the M&S industry. The ideas published by the Australian based company Calytrix Technologies Pty Ltd are used as an example [14]. Their Integrated Development Environment (IDE) called SIMplicity implements the core concepts of MDA. At the user interface level, an IDE like SIMplicity presents the developer with a modeling environment to specify the Platform Independent and Platform Specific models for their simulation, applying UML notation wherever applicable. The modeling process supports the developer through the design, implementation, and execution phases of the simulation development life cycle. From this model, a code generation engine is employed to automatically create all the integration and component stub-code required to support the simulation design on the targeted Platform Specific middleware. SIMplicity uses the MDA design and development process. M&S components can be derived for various platforms and middleware solutions, such as the various Runtime Infrastructure (RTI) derivatives 1.3 NG or IEEE 1516, or the generation of a necessary Distributed Interactive Simulation (DIS) protocol access layer.

In other words, the technical approach already has proven to be flexible and useful for rapid integration. Success stories posted on the website and presented on various recent workshop prove the applicability of the MDA idea in the context of M&S infrastructures and common integration framework.

However, there is an important management aspect to the use of the MDA as well, which has been pointed out to the U.S. Air Force in [1] and which will be dealt with in more detail in section 4.3 of this paper. The main idea is to use Platform Independent Models composed to a Common Shared Model of the Joint and Combined Mission Space, i.e., the application domain, to ensure conceptual interoperability.

3.6 New Developments: WSFL, WEWS, WSOL, BPEL4WS, and Alternatives

Although UML and MDA are applicable in the web-based domain, they are meta-standards coping with general issues of system and software design. In the recent past, web-based open-standards dealing with specific sub-challenges emerged that are worth to look at. In particular, the proposed Work Service Flow Language (WSFL), Workflow Environment for Web Services (WEWS), and the Web Service Offering Language (WSOL) are of interest, as for example presented by Lopes and Hammoudi in [15] and Tosic et al. in [16]. The general challenge to face with these approaches is the attempt to compose and orchestrate web services for more complex tasks.

Web services are proven a very flexible and technically highly interoperable backbone for exchange of data and invocation of services. However, when aiming at modeling complex processes in agile environments through service composition, the web service family of standards lacks the necessary logic-temporal constraints as managed by matured workflow management systems. To this end, Lopes and Hammoudi are using the MDA to formulate a web service metamodel utilizing WSFL as introduced by Leyman in [17]. WSFL supports composition of services. The composite is executed based on the defined flows. Using the metamodel as a guidance for the PIM development, the MDA now allows the derivation of orchestrated web services. The resulting environment is called WEWS. WEWS combines the web service standard family (XML, SOAP, WSDL, and UDDI) with workflow based on WSDL utilizing the repository idea based on PIM of MDA.

The ideas of Tosic et al. [16] are adding the concept that not everything of a web service is always needed in every dynamic execution environment. To cope with the constraints of offering only some of the available web services in a dynamic environment, they are recommending the use of WSOL. This concept is closely related to the composite diagram ideas introduced in the section on UML. The author considers WSOL very valuable when web services are used beyond toy world examples. The whole idea can be compared to setting up a FOM based on an existing extensive SOM. As not every offering of a simulation is used in every federation based on the HLA, there is no need to assume that this will be the case in a web-based environment.

The Business Process Execution Language for Web Services (BPEL4WS), actually specified in Version 1.1 by the OASIS group, is one of the most mature approaches in this domain [18]. BPEL4WS focuses on specifying the common concepts for a business process execution language, which form the necessary technical foundation for multiple usage patterns including both the process interface descriptions, required for business protocols and executable process models.

As normal in environments defined by emerging ideas, these examples are accompanied by several alternatives in various states. However, the trend becomes perceivable that the standard family is extending to higher levels of interoperability in the LCIM, and that the MDA is increasingly used to orchestrate and align not only the execution, but also the development of necessary standards.

The complete section 3 can be summarized as follows: Web services are maturing becoming a technical backbone for interoperability up to the syntactic level of LCIM. To achieve semantic level, domain-specific solutions, such as the DDMS for the military application domain, are needed in addition. In agile environments, components must be aligned on the pragmatic/dynamic level as well. This can be done by UML. The necessary information about a component can be captured using metadata, including the use of XMI to cope with dynamic aspects dealt with using UML. The PIM concept of the MDA allows embedding heterogeneous implementations into a common conceptual view to ensure conceptual interoperability. New web-based open standard proposals are utilizing the concept and are complementing the actual technology focused solution with a necessary management view.

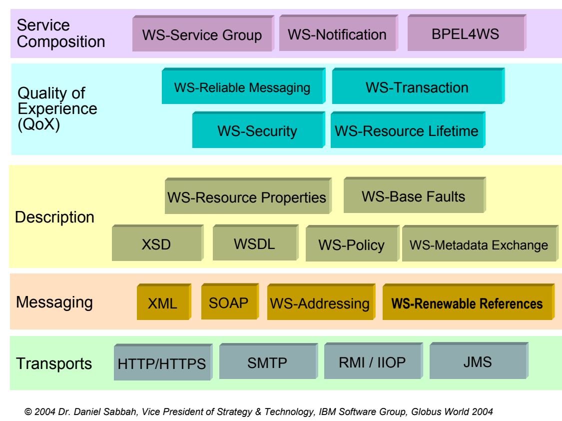

Figure 2: Extended Web Service Stack

Similar ideas are also discussed in the web service community. While until recently the web service stack just comprised the basic standards, new views – as shown during a recent Globus World Conference by the Vice President of Strategy & Technology of IBM, see Figure 2 – are supporting the view of the necessity of reuse, composability, and orchestration.

Many subsets of this framework are already in place. One example for various similar developments in the IT and M&S community is, that the TENA Architecture Management Team is looking to transition from TENA Description Language TDL to XMI as the standard format for object models. However, so far no agreement for an overarching view of these developments has been established and no framework orchestrating the developments is in place. To show the necessities and benefits, the following section will deal with some aspects of this management view for the military application domain.

4 Recommended Application and Orchestration of the Web-based Open Standards

So far, this paper was intended to give an overview of applicable standards and how they are interrelated. We will now shift the focus and look at what has to be done using these ideas in the military domains, as envisioned in papers such as [1, 2, 5, and 10].

4.1 Namespace Management

As mentioned before, the namespace management for DoD is aligned by DISA and the initial related process are described in section 3.3.

However, as already described in [5], to enable the efficient support of Joint Command and Control (JC2) using M&S functionality, a common namespace of real world DoD operations and simulated military operations is necessary. This leads to the first recommendation:

Every M&S system or component of operational relevance must participate in the DDMS. To this end, the interface must describe the input and output data using XML, either using DDMS registered tag sets or registering the tag sets with the necessary metadata at the DDMS. To this end, it is essential that projects check existing tag sets before they create their own.

Without namespace management, XML is just another way to lurch towards Babel. Using different tag sets leads to the same problem as using different data models in the data base world. Therefore, namespace management goes beyond simply registering tag sets. It is the active management of XML documents, mapping of alternative XML tag sets (utilizing, e.g., the Extensible Stylesheet Language: Transformations – XSLT, see section 4.2 for details), versioning, and more. As already pointed out in [2], the M&S community has a lot to contribute to the improvement of data and metadata to be registered in the process of namespace management. Active participation must be a mandate. In summary, namespace management based on XML tag set standardization – or better said, standardization on the management of the respective metadata to enable mapping of various application specific tag sets –, is the first set and a necessary but not sufficient requirement for a common repository as envisioned in this paper.

4.2 Data Engineering

In particular the C4I forum of SISO focused on the aspects of data engineering in the recent past. It is defined, among other papers, in [1]:

Data engineering is dealing with the questions *what* data is located *where*, the *meaning* of data and its *context*, and into what *format* the data have to be transformed to be used in respective distributed applications within the overall system. To this end, four disciplines are part of data engineering.

- *Data Administration* is the process of managing the information exchange needs that exist within a group of systems, including the documentation of the source, the format, context of validity, and fidelity and credibility of the data. Data Administration therefore is part of the overall information management process. In particular the UDDI ideas can help to facilitate this discipline.
- *Data Management* is planning, organizing and managing of data by defining and using rules, methods, tools and respective resources to identify, clarify, define and standardize the meaning of data as of their relations. This is a real management task and can only be facilitated by agreeing on a common reference model.
- *Data Alignment* ensures that the data to be exchanged exist in the participating systems as an information entity or that the necessary information can be derived from the data available, e.g., using the means of aggregation or disaggregation. If a common reference model is used and this model as well as the model to be aligned are using XML tag sets, data alignment is facilitated to the proof that a complete XSLT schema exists that transforms the source into the sink.
- *Data Transformation* is the technical process – actually usually implemented by respective proprietary mapping algorithms designed by the individual programmer within the gateways and interfaces – of aggregation and/or disaggregation of the information entities of the embedding systems to match the information exchange requirements including the adjustment of the data formats as needed. In case of agreeing to use the web based approach proposed in this paper, data transformation is the application of the XSLT schema generated by the data management for the data alignment and the connection can be done using the UDDI entries generated by the data administration.

Although data engineering in the current form only ensures interoperability on level 3 of the LCIM, this is sufficient when one of the partners is not agile but – at least from the information standpoint – more or less static. If no dynamics have to be taken into account, levels above level 3 of the LCIM are obsolete. Although many C4I system developers will not agree necessarily, the C4I systems of the recent past can be seen as very secure and highly flexible distributed databases tracking the status of the various sensors and their reports as well as the derived actual perceived situation. However, we are still living in the era of the search for the "Common Operational Picture," i.e., a description of the actual situation, and sometimes the recent statuses, that led to this situation. In order to be able to cope with the agile battle sphere, however, a dynamic approach in form of a "Common Operational (executable) Model" is necessary, as recommended in by Daly and Tolk in [5].

This implies the need for the development of a Common Shared Model of the Joint and Combined Mission Space, which is the topic of the next section. As pointed out before, this model can be described using the methods of the MDA, generating the necessary metadata which can be incorporated into the repository management.

4.3 Developing of a Common Shared Model of the Joint and Combined Mission Space

That UML can be used to describe dynamics of agile models should be obvious, in particular when M&S formalisms like DEVS are used to enrich the actual set of diagrams (see section 3.4). These methods can be used to reach level 4 of interoperability in the LCIM.

The idea to set up a common reference model of the relevant military operations and participating operational elements is not new. The use of open standard software engineering approaches to do so, however, is relatively new. Blake et al. introduced a UML based model of single ship operations to align the various related projects of the U.S. Navy in [19]. The main idea is to model the real world as good as possible in UML to have a reference for the M&S applications. This model shows where the M&S solutions have shortcomings or differ in their worldview. Concerning the author, this is the first step to develop a common conceptual model as needed to enable the level 5 interoperability in the LCIM.

In order to realize the vision of reuse, composability, and orchestration of M&S components forming a web-based repository, the following steps are recommended by the author:

1. Every M&S application must identify its information exchange requirement in form of XML descriptions. This does not imply that later on information has to be exchanged only in form of XML documents; however, this is mainly a management instrument to enable to handling of this information in a standardized way. If two applications realize that they can share information and they decide to establish a high-performance connection not based on XML, this is not a problem. For management purposes, XML documentation must be mandatory anyhow. The SEDRIS domain is a good example for this proposal: Although the information exchange is likely to be more efficient using the SEDRIS format, the integration of SEDRIS ideas into the broader concept of, e.g. the GIG, requires the use of the standards of the respective domain, which is XML, as in many other domains.[5]
2. To facilitate data administration, the UDDI idea must be utilized. All applications should register using the ideas of an enhanced WSDL. In order to support true heterogeneity, communication methods other than web services should be included as well; however, the XMSF group favors web based communication backbones based on open standards, and so does the author.
3. Common data management based on the XML documentation of the interfaces enables a much easier data alignment than ever has been possible before. In addition, the results can be directly applied in form of XSLT schemas that can be used to translate one XML dialect into another based on the common namespace management.
4. In addition to data engineering, behavior representation must be conducted as well. This should be done using the PIM concept of MDA. Every component must come with a PIM explaining the behavior of the component as a white box. As the PIM doesn't reveal implementation details protected as intellectual property of the implementing developers and their companies and in the same approach enables interoperability on level 4 of the LCIM, this approach as recommended by the author. These data must be part of the metadata of the component and must become part of the enhanced

[5] This is in particular true for the Environmental Data Coding Specification (EDCS) of SEDRIS, which provides the entity-level semantics in form of several dictionaries for the schemas used in the Data Representation Model (DRM).

WSDL. New efforts, such as WEWS, increase the applicability of web services beyond reuse and composability towards orchestration.

5. Finally, all PIMs must be embedded into a common description of the mission space to become the conceptual reference model. This model can be used to identify gaps (no model delivers the necessary functionality) as well as multiple fulfillment (multiple models deliver the functionality, maybe using different completing viewpoints).

The standard use case is now the following: A user describes his problem using the methods of the MDA, applying the design pattern principles described in section 3.5 for implementation for reverse or re-engineering. In this application of the MDA, a standardized description of the operationally necessary functionality in form of the PIM is derived. The component user than evaluates the resulting conceptual model of the component to find out if his functionality is already fulfilled by other components. Then, he selects the most appropriate components. Via UDDI he locates the components which can be executed remotely, while their execution is orchestrated via WEWS (or an alternative) and the information is automatically translated using the XSLT schemas generated by the data engineering efforts. In particular, all this can be done within web-based C4I devices. In the near future, this concept can be used to define M&S services for the GIG.

5 The Role of SISO

SISO is a not for profit organization that focuses on facilitating simulation interoperability and component reuse across the DoD, other government, and non-government applications and seeks to serve the broad Modeling and Simulation (M&S) community. If it comes to M&S standard, SISO is the organization to go to. It is therefore natural that SISO must be actively engaged in the processes described. Three examples, that are neither complete nor exclusive, shall be given.

5.1 V&V

The role of Verification and Validation and necessary standards to support respective processes is dealt with in the VV&A forum of SISO. The vision proposed in this paper will immediately support a more general applicability of V&V in the future.

As already pointed out before, we lived in the conflict of valid security issues concerning intellectual property on side of developers and need for transparency on side of the users, in particular when M&S is used for support of military operations. We also saw that the introduction of the PIM by the MDA can bridge this gap. However, this can only be done if escorted by reliable VV&A.

Validation is defined as the process of determining the degree to which a model or simulation is an accurate representation of the real world from the perspective of the intended uses [20]. It therefore deals with the Behavioral or Representational Accuracy and has to answer questions such as “Is my conceptual model correct?” “Am I modeling the right thing?” and “Can I answer the questions of the user?” The same set of questions has to be answered by the Platform Independent Model (PIM) of the MDA. When using the MDA for software engineering, this is a natural step: Using the PIM for Validation.

Verification is defined as the process of determining that a model or simulation implementation accurately represents the developer’s conceptual description and specifications [20]. It deals with the Transformational Accuracy and answers questions such as “Is my conceptual model correctly transformed into code?” “Am I modeling the right thing correctly?” and “Do I meet all the specifications?” The same set of questions has to be answered when transforming the Platform Independent Model (PIM) into the Platform Specific Model (PSM) and then into code. Verification can be tightly coupled to the application of the transformation rules.

To summarize, when developing software using the MDA V&V should be mapped to validating the PIM and verifying the transformation rules resulting in the PSM and the code. Validation assures an accurate PIM, Verification assures an accurate PSM.

In addition, both processes can be applied in the context of software reengineering as well. It is possible to re-engineer platform specific solution, such as legacy components or simulators and simulation systems, into a platform independent description of its functionality. In this process, V&V gets a new role: V&V is no longer limited to the development process, but it is integrated into the process of reuse, composability, and orchestration. Verification assures that the derived PSM as well as the resulting PIM are correct, and Validation evaluates if this PIM fits into the overall picture and really fulfills the user requirements. Furthermore, these processes are supported by standard tools and methods.

As stated out before, the paper proposes the V&V escorted development of a common description of the mission space to become a common conceptual reference model. Such a task can only be handled by an independent group such as a study group of SISO; otherwise, a conflict of interest is likely. Moreover, if these efforts really are leading to a common model with different views on the common mission space, the configuration and enhancement must be orchestrated with all participating organizations, and there number is increasing in the light of Homeland Security applications and similar new challenges for M&S.

Finally, another interesting approach is to use an extensible XML-based framework to provide a bridge between forthcoming M&S requirements and open/commercial web standards, as proposed by Broyles in [21]. He migrated the Navy Modeling and Simulation Office handbook, which will ensure the credibility of M&S based on application of the process of Verification and Validation (V&V) by defining respective processes and supporting methods, into an XML-based language. The framework for the resulting V&V Markup Language (VVML) implements all templates defined in the handbook. It is more than likely that VVML can be used to define additional metadata necessary to enable efficient V&V for the PIM.

5.2 XMSF Profiles

The XMSF Study Group of SISO is actually working on the definition of XMSF Profiles. The result so far is the following:

XMSF profiles are formal technical specifications for application of interoperable web based technologies enabling composable and reusable modeling and simulation, and facilitating enterprise integration. The objectives of XMSF profiles are to

- *Provide unambiguous specification of the functionality of components, and interfaces among components of the framework*
- *Ensure interoperability between existing and new web enabled technologies, both within M&S and in related domains*
- *Provide the necessary metadata to facilitate composability and reuse of components across multiple M&S application domains*
- *Facilitate development of new applications and services that are functionally interchangeable with existing applications and services*
- *Enable development of new applications and services that readily extend functionality for continuous evolution of capabilities.*

Beside improvements of this working definition, the group is actually working on concepts of operations. Concerning the author, the group can definitely contribute the vision described in this paper and the web-based repository comprising reusable and composable elements that can be orchestrated is a valuable application case. Furthermore, the profiles should be applicable to describe M&S components in a way that they can be integrated into net centric environments, such as the GIG, as described in [5].

5.3 Web-based M&S (WebSim)

The last activity is the recently launched Web-based M&S (WebSim) initiative [22]. WebSim is a common effort of the Object Management Group (OMG), The Open GIS Consortium (OGC), SISO, and the Web3D Consortium with the objective to align their standardization effort concerning web-based M&S. The four partners realized that, although everyone has a legal interest in specific solutions, the alignment of the core activities is essential to ensure interoperable solution by integration from the initialization instead of alignment as an aftermath of domain specific implementations. SISO's role is defined to be the standardization organization for the M&S domain. The author hopes that the collaboration will be intensified in the near future and mutual benefit for all partners becomes obvious. The tremendous potential became obvious to all participants in the First Annual Workshop on Web-Based M&S, which was conducted in October 2003 in Reston, VA.[6] In the opinion of the author, such a common effort is a prototype for future co-operations necessary to ensure inter-agency interoperability for joint and combined operations as well as for Homeland Security applications.

There are more domains affecting the vision and the development of necessary standards beyond those described in this paper. Web services, however, are becoming a unifier for heterogeneous IT solutions, no matter if various legacy systems should be migrated or if the gap between heterogeneous platforms (among them modern Grid solutions as well as legacy mainframes) or between operating systems has to be bridged. The more organizations participate, the more important becomes the role of SISO as the deputy for M&S interests in the process of future standards.

5.4 Engineering Standards and the LCIM

The following table shows a first mapping of a subset of standards described in this paper to the improved LCIM as described in section 2. As pointed out before, web standards are just one way of implementation, but they support interoperability issues on the management and V&V level as well. Nonetheless, alternative

[6] The workshop proceedings can be downloaded from [22] and http://www.vmasc.odu.edu/wscc/wscc.html.

standards can and should be taken into account. The MDA can serve as a common denominator, as every standard line can be mapped to a respective PIM and PSM, as showed in [1].

Table 1: LCIM and applicable Standards

LCIM Level	Examples for Standards
Technical	TCP/IP, HTTP, SMTP, IIOP
Syntactic	HLA OMT, PDU, XML, SOAP, WSDL
Semantic	C2IEDM, PDU, RPR FOM, DoD XML Tags
Pragmatic/Dynamic	UML, WEWS, MDA, DEVS
Conceptual	DoD Architecture Framework, UML, MDA, DEVS

This mapping is just a tentative beginning and definitely must be improved by taking results such as described in [23] into account. They should also be group in groups of family standards, such as web services. Furthermore, in particular on the conceptual level, the standards are necessary but not sufficient. A study group dealing with this issue could be of help to get this organized.

6 Summary

This paper shows how the consequent application of web-based open standards in an aligned and harmonized manner can be used to make the vision of a web-based repository of composable M&S applications a reality now, not only in a distant future. We already have in hand what we need to start these processes.

The approach proposed in this paper is for the data exchange interface to be described using XML, where tag sets are managed in the context of data engineering, using the ideas of DDMS. XSLT can be used to map the various aligned tag sets. Using UML and the other methods of the MDA, the component or application can be described in the form of a white box without violating intellectual property rights of the developer. These descriptions can then be used to enrich the actual version of WSDL to comprise not only the data exchange component, but also the behavior representation of the participating components. The use of common reference models enables the gradual development of a common conceptual model, the Common Shared Model of the Joint and Combined Mission Space. It is not necessary to initiate all processes described here at once, but a gradual introduction of required documentation can be chosen as well. It is, for example., possible to start with requesting to document all information exchange request in XML and enrich the namespace with this definition without requiring the PIM of the MDA. To support all levels of interoperability as defined in the LCIM, however, an overarching approach as described in this paper is necessary. It is topic of ongoing research if this will be sufficient as well.

A web-based repository of reusable and composable elements that can be orchestrated to fulfill a user requirement driven task is an idea who's time has finally come with web services, although they have to be enhanced and extended to meet the needs of M&S. Very useful hints can be found in earlier works, which simply were ahead of their time. In particular, these ideas are principally outline by the recent "Reuse Library Interoperability Group." Many of their ideas can be reused in this updated context.[7] An extended evaluation of this work can avoid the reinvention of the wheel in many domains. Of particular interest should be the recommended data/metadata model developed to support interoperability, reuse, and composability (although the orchestration part is not explicitly mentioned in their reports.)

The actual debate on a future direction is fueled only in part by science; the other part is distinct philosophy about how M&S should be done and if open standards are applicable or better said should be applied anyhow. Many M&S expert still are very comfortable in their niche of knowledge, and the application of open standards would force us out of our ivory towers and toy worlds to face real problems in the real world. However, the real world is the only final test enabling the process of pruning irrelevant ideas and models. Open standards, in particular web and web service based standards are the actually most promising direction to achieve the goal to bring M&S to the Warfighter to support his operations as well as, in a more general sense, bringing M&S to the user to solve his problems on a daily basis.

Acknowledgements

The research work underlying this paper was funded by the Defense Modeling and Simulation Office (DMSO), the U.S. Air Force Joint Synthetic Battlespace (JSB-AF) Initiative and General Dynamics – Advanced Information Systems (GD/AIS). The work described in this paper is part of the XMSF initiative conducted by the Naval Postgraduate School, George Mason University, Old Dominion University, and SAIC, and supported by sponsors in government and industry.

[7] The best overview known to the author is given in [24].

7 References

[1] Don Brutzman, Andreas Tolk: "JSB Composability and Web Services Interoperability via Extensible Modeling & Simulation Framework (XMSF), Model Driven Architecture (MDA), Component Repositories, and Web-based Visualization," JSB ATA Study for USAF, November 2003

[2] Andreas Tolk, James Muguira: "The Levels of Conceptual Interoperability Model (LCIM)," Fall Simulation Interoperability Workshop, Orlando, FL, September 2003

[3] Bernard P. Zeigler during the Panel Discussion on "Priorities for M&S Standards," Spring Interoperability Workshop, Orlando, FL, March 2003

[4] Marko A. Hofmann: "Essential preconditions for coupling model-based information systems," NATO M&S Group (NMSG) Conference on C3I and M&S Interoperability, RTO-MP-123, Antalya, Turkey, October 2003

[5] John J. Daly, Andreas Tolk: "Modeling and Simulation Integration with Network-Centric Command and Control Architectures," Fall Simulation Interoperability Workshop, Orlando, FL, September 2003

[6] Paul Gustavson, Larry Root, Steve Goss, Jane Bachmann, Mark McAuliffe: "CODE-Net: XML, SOAP and Simulation Development," Spring Simulation Interoperability Workshop, Orlando, Florida, March 2002

[7] Katherine L. Morse, Ryan P.Z. Brunton, David L. Drake: "WMDOA Integration Employing Web Services for Federate Communication - an XMSF Exemplar," Spring Simulation Interoperability Workshop, Orlando, Florida, March 2003

[8] Eric Newcomer: "Understanding Web Services – XML, WSDL, SOAP, and UDDI," Addison-Wesley, 2002

[9] U.S. Department of Defense: "DoD Metadata Registry and Clearinghouse," online specification at http://xml.dod.mil/mdreg/user/index.cfm

[10] U.S. Department of Defense, Deputy Assistant Secretary of Defense: "Department of Defense Discovery Metadata Specification (DDMS) – Version 1.0", Washington, D.C., September 2003

[11] Andreas Tolk, Mark Pullen: "Ideas for a Common Framework for Military M&S and C3I Systems," European Simulation Interoperability Workshop, Stockholm, Sweden, June 2003

[12] Paul K. Davis, Robert H. Anderson: "Improving the Composability of Department of Defense Models and Simulations," RAND Corporation, 2003

[13] Daryl Ralph Hild: "Discrete Event System Specification (DEVS) Distributed Object Computing (DOC) Modeling and Simulation," Dissertation, University of Arizona, 2000

[14] Shawn Parr, R. Keith-Magee: "Making the Case for MDA," Fall Simulation Interoperability Workshop, Orlando, FL, September 2003

[15] Denivaldo Lopes, Slimane Hammoudi: "Web Services in the Context of MDA," 2003 International Conference on Web Services (ICWS'03), Las Vegas, NV, June 2003

[16] Vladimir Tosic, Bernard Pagurek, Babak Esfandiari, Kruti Patel: "On the Management of Compositions of Web Services," Workshop on Object-Oriented Web Services (OOPSLA 2001), Tampa, USA, October 2001

[17] Frank Leymann: "Web Service Flow Language (WSFL 1.0)," IBM, May 2001

[18] Tony Andrews et al.: "Business Process Execution Language for Web Services, Version 1.1," March 2003, http://xml.coverpages.org/WS-BPELv11-20030331.pdf

[19] Donna W. Blake, Carolyn Little, Judy Morse: "The Navy's Probability of Raid Annihilation Assessment Process Standards & Architecture and Systems Engineering Concept Model," Fall Simulation Interoperability Workshop, Orlando, FL, September 2003

[20] DoD Instruction (DoDI) 5000.61: DoD Modeling and Simulation (M&S) Verification, Validation, Accreditation (VVA), April 1996

[21] David Broyles: "Developing an XML Architecture for VV&A; The Development of the Verification, Validation Markup Language (VVML)", Spring Simulation Interoperability Workshop, Washington, D.C., April 2004

[22] Homepage of the Web-based M&S Initiative, http://www.websim.org

[23] E. Weisel, M. Petty, R. Mielke; "A survey of Engineering Approaches to Composability," Fall Simulation Interoperability Workshop, Washington, D.C., April 2004

[24] Shirley V. Browne, James W. Moore: "Reuse Library Interoperability and the World Wide Web," International Conference on Software Engineering (ICSE 97), pp. 684-691, 1997

Authors' Biography

ANDREAS TOLK is Senior Research Scientist at the Virginia Modeling Analysis and Simulation Center (VMASC) of the Old Dominion University (ODU) of Norfolk, Virginia. He has over 12 years of international experience in the field of Applied Military Operations Research and Modeling and Simulation of and for Command and Control Systems. In addition to his research work, he gives lectures in the Modeling and Simulation program of ODU. His domain of expertise is the integration of M&S functionality into related application domains, such as C4ISR or web-based services, in particular based on open standards.