2006

Ontological Implications of the Levels of Conceptual Interoperability Model

Andreas Tolk
Old Dominion University, atolk@odu.edu

Charles D. Turnitsa
Old Dominion University, cturnits@odu.edu

Saikou Y. Diallo
Old Dominion University, sdiallo@odu.edu

Repository Citation

Tolk, Andreas; Turnitsa, Charles D.; and Diallo, Saikou Y., "Ontological Implications of the Levels of Conceptual Interoperability Model" (2006). *Modeling, Simulation & Visualization Engineering Faculty Publications*. 33.
https://digitalcommons.odu.edu/msve_fac_pubs/33

Original Publication Citation

Tolk, A., Turnitsa, C. D., & Diallo, S. Y. (2006). *Ontological implications of the levels of conceptual interoperability model.* Paper presented at the World Multi-Conference on Systematics, Cybernetics and Informatics (WMSCI), Orlando, FL, July 16-19, 2006.

Ontological Implications of the Levels of Conceptual Interoperability Model

Andreas TOLK, Charles D. TURNITSA, Saikou Y. DIALLO
Virginia Modeling Analyses & Simulation Center, Old Dominion University
Norfolk, VA 23529, USA
+1.757.686.6200 (Phone), +1.757.6214 (Fax), [atolk,cturnits,sdiallo]@odu.edu

ABSTRACT

The Levels of Conceptual Interoperability Model (LCIM) was developed to cope with the different layers of interoperation of modeling & simulation applications. It introduced technical, syntactic, semantic, pragmatic, dynamic, and conceptual layers of interoperation and showed how they are related to the ideas of integratability, interoperability, and composability. This paper will be presented in the invited session “Ontology Driven Interoperability for Agile Applications using Information Systems: Requirements and Applications for Agent Mediated Decision Support” at WMSCI 2006.

Keywords: Integratability, Interoperability, Composability, Syntax, Semantics, Pragmatics, Dynamics

1. INTRODUCTION

Until recently, the support of decision makers often focused on representing data. However, the advent of intelligent software agents using the Internet introduced a new quality to decision support systems. While early systems were limited to simple situations, the examples given by Phillips-Wren and Jain [1] show that state-of-the-art decision support is based on agent-mediated environments. Today, real-time and uncertain decision problems can be supported to manage the decision making process in a highly dynamic and agile sphere. Simple data mining and presentation is no longer sufficient: based on historic data, trend analysis and possible development hypotheses must be developed and compared. This requires a purposeful abstraction of reality and the implementation of the resulting concept to make it executable on computers. These processes are better known as “modeling,” the purposeful abstraction of reality and capturing of assumptions and constraints, and “simulation,” the execution of a model on a computer. Modeling & simulation (M&S) becomes more and more a backbone of operational research to cope with highly complex and dynamic environments and decision challenges that are often ill- or semi-structured in nature.

While M&S systems are valuable contributors to the decision makers toolbox, the task to compose them in a meaningful way is everything but trivial. Currently, various organizations are coping with the task to develop a theory of composability. Petty and Weisel [2] formulated the current working definition: “*Composability is the capability to select and assemble simulation components in various combinations into simulation systems to satisfy specific user requirements. The defining characteristic of composability is the ability to combine and recombine components into different simulation systems for different purposes.*” In order to be able to apply engineering methods to contribute to a composable solution, several models have been developed and applied. However, at the end a machine readable and understandable implementation based on data and metadata is needed to enable agents to communicate about situations and the applicability of M&S applications. They must share a common universe of discourse in support of the decision maker, which requires a common language rooted in a formal specification of the concepts. A working definition of a common ontology is a formal specification of a conceptualization.

This paper shows how various layered composability approaches contributed to the definition of the Levels of Conceptual Interoperability Model (LCIM) and how the results can be used to derive implications and requirements for ontologies describing the universe of discourse in which intelligent agents serve to mediate between agile applications in order to compose the individual systems into a meaningful system of systems.

2. MOTIVATION FOR AGENT MEDIATED DECISION SUPPORT

This section deals with the rationale for working on agent-mediated support and how this is applicable in the broader context of complex business operations to be supported by agile systems. For the military application domain, Alberts and Hayes [3] define the quality of support by decision support systems in net-centric environments using the net-centric value chain, which distinguishes four categories. They are easily applicable in the broader context as well.

- The value chain starts with *Data* Quality describing the information within the underlying command and control system. This definition can be generalized to be applicable to decision support systems.
- *Information* Quality tracks the completeness, correctness, currency, consistency and precision of the data items and information statements available.
- *Knowledge* Quality deals with procedural knowledge and information embedded in the decision support system such as templates for behavior, assumptions about capabilities of entities, and domain specific assumptions, often coded as rules.
- Finally, *Awareness* Quality measures the degree of using the information and knowledge embedded within the decision support system. Awareness is explicitly placed in the cognitive domain.

Data representing decision support systems were only able to reach the data quality. By bringing the data of heterogeneous systems together into a common situation display adds the necessary context needed for information. However, in order to reach the next level of knowledge, procedural knowledge is needed. Finally, if data and metadata enables software agents to select different M&S components and compose them to evaluate alternative hypotheses, even the cognitive domain of awareness can be supported.

3. LEVELS OF CONCEPTUAL INTEROPERABILITY

As in the last section, the underlying work on composability of M&S applications is mainly based on military applications, in particular from the domain of using simulation systems for training and experimentation in support of armed forces. Nonetheless the results are easy to be generalized for other application domains, such as complex business scenarios, traffic flow [4], or medical emergencies [5].

Models for Composability

The composability discussion started with Harkrider and Lunceford [6] making the case that technical integration of systems is necessary but not sufficient. Based on similar observations, Dahmann [7] distinguished between technical interoperability and substantive interoperability. Petty [8] extended the technical interoperability layer and introduced hardware, communication, and protocol layer. However, while the community focused on implementation questions, it became obvious that many challenges are on higher levels: the underlying concepts and models that have to be aligned in the process of federating systems. While most current standardization efforts, such as IEEE 1278 [9] and IEEE 1516 [10], are focused on the implementation level, standardization must be aimed at the modeling level to ensure interoperability between systems. Page et al. [11] introduced the idea to differentiate between technical layers for integratability, implementation layers for interoperability, and modeling layers for composability. Therefore, the LCIM detailed the substantive interoperability level in order to cope with these challenges explicitly.

Overview of the LCIM

The research on composability conducted at the Virginia Modeling Analysis & Simulation Center resulted in the LCIM, which underwent several improvements since its first publication [12]. The current version of LCIM as depicted in Figure 1is documented in [13]. The different levels are characterized as follows:

- Level 0: Stand-alone systems have *No Interoperability*.

- Level 1: On the level of *Technical Interoperability*, a communication protocol exists for exchanging data between participating systems.[1] On this level, a communication infrastructure is established allowing it to exchange bits and bytes, the underlying networks and protocols are unambiguously defined.

- Level 2: The *Syntactic Interoperability* level introduces a common structure to exchange information, i.e., a common data format is applied. On this level, a common protocol to structure the data is used; the format of the information exchange is unambiguously defined.

- Level 3: If a common information exchange reference model is used, the level of *Semantic Interoperability* is reached. On this level, the meaning of the data is shared; the content of the information exchange requests are unambiguously defined.

- Level 4: *Pragmatic* Interoperability is reached when the interoperating systems are aware of the methods and procedures that each other are employing. In other words, the use of the data – or the context of its application – is understood by the participating systems; the context in which the information is exchanged is unambiguously defined.

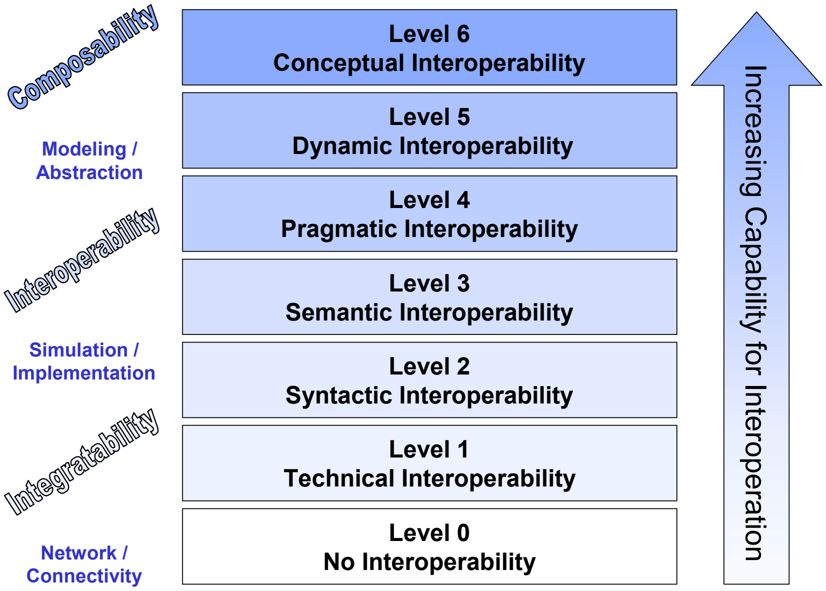

Figure 1: Levels of Conceptual Interoperability

- Level 5: As a system operates on data over time, the state of that system will change, and this includes the assumptions and constraints that affect its data interchange. If systems have attained *Dynamic Interoperability,* then they are able to comprehend the state changes that occur in the assumptions and constraints that each other is making over time, and are able to take

---

[1] Some early alternatives distinguish furthermore between hardware level and communication level when analyzing the domains of technical interoperability.

advantage of those changes.[2] In particular when interested in the *effects* of operations, this becomes increasingly important; the effect of the information exchange within the participating systems is unambiguously defined.

- Level 6: Finally, if the conceptual model – i.e. the assumptions and constraints of the meaningful abstraction of reality – are aligned, the highest level of interoperability is reached: *Conceptual Interoperability*. This requires that conceptual models will be documented based on engineering methods enabling their interpretation and evaluation by other engineers. In other words, on this we need a “fully specified but implementation independent model” as requested in Davis and Anderson [16] and not just a text describing the conceptual idea.

It should be pointed out that these layers of operations are still driven by implementations of agile systems that should be described in order to enable intelligent software agents to evaluate their applicability to support a decision and their composability with other solutions. As such, it is a typical bottom-up approach. The objective is to generate a usable and sufficient description based on data and metadata supporting the composition of applicable agile components and systems to support the decision maker; it is not to generate a general and complete description of the problem sphere. We are well aware of alternative top-down approaches that start with a common understanding to derive necessary implementations; however, the application domain we are focusing on in this paper uses already implemented agile systems to support a higher goal of the decision maker, so capturing the capabilities and constraints of available services, applications, and systems was the primary driver behind this effort. To what degree the bottom-up approach can be merged with top-down approaches, such as the coherence/correspondence approach described by Sousa-Poza [17] is topic of ongoing research.

4. ONTOLOGIES FOR COMPOSABILITY

Our working definition is that *“an ontology is a formal specification of a conceptualization.”* As mentioned at the end of the section on the LCIM, this definition is not aimed at the definition of an upper ontology describing everything within a possible universe of discourse, but to describe the information exchange requirements and means for orchestration and choreography of highly agile, independently developed systems into a supported framework mediated by intelligent agents.

Entities, Relations, and Rules

In order to access the conceptualization that an ontology is a formal specification of, it is necessary to break that specification up into accessible components. The first three types of components that are discussed are entities, relations and rules. Entities and relations are quite familiar to the data modeling community, and also appear within most modern ontological engineering theories. Rules, however, are an additional component that assists with the ontology model being useful to systems, and will be described here in more detail. A fourth component, concepts, is essential to the other component types and will be addressed in its own section, below.

As this paper is addressing ontology of information systems, and more specifically, ontology for the purpose of assisting interoperability between information systems, entities become quite easy to define. As they are revealed in [18], it can be seen that they are easy to recognize within a model. Entities are the exchangeable symbols (words, data elements, etc) that represent the *things* of which our systems can address. Things are further defined as being not only physical things, but also everything, which can be addressed by systems (things, both physical and otherwise; phenomena, including both processes and events; modifiers for both of these).

Entities, in order to satisfy the specification presented here, need to be represented as both types and instances. Entity-types may be divided up further into subtypes, but each child of an entity-type (whether a true instance, or a subtype) retains all of the identity of the parent type. This idea of terms of understanding being less generally defined than their parents is known in the knowledge representation and artificial intelligence communities as subsumption, and a treatment of the topic can be found in [19]. The organization of all of an ontology model’s entities into an interconnected graph is referred to as a taxonomical model, or an entity model.

Different entities, originating from different systems, may have the same “name”, or symbol, representing them and have different characteristics. This leads to a situation making the enablement of interoperability very difficult. Additionally, difficulties in enablement would arise when differently named entities are meant to represent the same thing from our limited universe of discourse. In both situations, and as hinted at above, it can be seen that entities differ from each other based on their characteristics. These characteristics are defined by the concepts that the entities can exhibit. This is discussed further, below.

The entity-model of type-subtype-instance relationship is not the only class of relations between entities that can exist. Relations can provide a semantic link between entities in any number of different ways. The universe of discourse under consideration might find it useful to, for example, have a semantic link showing the relation between part entities of a whole entity. In this case, the part is not a child of the whole in the type-subtype-instance sense, but there is certainly a semantic link. The enumeration of particular relation types is potentially unique for each universe of discourse [20].

Relations can exist between entity-types and other entity-types, between entity-types and entity-instances, or between entity-instances and other entity-instances (of the same or different entity-type). The applicability of the relation to particular entities is based on the defining characteristics of the entities in question. It should be noted that these defining characteristics also are associated by the common concepts, which are exhibited by the entities in the context of the application.

System-to-system interoperability requires exchange of data, and that data (in order to move past what the LCIM refers to as Level 1) must have a syntactic form. Further, to proceed to even higher levels of conceptual interoperability, semantics are

---

[2] Methods that enable such interoperability can be (documented) open source, reference implementations, or adequate documentation, such as complete UML models or DEVS models [14]. Tolk and Muguira [15] proposed an initial framework based on the LCIM merging several engineering approaches, including UML and DEVS, to insure consistent interoperation of services.

required of the data interchange. In both cases, and for further extension, a rule set, or grammar, is required to control the syntax and semantics of the data exchanged. But the data within a system undergoes certain operations defined by that system. A set of rules defining the syntax and semantics of those operations is also required.

For the purposes of explaining ontology to assist with system-to-system interoperability rules can be viewed in two ways. Rules exist as both *internal* rules and *external* rules. Internal rules are the rules, similar to a grammar, that determine within a formal ontology, which entities can, and do, operate upon each other by the internal workings of a system exhibiting that formal ontology. The internal rules are the functional for the entities to interact with each other within the ontology under consideration. Functional rules often represent business rules. External rules, on the other hand, are not captured within the ontology, but must be supported by the entities (and their characteristics). External rules, for the purpose of this definition, are the rules defining interaction between systems.

The existence of an entity model that systems can reference allows for the specific identification of entities referred to during system-to-system communications [21]. A set of rules (internal and external) can provide for a semantically meaningful method for combining those entities into communications that satisfy the system-to-system communications supporting interoperability up to the semantic level. Internal relations identified among the entities of a system's data model even allow, in effect, inference to be made within the interoperability supporting data exchanges between systems[3]. What is still missing from our ontology, although it was mentioned several times above, is the specific characterization of our entities. This characterization provides for definition of our entities, and also allows for the application of the relations and rules defined above. Concepts, which are exhibited by entities, provide this characterization.

Concepts: Atomic Elements of Understanding

Concepts are the basis for giving entities definition and characterization. They are the most difficult component of the ontology to define. They are also often difficult to see within the entities that exhibit them. It is helpful to have a good definition of what is meant by concept in order to see how the ontology model requires them. One aspect of concepts to consider during the definition of the term is that concepts are the only component of our ontology that exists within actual items. They are the link between a data representation of an item, and the actual item itself. The concepts behind, for instance, a truck, and the data representation (within an information system) of a truck are the same [22].

A concept is defined as "what is universal to all entities said to exhibit the concept" [23]. Concepts are not contained within entities, but they are exhibited by entities. Concepts are larger than entities, in the sense that they be exhibited by any number of entities within a formal ontology, yet a single entity may exhibit any number of concepts. An entity is required to exhibit a concept. As such, concepts reflect ideas as units of semantics in the mental or knowledge representation model of the universe of discourse, and it requires an entity to instantiate this idea. Removing of the exhibition of a concept from an entity will fundamentally change the entity (or it would not be a characteristic of the entity).

Just as entities exhibit concepts, concepts themselves exhibit properties. Those properties can exhibit property values. The properties that are exhibited by a concept cannot change, however the property-values can exist within a range. The limits for a property-value are as far as they can be changed, without changing the generally perceived meaning of the concept, which exhibits them. An example of how far a property-value can be extended, is seen by considering the concept "red". Certain entities exhibit the concept red, which has a property of "measurement of red in the 256x256x256 RGB palette". That property, of course, has a series of triples as property-values. The range of values of the triples can change a great deal, so long as the resulting color is identifiable as being within the spectrum of "red". If it strays out of that spectrum, the basic identity of the concept has changed.

As mentioned above, concepts can be difficult to be seen within entities. The properties and property-values exhibited by the concepts, however, should not be so obtuse. Such characteristics (including both properties and property-values) are the measurable indicators of something, which is just a product of mind, namely a concept. Some experts come to similar conclusions without the use of concepts, such as [24], however, the existence of characteristics is generally accepted. An entity generally has certain properties or property-values that can be identified. By the definition given above, these cannot be exhibited by the entity, but only by concepts, which are exhibited by that entity. Therefore, once properties, or property-values, are identified as being exhibited by an entity, then the parent concepts of those properties can be identified.

If the concepts, which give identity to an entity, are known, and captured within the ontology, then regardless of any ambiguities with the entity's name (or symbol), it can still be clearly identified by using exactly these concepts [22]. Similarly, proper definition of the concepts that give definition to the entities of two different systems interoperating with each other can show where there may be conceptual gaps or misalignment between those entities.

Apparent Ontologies defined by Interface Specifications

By looking at the agreed to interface specification (which have been identified as a source for external rules, for the purposes of the ontology definition), we can help to understand the apparent ontology of a system supporting the interface. The process of revealing this apparent ontology, in the same language (using the same component structure) as other systems interoperating with can help to identify gaps (to be filled, if possible) in conceptual support of entities exchanged, and can also assist with the assessment as to the strength of the overall system-of-systems is concerned.

A definition of apparent ontology may be helpful before proceeding. Many of the existing systems, and systems yet to be developed, will have been constructed without a formal ontology being recorded. This does not mean that the system architects did not have an ontological view of the system's universe of discourse in mind when the design was taking place. Rather, this ontology is inherent in the data model of the

---

[3] Internal relations, as defined here, support inference in this way – if a semantic exchange of data is made referring to the entities of a system, and those entities have internal relations semantically linking to other entities, then the chain of related entities is affected, via inference of the semantic links, by the semantic exchange.

system, in the assumptions concerning the structure and meaning of that model, and in the operational functions and transformations that the system makes on that data. By examining the data elements of the system, this apparent ontology can be revealed, and described in an accessible artifact, so that it can assist with system-to-system interoperability.

To reveal this apparent ontology, it is helpful to begin with the interface specification. As mentioned, this suffices as the external rules for the ontology of the system, as it provides an effective grammar for the system to communicate.

From the interface specification, we can enumerate and codify the types and possible instances of entities coming from within the system. Any semantic relations between these entities will now suggest themselves, including any hierarchical structure (leading to an entity-model).

The entities of the system and their functional transformation that take place within the system exhibit the properties and property values. These characteristic properties allow for the identification of the underlying concepts. Once this is accomplished, we have a partial view of the apparent ontology of the system.

Working with the revealed apparent ontology allows us to compare, at the concept level, the sufficiency of meaning and depth of understanding of the exchanged entities. The enumeration of rules and relations reveals the inferred meanings of those entities, and the operation up on those entities within the system, thus revealing what may be needed in support from a foreign system to fully support interoperability to the semantic level, and perhaps to move beyond.

The existence of the revealed apparent ontology is itself useful for future developments of interfaces and evaluation of the soundness of combining the system with others. There is also value, however, in the process of revealing the apparent ontology, as it assists with evaluating the internal rules, the relations, and the entities of the system being investigated.

5. APPLICATION EXAMPLE

Our application example is rooted in the idea to generate a common language between operational entities, simulated entities, and robots operating in the same application domain to generate orders and plans from a planning organizations to the executing entities as well as to generate reports contributing to the awareness of the current developments from these entities to the planning organization. The underlying application is the international Coalition Battle Management Language (C-BML) effort discussed by Sudnikovich et al. [25]. Tolk et al. [26] describe the technique used to implement the ideas.

The different levels of interoperability are supported by the application of complementary standards and processes.

- C-BML uses the service-oriented architectures executed on the Internet – or the military counterpart called the Global Information Grid (GIG) – to exchange information elements. TCP/IP ensures that the elements can communicate with each other on the technical level.

- C-BML targets operational command and control systems, military simulation systems, and robotics. All these domains have domain-specific solutions, such as IEEE 1278 [9] and IEEE 1516 [10] for distributed simulation systems, but there are not many common standards. However, all systems can support web services, so XML becomes a common basis for structuring the data, hence we support the syntactical level.

- C-BML identified a common information exchange reference data model with broad acceptance. This common reference model comprises all concepts identified to share tasks and reports, hence we support the semantic level. Tolk and Diallo [27] show how these ideas can be generally used to not exclusively support military operations but other domains as well, such as complex business scenarios, traffic flow, medical emergencies, and other elements of critical importance for decision makers.

- In the implementation depicted in Figure 2, we used open sources and open standards to construct a web-based ontology-driven service-oriented architecture for information exchange and storage. In order to achieve

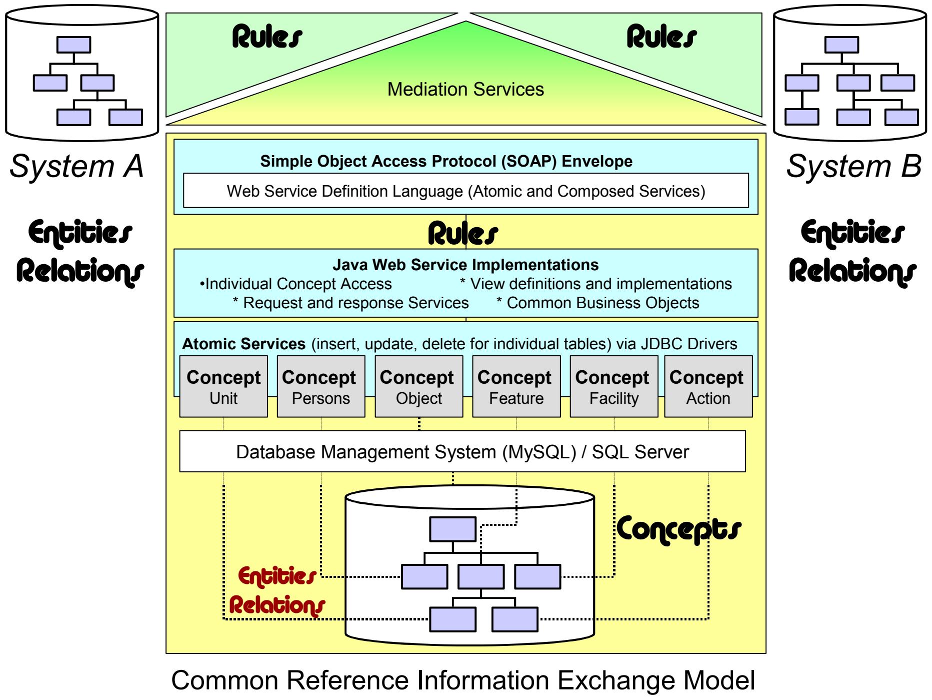

Figure 2: Ontology Driven Service-Oriented Architecture

> pragmatic interoperability, the concepts captured in the common information exchange reference data model were accessible via atomic web services. Following the rules, these concepts are combined into entities and relations of the apparent ontologies of the participating systems, resulting in composed web services which incorporate the business rules and objects of the targeted systems.

The ontological constructs *entities* and *relations* are used to describe the information exchange requirements of the participating systems, in the figure referred to as systems A and B, based on the implicitly defined apparent ontologies. How they are populated or how they disseminate information is captured in the construct *rules*. The common elements with a common interpretation in the universe of discourse and supporting the decisions are modeled as *concepts*. All these concepts can be accessed individually, so that all every possible composition can be generated based on the rules. In addition, commonly accepted business object comprising of more than one concept can be defined as well.

In practice, this effort has some limitations if using a common information exchange data model that is already established for operational use to exchange data between real system, as such a model usually already comes with in intended business logic to support. In other words, we already have a couple of business objects that comprise more than concepts. The developer is faced with mandatory fields that may be only of tangential interest for his application.[4] In a perfect world, such business objects are exclusively defined via rules. In practice, established information exchange data models can still be applied to model the necessary concepts as long as it is possible to insert, update, and access concepts individually via atomic web services.

The next step of our research will focus on the remaining two levels of interoperability: dynamic and conceptual. Currently, we are evaluating the use of UML and capturing the information using XML Metadata Interchange (XMI) to generate the necessary metadata. In particular when embedded into the higher constructs of OMG's Model Driven Architecture (MDA). However, the current state of our prototype only implements the levels up to pragmatic. Also, the use of intelligent software agents is under investigation and not yet a broadly accepted idea, but it works in related domains, in particular in the domain of semantic web applications such as described in Pohl [28], which is at least encouraging for the application domains dealt with in this paper.

6. SUMMARY

Our research showed that meaningful interoperability requires much more than technical layers of interoperability. The LCIM identifies the technical, syntactical, semantic, pragmatic, dynamic, and conceptual layers of interoperation. Ontologies have been shown to be a potential contributor on the semantic and the pragmatic level. To what degree they can support the dynamic and conceptual layer, however, is topic of ongoing research. In connection with web services, first implementations showed the potential.

We assume that the research we are contributing to with this paper will enable discussions on the objective beyond the Semantic Web, as envisioned in [29]: Our view is that we are moving towards a "Dynamic Web," supporting the orchestration and alignment of agile components at least up to the dynamic layer with standardized metadata and clearly going beyond the currently discussed concept of choreography based on business process languages [30]. These developments will enable us to support not only higher levels of interoperability, but also to contribute significantly to knowledge and awareness quality within agent mediated decision support system, as envisioned in [1].

7. REFERENCES

[1] G.E. Phillips-Wren and L.C. Jain (Eds.), **Intelligent Decision Support Systems in Agent-Mediated Environments**, Volume 115 Frontiers in Artificial Intelligence and Applications, IOS Press, 2005

[2] M.D. Petty, E.W. Weisel, *"A Composability Lexicon,"* **Proceedings IEEE Spring Simulation Interoperability Workshop**, IEEE CS Press, 2003

[3] D.S. Alberts, R.E. Hayes, **Power to the Edge, Command and Control in the Information Age**, CCRP press, 2003

[4] J.A. Sokolowski, R. Mielke, "Supporting Evacuation Planning with Modeling and Simulation," **Proceedings IEEE Spring Simulation Interoperability Workshop**, IEEE CS Press, 2006

[5] F. Agrò, I. Giuliano, F.A. Montecchia, "A new human-dynamic simulator (Air Man) for airway training in emergency situations," **Am J Emerg Med.** 2002 Sep; 20(5):495.

[6] S.M. Harkrider, W.H. Lunceford, "Modeling and Simulation Composability," **Proceedings Interservice/Industry Training, Simulation and Education Conference** (I/ITSEC), 1999

[7] J.S. Dahmann, F. Kuhl, and R. Weatherly, "Standards for Simulation: As Simple as Possible But Not Simpler: The High Level Architecture for Simulation," **Simulation, Journal**, Vol. 71, No. 6, 1998

[8] M.D. Petty, "Interoperability and Composability," M&S Curriculum of Old Dominion University: Short Course Presentation, **Old Dominion University**, 2002

[9] IEEE Standard Group 1278: **Distributed Interactive Simulation** (Revision 2002), IEEE CS Press

[10] IEEE Standard Group 1516: **High Level Architecture** (Revision 2000), IEEE CS Press

[11] E.H. Page, R. Briggs, J.A. Tufarolo, "Toward a Family of Maturity Models for the Simulation Interconnection

---

[4] In military command and control systems, the timestamp and origin of a report is of essential interest in order to be able to evaluate how to use the message when contributing to the situational awareness, therefore such fields are mandatory for the command and control domain. M&S applications have another focus for information exchange, so that they often not even support such fields.

Problem," **Proceedings IEEE Spring Simulation Interoperability Workshop**, IEEE CS Press, 2004

[12] A. Tolk, J.A. Muguira, "The Levels of Conceptual Interoperability Model (LCIM)," **Proceedings IEEE Fall Simulation Interoperability Workshop**, IEEE CS Press, 2003

[13] C.D. Turnitsa, "Extending the Levels of Conceptual Interoperability Model," **Proceedings IEEE Summer Computer Simulation Conference**, IEEE CS Press, 2005

[14] B.P. Zeigler, H. Praehofer, T.G. Kim, **Theory of Modeling and Simulation**. 2nd Edition, Academic Press, 2000

[15] A. Tolk, J.A. Muguira, "M&S within the Model Driven Architecture," **Proceedings Interservice/Industry Training, Simulation and Education Conference** (I/ITSEC), 2004

[16] P.K. Davis, R.H. Anderson, **Improving the Composability of Department of Defense Models and Simulations**. RAND, National Defense Research Institute Report, 2003

[17] A.A. Sousa-Poza, "Pragmatic Idealism as the Basis for Understanding," **Proceedings International Conference on Systems, Man and Cybernetics**, IEEE Press, 2005

[18] P.P. Chen, "The Entity-Relational Model – Toward a Unified View of Data," **ACM Transactions on Database Systems**, 1,1, pp. 9--36, March, 1976.

[19] R. Brachman and J. Schmolze, "An Overview of the KL-ONE Knowledge Representation System," **Cognitive Science,** 9(2): 171–216, 1985

[20] B. Smith, W. Ceusters, B. Klagges, J. Kohler, A. Kumar, J. Lomax, C. Mungall, F. Neuhaus, A. Rector, C. Rosse, "Relations in biomedical ontologies," **Genome Biology**, April 2005

[21] G. Zhang and B.P. Zeigler, "The system entity structure: Knowledge representation for simulation modeling and design," in **Artificial Intelligence, Simulation and Modeling**, L. E. Widman, K. A. Loparo, and N. R. Nielsen, Eds. New York: Wiley, 1989, pp. 47-73.

[22] J. Sowa, "Ontology, Metadata, and Semiotics", Published in B. Ganter & G. W. Mineau, eds., **Conceptual Structures: Logical, Linguistic, and Computational Issues**, Lecture Notes in AI #1867, Springer-Verlag, Berlin, 2000, pp. 55-81

[23] C. Turnitsa, "A Component Based Formal Ontology Model, A Method for Evaluating Such a Model, And the Results of an Application of that Method", Master's Thesis, Old Dominion University, 2006

[24] B. Smith, "Beyond concepts: ontology as reality representation," **Proceedings of FOIS 2004. International Conference on Formal Ontology and Information Systems**, Turin, 4-6 November 2004

[25] W.P. Sudnikovich, J.M. Pullen, M.S. Kleiner, S.A. CareyExtensible Battle Management Language as a Transformation Enabler. **Simulation Journal**, Vol. 80, pp. 669-680, 2004

[26] A. Tolk, S.Y. Diallo, K. Dupigny, B. Sun, C.D. Turnitsa, "A Layered Web Services Architecture to Adapt Legacy Systems to the Command & Control Information Exchange Data Model (C2IEDM)," **Proceedings ACM European Simulation Interoperability Workshop**, ACM Press, 2005

[27] A. Tolk, S.Y. Diallo, "Model-Based Data Engineering for Web Services," **IEEE Internet Computing**, Vol. 9, Nr. 4, 2005

[28] J. Pohl, "The Evolution of Intelligent Computer Software and the Semantic Web," **Proceedings 16th International Conference on Systems Research, Informatics and Cybernetics**; Intelligent Software Systems for the New Infostructure, pp. 11-34, 2004

[29] M. Daconta, L. Obrst K. Smith, **The Semantic Web: A Guide to the Future of XML, Web Services, and Knowledge Management**, Wiley, Indianapolis, IN., 2003

[30] A. Tolk, "What comes after the Semantic Web – PADS Implications for the Dynamic Web," **Proceedings 20th ACM/IEEE/SCS* Workshop on Principles of Advanced and Distributed Simulation**, IEEE/ACM Press, 2006