REUSABLE INTEROPERABILITY COMPONENTS IN MILITARY IT

Sandor MUNK
munk.sandor@uni-nke.hu
National University of Public Service, Budapest, Hungary

ABSTRACT
In our days the range of information activities supported by IT equipment, and the volume of information stored in IT systems continually grows. So the cooperation among different organizations is practically impossible without extensive, meaning preserving information exchange between their IT systems. Practically all today’s interoperability solutions are based on a previously agreed intermediary representation (formatted message standard, or standard data elements), but these solutions have a number of limitations. Recent paper analyses the reasons, circumstances, and conditions of the application of reusable interoperability components. For this reason: summarizes the basics of reusable interoperability components, and analyses reasons for their application; examines how reusable interoperability components fit military interoperability solutions; and finally outlines the basics of an interoperability component model.

KEYWORDS: IT interoperability, component-based software development, reusable interoperability components

1. Introduction
In a previous publication [1] we justified the need for research of reusable IT interoperability components. Significance of IT systems interoperability is particularly great in such complex systems of organisations as the defence sector (military forces, law enforcement, disaster management, etc.), the public administration. Practically all interoperability solutions of our days are based on previously agreed intermediary representations, and devolve the tasks of transformation between different information representations to the relevant IT systems, so the interoperability capabilities, functions embodied in these solutions do not become “public property”.

Another significant problem is, that the adaptation to changing cooperation environments, emerging, and changing information exchange needs, the improvement of interoperability capabilities are limited, and time-consuming. This requires application of state-of-the-art – among others component based, service oriented, or middleware – IT development methods, and solutions. The purpose of recent paper is to analyse the reasons, circumstances, and conditions of the application of reusable interoperability components.
2. Basics of Reusable Interoperability Components, and Reasons for Their Application

Reusability has long been used as a means of human problem solving. Successful solutions later have been used in solving new, similar problems [2]. This applies also for IT systems, where the reuse has been used from the beginning of software development. Feasibility, benefits, application conditions of reuse depend on the characteristics of the given application area. Scientific publications primarily examine the role of interoperability in the implementation of reusability of software components. This publication is based on a different approach: discusses the role of reuse in the implementation of interoperability.

2.1. Software Reusability, Component-Based Software Development

Software reuse is the use of existing software to construct new software, its purpose is to improve software quality and productivity. The precondition is the ability to build larger systems from smaller components, and being able to identify common functions of the components used in the same, or different systems. It has its own problems, and drawbacks: higher maintenance costs, lack of support toolkit, “not invented here” syndrome, and difficulties of publishing, finding, understanding, and adapting reusable components [3, 4, 5].

The Component-Based Software Development (CBSD) is a software development methodology that is based on the separation of software system development, and software components development. Accordingly, software systems should be built up from prefabricated, existing components, and software components should be developed so that they can be used in different systems. A software component “is a unit of composition with contractually specified interfaces and explicit context dependencies only. A software component can be deployed independently and is subject to composition by third parties” [6].

2.2. Basics of IT Interoperability

Interoperability in general is a mutual capability of two or more objects to support cooperation, interoperation. The fundamental type of interoperability is operational interoperability between active actors cooperating to achieve a common goal. The preconditions are different part – apabilities, among them the key prerequisite, information interoperability: a mutual capability of different actors to ensure exchange and common understanding of information needed for their successful cooperation.

Information exchange between cooperating actors increasingly happens without human assistance, by direct data exchange between the actors’ IT systems. So IT interoperability is a mutual capability of IT systems, devices, applications (hereinafter briefly IT entities) to send, receive, exchange data preserving the meaning assigned by the primary user community. An interoperability problem occurs, when at some level there are differences, disagreement in the representations used, or in their interpretations.

The purpose of an IT interoperability solution is to resolve heterogeneity between disparate, heterogeneous IT entities, to ensure conditions of meaning preserving information exchange. Interoperability interfaces, connectors, wrappers implement meaning preserving transformations between the entity’s internal representation, and the intermediary representation used. Interoperability gateways perform transformations between different intermediary representations. The transformations can be implemented in the form of an interoperability infrastructure that is a value-added network service layer, or an independent middleware layer.

2.3. Reasons for component-based interoperability solutions

A component-based interoperability solution can be built using functional decomposition of the transformation of a complete information exchange unit into transformations of lower level composite
and elementary information units. In the following under reusable interoperability component (functional unit) we understand a hardware and/or software entity, which purpose is to implement meaning preserving transformation between different information representations.

Because most of the IT systems use the same, or similar information representation formats, and several common data element types, IT interoperability solutions, to a considerable extent, are based on identical, widely used functions. This is especially true for the syntactic level of transformations: decomposition and composition of structured, and semi-structured information representations; or conversions between different data formats, different encodings of textual information. These interoperability functions may be ideal for multi-use (reusable) building blocks.

Semantic level meaning preserving transformations are typically based on significant domain-oriented knowledge contained in ontologies describing concept systems, and related rule systems, algorithms. Expert knowledge required for the implementation of these functions is an 'expensive' resource, so its hiding in specific systems is extremely inefficient.

Finally interoperability solutions are characterized by a dynamically changing environment, and the need for rapid adaptation to changing circumstances. IT systems go through less frequently syntactic, more frequently semantic changes. In addition military IT systems often find themselves in an environment where they have to work together, exchange information with previously unknown IT systems. Component-based systems’ capabilities can be extended, improved without manufacturer-level software development. Available, assured quality interoperability components may help to adapt to new, or changed requirements within a relatively short period of time.

3. Reusable Interoperability Components and NATO Interoperability Solutions

As a part of the analysis of reusable interoperability components it is essential to compare their basic issues against existing, and proposed interoperability solutions. Based on our research goals we selected two NATO interoperability solutions: Multilateral Interoperability Program, and NATO Semantic Interoperability Logical Framework. In case of both solutions we examine how the reusable interoperability components fit into their overall framework, what kind of role they could play in them, what kind of relationships can be found between them, and whether they offer a solution to problems, which are not dealt with.

3.1. Interoperability components and the MIP

Multilateral Interoperability Programme (MIP) is an international organization that contributes to ensuring the conditions of interoperable information exchange between military command and control information (hereinafter IT) systems in multinational, coalition environments with developing a set of interface specifications and supporting documents.

The foundation of MIP solution is the MIP gateway implementing the MIP Common Interface. MIP Common Interface supports interoperable information exchange between national IT systems providing two standardized information exchange mechanism. Transformations between data formats, message formats, communication protocols used in national IT systems and standardized MIP solutions is national responsibility.

MIP Data Exchange Mechanism (DEM) supports automated, selective exchange of situational awareness database information by a replication mechanism. MIP Message Exchange Mechanism (MEM) supports transmission of operational information (orders, plans, reports,
notifications, etc.) using formatted messages and related file attachments.

In practice there are a number of semantic and syntactic differences between inner information representations of national IT systems, and standardized MIP intermediary representation. The degree of differences depends on the application area similarity of cooperating systems, and on the preliminary agreements. Heterogeneity between inner and intermediary representations cannot, or should not be completely eliminated. Inner representations should primarily support the intended purpose of the affected system.

The heterogeneity between inner and MIP representations implies, that national transformation functions play an important role. These transformations may be necessary on different levels (protocols, information structures, elementary information used, etc.). Both DEM (with structured data) and MEM (with semi-structured data) require transformations between elementary data. The components providing format-oriented, syntactic transformations can be implemented easier, and can be used widely. However components performing content-oriented, semantic transformations usually require substantial domain-oriented knowledge, and can be utilized within the given application area.

3.2. Interoperability Components and the NATO SILF

NATO Semantic Interoperability Logical Framework (SILF) is a high level (logical) framework that forms the basis of ensuring semantic interoperability between heterogeneous military IT systems that is based on the assumption that technical and syntactical levels of interoperability exist between concerned systems. It aims a solution that does not require any major changes in the existing systems, and the systems need not know anything about the other systems and their characteristics. SILF is a middleware solution that performs interoperability in the communication medium. Essential parts of SILF are semantic technologies, and the existence of semantic descriptions of data used by the particular systems [7].

From the point of view of our research goal, fundamental characteristic of the SILF solution is that interoperability is realized by a rule-based mediator (programmable gateway). Translation rules applied during the operation phase are generated in the preparation phase by ontological means, using semantic descriptions of the given systems, and a constantly evolving common semantic base.

Interoperability components in the SILF model does not appear explicitly. It does not determine the way of the implementation of the rule-based mediator, but it mentions intermediate translation/mediation components as parts of the runtime environment [8]. These components are parts of a gateway which also contains other components solving syntax-level translations, or technical problems of tactical communication networks [9].

The external events for event – condition – action type rules are arrivals of the incoming information representations, internal events are initiations of some transformation subtasks, and elementary actions are elementary transformations that can be implemented as interoperability components. Due to the limitations of rule markup languages a rule-based, declarative solution should not appropriate, or efficient in any case. So the SILF generated rules can be used for building traditional algorithmic components too.

4. Conditions of use of Reusable Interoperability Components

Interoperability components may be operational only in component-based systems, based on well-defined standards and agreements (component model), and making use of the services of a supporting infrastructure. The component model is a rule system (standards and conventions)
that defines the interaction and composition of components (forms of communication, representations of information, conditions of deployment, etc.), while the component framework is a platform (a set of interacting components) on top of which other components operate, and which provides different (e.g. communication, or resource management) services for them [10, 11].

4.1. Basics of Interoperability Component Models, Frameworks

Over time, in IT practice several different component model and related framework (infrastructure) have been developed and widely used. In essence, service oriented solutions are very similar, which are also based on interoperable, reusable – but loosely coupled – components, however they regulate only the communication between components, and do not contain specifications for the language and platforms of components.

Component models can be classified into two major groups. General component models are general purpose solutions that determine the rules of interaction and composition of components. According to their purpose in these models do not appear domain specific requirements, and they do not contain any domain specific solutions. Domain-oriented knowledge can be represented in domain-specific component models, which establish special component types and special compositional solutions in the framework of a general component model [12]. To both component models belongs an appropriate component framework, providing general purpose, and domain specific services. Essential elements of component models, and frameworks are component types and component interfaces.

Component-based solutions have well-defined architecture. The planning and development of individual solutions can be supported by general, reusable architectural patterns (logical models) designed for repetitive tasks. For each component model there is one or a few related architectural patterns that determine the basic component types, their purpose, basic functions, and the rules of their interconnections, and interactions (and the appropriate interfaces). Domain-oriented component models usually based on more complex architectural patterns that represent the specialities of the given domain.

Since our research goal is related to military IT interoperability solutions, in the following we overview the relationships of interoperability components and NATO IT frameworks. Today’s NATO IT frameworks and related documents are based on the service-oriented approach. These documents classify the individual services into three basic service area (community of interest, core enterprise and communication services), and within them into service categories. Interoperability components (may) appear in all the three service areas: in the first two on semantic and syntactic level, in the third, on technical level. Currently in the relevant NATO documents the core enterprise [IT] services category contains services related to our research topic.

NATO C3 Classification Taxonomy contains composition services, and mediation services in the SOA Platform Services category. These services use semantic models, either explicitly, or implicitly [13]. Interoperability solutions also appear in the catalogue of the NATO Interoperability Standards and Profiles that contains translations services in the mediation services category. These services manipulate messages in-flight between a service provider and a consumer. They make protocol, and message content transformations, augment messages by adding information from external data sources, and derive complex events from message and event streams [14].

4.2. Outlines of an Interoperability Component Model

The essential condition of a widely usable interoperability solution, based on
reusable components, is a commonly accepted domain-oriented interoperability component model and a supporting component framework (infrastructure). In practice such a model with the same functionality can be implemented built on different generic component models.

In a previous publication [15] a comprehensive model of an interoperability solution has already been presented that is based on three basic interoperability functional unit types: decomposition of the source representation into elementary components, meaning preserving transformation between elementary components, and composition of target representation from elementary components.

Based on the above model interfaces of basic interoperability component types at high level can be standardized. The proposed generic interface for all three types contains a unique provided interface, optionally one (or more) required interface, and a single operation. The interface includes one or more input information representation with their descriptions; one or more output information representations with their description; and the status result of the execution of the interoperable transformation function.

For the purpose of general applicability, information representations are bit strings without any format specification. Descriptions of information representations minimally contain a universally unique ID of representation type (UURTID), and optionally additional descriptive characteristics. UURTID uniquely identifies the content (meaning) of the information conveyed, and its format, including its structure. Additional characteristics may serve later, or special application purposes.

Cooperation of interoperability components based on the proposed interface ensures the implementation of any interoperable transformation tasks. Transformations between elementary information representations can usually been performed without the services of other components. In case of complex information representations, the interoperability component implementing the required transformation contains a control structure activating ‘simpler’ decomposition, transformation, and composition components. The interoperability component fitting into the above model can be a documented executable format software component, made available for use, or a service with a self-contained functionality in a service-oriented architecture (e.g. web service).

In addition the interoperability components may have additional functions and interfaces too. One such example is the recognition of the type of a given representation that can be useful in case of handling representations which are not documented (e.g. derived from foreign sources). This functionality determines the type, or possible types of one, but preferably a set of information representations (e.g. derived from a database).

5. Summary, Conclusions

Because IT interoperability solutions based on pre-agreed intermediate representations provide an effective solution only in case of a well-defined, long term, and close cooperation; their implementation and maintenance requires extensive coordination and significant time; they respond hardly and slowly to the new information exchange requirements; it is justifiable to examine the possibilities, and conditions of the application of modern software development approaches. One of these approaches is the component-based software development and the closely related service-oriented approaches.

Analysing the NATO interoperability solutions it can be stated that the MIP interoperability solution does not provide any support to national information transformation functions, so they should be implemented as independent, national responsibility interoperability solutions.
The SILF interoperability solution also does not address the implementation details of the semantic mediator performing interoperable transformations. The rule-based SILF approach indirectly involves the need for components performing elementary transformations, in addition allows the creation of interoperability components using translation rules.

Interoperability components can be inserted into the framework of both solution. Under appropriate conditions these components can be used more than once, e.g. re-used. The application of component-based implementation, similarly to other application areas, can be justified. For this purpose we propose a domain-specific interoperability component model that fits the existing general component models, or service-oriented architecture. The model is based on decomposition, transformation, and composition components. Furthermore we propose a common generic interface which contains information representations and their descriptions (identifiers). This unique standardized interface supports the interoperation of interoperability components, and the implementation of any interoperability function.

REFERENCES

15. Munk Sándor, cit.ed. 42.
BIBLIOGRAPHY


