

2008 Command and Control
Research and Technology Symposium
“C2 for Complex Endeavors”

**Community-of-Interest (COI) Model-based Languages
enabling Composable Net-centric Services**

Andreas Tolk, Ph.D.
Old Dominion University
241 Kaufman Hall / EMSE
Norfolk, VA 23529, USA
+1.757.683.4500
atolk@odu.edu

Saikou Y. Diallo
Virginia Modeling Analyses & Simulation Center
1030 University Blvd
Suffolk, VA 23435, USA
+1.757.638.6320
sdiallo@odu.edu

Cite as: Tolk, A. and S.Y. Diallo (2008). Community-of-Interest (COI) Model-based Languages enabling Composable Net-centric Services. *Proceedings of CCRTS 2008: the Command and Control Research and Technology Symposium*, June 17–19, Bellevue, WA, USA.

Community-of-Interest (COI) Model-based Languages enabling Composable Net-centric Services

Andreas Tolk, Ph.D.
Old Dominion University
241 Kaufman Hall
Norfolk, VA 23529, USA
+1.757.683.4500
atolk@odu.edu

Saikou Diallo
Virginia Modeling Analyses & Simulation Center
1030 University Blvd
Suffolk, VA 23435, USA
+1.757.638.6320
sdiallo@odu.edu

Abstract

Net-centric services shall be designed to collaborate with other services used within the supported Community of Interest (COI). This requires that such services not only be integratable on the technical level and interoperable on the implementation level, but also that they are composable in the sense that they are semantically and pragmatically consistent and able to exchange information in a consistent and unambiguous way.

In order to support Command-and-Control with Composable Net-centric Services, the human-machine interoperation must be supported as well as the machine-machine interoperation. This paper shows that techniques of computer linguistic can support the human-machine interface by structuring human-oriented representations into machine-oriented regular expressions that implement the unambiguous data exchange between machines. Distinguishing between these two domains is essential, as some requirements are mutually exclusive. In order to get the “best of both worlds,” an aligned approach based on a COI model is needed. This COI model starts with the partners and their respective services and business processes, identifies the resulting infrastructure components, and derives the information exchange requirements. Model-based Data Engineering leads to the configuration of data exchange specifications between the services in form of an artificial language comprising regular expressions for the machine-machine communication. Computer linguistic methods are applied to accept and generate human-oriented representations, which potentially extend the information exchange specifications to capture new information not represented in the system requirements.

The paper presents the framework that was partially applied for homeland security applications and in support of the joint rapid scenario generation activities of US Joint Forces Command.

1 Introduction

It seems obvious that effective and efficient collaboration must be based on effective and efficient communication between participating organizations and individuals. However, current research often addresses only parts of the overall problem that spans more than one domain. To support efficient communication between organizations and systems, it is essential to understand the overall context of the information exchange and assess existing solutions in the affected domains with respect to their usability in other domains. This paper introduces a framework that allows one to specify a domain, analyze existing solutions, and show how they can play together in support of a more general solution. It is strongly motivated by the NATO Code of Best Practice for Command and Control Assessment (NATO, 2001), in particular by the recommendation not to rely on one model or tool to solve a complex problem, but to apply an orchestrated set of tools to evaluate the different relevant aspects and facets of a problem.

Communication in the Net-centric environment is without a doubt a complex problem. It seems to be logical to divide the overall problem into a set of sub-domains in order to make the problem more understandable and manageable. This results in a set of solutions that when combined solve the original problem. The idea behind the proposed framework is to base the orchestration of solutions on a common model encompassing the set of sub-domains and treat contributing solutions as views on this common model. The common model connects the human components – the members of the Net-centric organizations and users of the supporting systems – and the machine components – the supporting Command and Control Information Technology Infrastructure. Existing solutions can contribute efficiently to their application domain and – at the same time – they are aligned with the constraints of other contributions, as all solutions first contribute to the common model (making sure that the information of interest is part of the model by extending and enhancing it following engineering methods) and the resulting mapping allows to find out immediately which other solutions use the same information (as same pieces of information is mapped to the same elements of the common model) and which constraints need to be fulfilled (as the context of each information element is captured in form of metadata).

Although the framework addresses all aspects of the challenge of coming up with a common understanding of the operations and a supporting language usable for Command and Control by humans and machines, the paper will focus mainly on IT system related issues. The paper will present a general framework resulting from studies conducted in support of Community-of-Interest (COI) Model-based Languages enabling Composable Net-centric Services for NATO's Research & Technology Organization, the US Joint Forces Command, the US Army Test & Evaluation Command, and the US Army Topographic Engineering Center. The paper will discuss the general challenges that have to be addressed by the framework and introduce the framework and sample examples. In addition, the authors will present real world cases in which the framework was applied before summarizing conclusions and recommendations.

2 A Framework for COI Model-based Languages

The initial question that needs to be answered is how to use models to support effective and efficient communication within a COI. The approach recommended in this paper is motivated by the research conducted by the authors in support of US Joint Forces Command, the US Army, US Homeland Security, and NATO starts with a common understanding of data exchanged in form of a common reference model (CRM). The CRM is then used to build sentences and ultimately a web-based language to communicate these data in applicable contexts of the COI.

The idea of using COI approaches in support of interoperable IT solutions is not new. In (Renner, 2001), a COI is defined as *the collection of people that are concerned with the exchange of information in some subject area*. The community is made up of the users/operators that actually participate in the information exchange; the system builders that develop computer systems for these users; and the functional proponents that define requirements and acquire systems on behalf of the users. Renner stresses the importance of COI data panels and their task to support Common Data Representations (CDR) to be used within the COI for data exchange.

2.1 Defining the Problem Space

In order to set up the framework, we must first distinguish four categories of communication, as captured in the following figure: humans talking to humans, humans getting information into supporting IT systems, potentially heterogeneous IT systems exchanging information, and IT systems producing reports for humans.

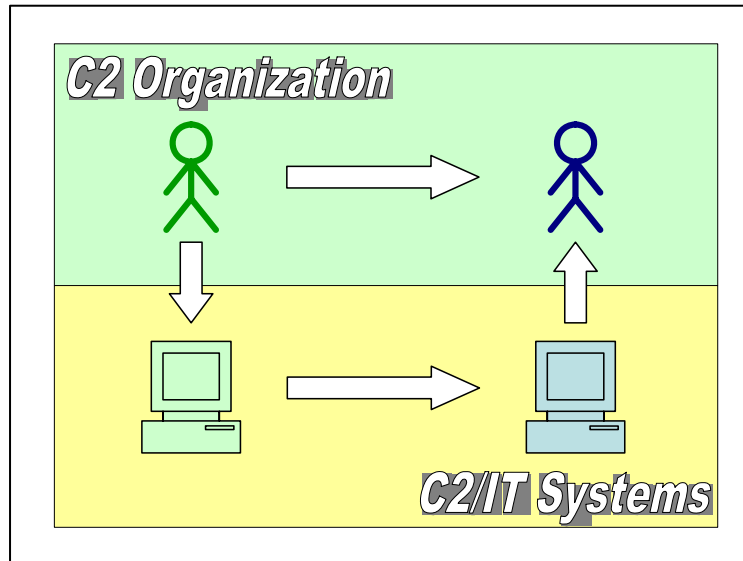


Figure 1: Categories of C2 Communications

- **Human-to-Human Communication:** This category includes organization-to-organization communication as well as human-to-human communication. It has been pointed out in several CCRP publications that organizational changes in support of net-centricity of organizations is often more important than IT system updates (see among others Alberts & Hayes, 2003). Many challenges of C2 are situated in this domain. Difficulties range from overcoming cultural and linguistic challenges to the quest for a common situational awareness. Although a very important aspect – if not the most important one, as the other aspects are merely supportive of this aspect – the focus of this paper is on the other three components. However, as the expertise of the authors of this paper lies in the IT section, we will not focus on these issues in more detail besides recognizing their critical importance.
- **Human-to-Machine Communications:** The Human-Machine-Interface is an important component for every IT application. However, when it comes to common communication concepts as captured in net-centric languages, a new aspects needs to be considered. While the spoken language of humans – the natural languages – is ambiguous, often ill-structured, and normally requires a rich context for interpretation, the language of machines is governed by the rules of logic. This category translates the human-oriented view into the machine-oriented view. This task may be the most challenging in the framework. The discipline of computational linguistics deals with related

challenges. Until recently, the Human-machine interface was the major element of concern for this component. The programming of user-friendly graphical interfaces with context sensitive pull-down menus to guide the human in constructing expressions valid for the machine was the state of the art. The result was a regular expression understandable and valid for the machine. Messages can be interpreted as special regular expressions. The first prototypes in support of the BML efforts – as described by Sudnikovich et al. (2004) – followed this trend by providing such an interface. More recent research applies computational linguistics to enable a better communication based on information exchange, such as summarized, among others, by Hecking (2004b). Computational linguistics started as the scientific study of language from a computational perspective. As such, it is an interdisciplinary approach combining linguistic theory – including phonology, syntax, semantics, pragmatics – with computer science – including artificial intelligence, theory of computation, programming methods. It applies statistical as well as rule-based models. The traditional focus is the modeling of natural language from the computational perspective. As recent research has shown that human language is much more complex than previously thought, computational linguists often work as members of interdisciplinary teams, including linguists, project specific application experts, and computer scientists. The resulting tools enable to process natural language and transform this information into regular expressions understandable by machines. Hecking (2004a) showed that this process can be facilitated significantly by formalizing the expressions used, which is in particular of interest in the military domain, as warfighters have been trained for generations to use clear and unambiguous expression when giving orders or reports. The work of Schade and Hieb (2007) in support of C-BML contributed significantly to this component as well. The premise is that the use of a structured language to convey orders including the commander's intent as well as reports must be captured in a formal grammar. The grammatical approaches predominantly taught in computer science, the so-called Chomsky grammars, are described as insufficient for C-BML purposes. Instead, the use of lexical functional grammars (LFG) is proposed. From an Internet survey conducted by the authors, the main applications of LFG are translation challenges. The LFG Bibliography maintained by the Universities of Essex, Stanford, and Koblenz¹ gives examples of how LFG is used to parse natural languages into formal structures for further computer manipulations.

- **Machine-to-Machine Communication:** A lot of research has been done regarding machine-to-machine interoperability. Federated databases, common data replication mechanisms and common message formats are all examples of IT-system centric communication. The domain of expertise of the authors of this paper falls into this category. It is rooted in the theory of federated databases (Sheth & Larson , 1990). The objective of federated data bases is to merge different data sources, which are – and remain – distributed,

¹ <http://www.essex.ac.uk/linguistics/LFG/www-lfg.stanford.edu/bibliography.html>

heterogeneous, and autonomous but appear as one coherent federated data schema to the outside application. The definition of the CRM – and capturing the data in the recommended CDR – is closely related to the definition of a federated schema, as already pointed out by Krusche and Tolk (2000). In addition, machines need to follow the principles of logic. Without logic, proofs are not possible. Consequently, construction techniques must be applied to derive the different views from one common model. As shown by Hein (2002), these techniques can be divided mathematically into defined sets and grammars. Defined sets are used for data modeling, XML modeling, and other inductive techniques. Grammars introduce the production rules. It has been shown that for regular expressions the construction techniques can be mapped to each other, as both use finite automata as the underlying logical construct. That means that for every data model, an XML schema and/or a regular grammar can be constructed representing the same finite automata. It doesn't matter if one starts with the data model, the grammar, or the XML schema: if the transformations are consistent, always results in the same finite automata which determine scope, resolution, and structure of the set of valid regular expressions. This leads us to the result that machine-machine communications can and should be governed by regular expressions. The authors of this paper are convinced that in order to reach machine-to-machine interoperability the use of regular expressions is necessary, although it may not be sufficient. The applied solution cannot leave room for interpretation, as machines cannot support such interpretations, if they are not programmed to support this. If they are programmed, the programs will transformation the data model into yet another data model, which either represents the same or a different finite automata. In both cases, however, the result is a set of regular expressions, so that the argument holds. In summary, the theory of federated databases is necessary – and potentially sufficient, but there is no proof for this assumption, only experience by application in various domains – to handle the challenges of machine-to-machine communications.

- **Machine-to-Human Communication:** It is important to understand that the translation from the machine-oriented view into the human-oriented view has some challenges as well, but that it is easier to solve than category two. The machine-human-component has the task to map from the logical world of regular expressions to the natural language of humans, which is much easier than the other way round. Many publications deal with the graphical representation of data to support an easier interpretation, which in the military domain leads to improvements in situational awareness. Another aspect of this component is the ability to generate written – or even spoken – reports in human-acceptable form from regular expressions. As mentioned earlier, regular expressions can be transformed into a relational data model, so the example here will utilize data models without limiting the applicability of the underlying ideas for other forms of regular expressions. In relational data models, data are captured in tables that can be related with each other. Each table represents a concept with a common understanding in the COI that is

described by its attributes. Attributes can have predefined values. For each attribute, a sentence of the following form can be generated:

The <attribute> of the <concept> is <attribute-value>.

Examples are the “Readiness Status” of “2nd Infantry Battalion” is “80%” or the “Reporting Data Reliability Code” of “the Current Report” is “Completely Reliable.” What each term exactly means is captured in the definitions of terms in the COI CRM, such as this is the case for the Joint Consultation, Command and Control Information Exchange Data Model (JC3IEDM) within the MIP community.

Similarly, associations between tables are rooted in operational associations between the concepts described by the table. A typical example is the order of battle that builds a hierarchy using the military organizations. The general form is as follows:

<subject-name> <association-type> <object-name>

Using the order of battle example, a possible sentence is that the “2nd US/GE Corps” “has operational command of” the “2nd Infantry Battalion.”

Most current database administration systems allow the definition of user-specific reports based on these features. Some system architecture tools use similar ideas to generate system specification documents from the repository, which is filled by defining the artifacts of the supported system modeling and architecture method.

Using speech generators, this technique can be used to generate spoken reports as well. In particular for training purposes, this feature was already supported. The US Air Force uses them to simulate communications between pilots and air traffic controllers. The ability to automatically generate speech is invaluable in situations where it is required to represent a number of external agencies or operators within a simulated environment. Stoner et al. (2003) give additional examples. Kenny et al. (2007) even describe “virtual humans” that report to the training audience using related ideas.

However, as our focus in this paper is a COI-model driven language we will not deal with them in detail. The interested reader is pointed to Tufte’s work (1997, 2001) as a good starting point.

These four categories establish the problem space that a framework enabling the definition of COI model-based languages must solve.

2.2 Implications for COI Model-based Languages

In order to define a COI model-based language all four components captured in the problem definition must be taken into account. Even when focusing on a pure system-oriented view (machine-to-machine communication), the other three aspects need to be reflected; otherwise the result is an inconsistent and incomplete representation of the processes within the COI. However, it is also important to understand what challenges have to be solved in which component and how the different aspects of solutions work together, otherwise, suboptimal solutions are developed. The authors offer the following list of challenges and where they should be solved. This list is neither complete nor

exclusive. It is not only intended to present already conducted research but also to start additional research:

- **System-to-System Interoperability:** If the objective is the interoperability of two IT systems supporting a common operation, the assumption is that the IT systems as they are support the sub-tasks of an operation in their application domain. If the IT systems are not sufficient to support the sub-task, this is another challenge to be solved, but interoperability needs to identify which information elements (which are data elements defined by scope, structure, and resolution in a given context) can be mapped to each other. This challenge falls into the machine-to-machine component.
- **Sufficiency of an IT Solution for an Operation:** The question if an IT system is sufficient to support a task in an operation cannot be answered in the machine-to-machine component. To answer this question requires that the human/organizational component specifies the operational needs first. These needs, normally derived from doctrine and task lists by applying methods, such as the Military Mission and Means Framework (Sheehan et al., 2003), belong in the human-to-machine component of the framework. The result of such work is to make sure that all information elements needed are represented in the IT systems in the appropriate form.
- **Ambiguities in Expressions:** The task to reduce ambiguities in expressions is more important than ever if the recipient of such an expression is a machine, as machines don't make interpretations, they need unambiguous structures to read from. As Schade and Hieb (2007) formulate it: "*Without ambiguity means that there is an explicit structure that the Command Intent can be put into and then parsed out of. It also means that only one clear and definite outcome results from the parsing.*" The task of reducing ambiguity therefore falls into the human-to-machine component of the framework. Once the explicit structure is defined and populated, Model-based Data Engineering (MBDE) can be applied to mediate this structure into other representation forms.²
- **Defining a Common Language for System Interoperability:** This task is closely related to the question of the sufficiency of an IT solution for an operation. In order to define a language, the required information elements for possible operations need to be defined. What is needed is derived from the human/organizational component by operational requirement, but the result is an information exchange model that captures the mandatory information elements in machine adequate form, i.e., using construction techniques representing the necessary finite automata. An example how to apply grammars is given by Schade and Hieb (2007). However, in order to support an interoperable solution, the system-to-system interoperability aspect needs to be solved for the targeted systems in addition.

In summary, a common language must be defined by analyzing the operational needs (human/organizational component) resulting in the definition of information elements

² Model-based Data Engineering is based on the Data Engineering as introduced in the NATO Code of Best Practice for C2 Assessment (NATO, 2001). It is characterized by the use of a common reference model to facilitate the data management and alignment process, which is a representation of the COI data model.

(data defined by scope, resolution, and structure in a context) in construction techniques representing a finite automata. Systems that need to support an operation must be sufficient, i.e., their information elements must be mappable to the subset of the common language representing their tasks, which are also represented as information elements. MBDE methods can be used as soon as information elements are defined. To this end, the authors designed a top-down/bottom-up approach that will be presented in the following section.

3 The Top-Down/Bottom-Up Approach to derive a COI Model-driven Solution

In order to derive a common understanding between diverse organizations supported by heterogeneous IT systems, the following multi-layer, model-based approach was proposed, published, and discussed during several conferences (Tolk et al., 2007). It applies the ideas of the framework discussed in section 2.

3.1 Overview of the process

The approach combines top-down analysis with bottom-up systems alignment. The top-down approach concerns itself with deriving accurate conceptual models of the operational goals of the organizations, and then revealing the important data concepts that constitute those models. The bottom-up approach starts with the information models used in the supporting IT systems and results in a CRM describing the information exchange that needs to be supported between the systems regarding the data concepts identified by the top-down approach. The following figure exemplifies these ideas. Figure 2 shows an overview of the steps of this process.

There are three initial top-down steps followed by three bottom-up steps. The result is a COI CRM plus web services implementing a COI model-based web language allowing the systems to exchange information in support of the common conceptual model of the underlying processes.

1. In the first top-down step, the participating organizations agree on the common process to be conducted. This is a high-level business model in which the leaders of the organizations must agree to cooperate and to accomplish common goals.
2. Based on these decisions, a common conceptual model needs to be set up. The recommendation of the research team is to use standardized methods, such as the Unified Modeling Language. Examples how this can be done for business processes are given by Erikson and Penker (2000).
3. The result of step 2 is a model of the business processes to be conducted and the organizations that have to conduct these processes. In step 3, the interfaces between the processes are evaluated in regard to which information needs to be exchanged to orchestrate these processes: Who says what to Whom in which contexts based on which results?
4. The first step of the bottom-up process is to document which IT systems are needed in support of the processes. This includes the identification of programs, components, and services plus supported interfaces.

