Applying the Levels of Conceptual Interoperability Model in Support of Integratability, Interoperability, and Composability for System-of-Systems Engineering

Andreas TOLK, Saikou Y. DIALLO, Charles D. TURNITSA
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. The model was successfully applied in various domains of systems, cybernetics, and informatics.

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, in particular when such M&S systems utilize knowledge management and agent directed simulation to enable intelligent decision technologies, such as agent mediated decision support.

While such enriched M&S systems are valuable contributors to the decision makers toolbox, the task to compose them in a meaningful way is everything but trivial. The challenge is not to exchange data between the system: the technical side is sufficiently dealt with by interoperability standards. The problem is that the concepts of the underlying models – or the implemented world view captured in the model – need to be aligned as well. 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 formal specification of a conceptualization, however, is a working definition of a common ontology. This ontology can then be applied to derive conceptually aligned and orchestrated configurations for conceptually composable, technically interoperable, and integrated solutions.

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. Cybernetics shows that simple methods often are to limited to be applied in complex environments like the system-of-system integration as envisioned here. The described method is therefore phased and combines bottom-up and top-down approaches: The information exchange requirements are identified by top-down analysis of the business processes to be supported and the informatics be applied. This is followed by a bottom up approach leading to a common ontology representing the various aspects of participating systems in phase two. Finally, the composition and integration of systems is orchestrated using top-down means in phase III.

The rest of this paper is organized following these ideas. After a short motivation why we need agent mediated decision support and how the work presented here fits into this vision, section 3 will introduce the LCIM. The three phases are described in sections 4 to 6. Section 7 gives an application example before we will give a summary.

2. MOTIVATION FOR AGENT MEDIATED DECISION SUPPORT

Before going into the details of LCIM and the three-phased method, 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. Instead of endless lists of data and messages, a common operational picture becomes possible. However, this picture is still only a snapshot. In order to reach the next level of knowledge, procedural knowledge is needed. This procedural knowledge can be provided in form of simulation services, as simulations are based on models, which are purposeful abstractions of reality, and simulations are the means to execute a model over time. We therefore move from the common operational picture to the common operation model. While a picture says more than 1,000 words (or the 1,000 pieces of enumerated data), an executable M&S application says more than 1,000 pictures!

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. However, in order to enable agents to become the ambassadors for M&S components (or other agile and dynamic processes and services), the agent must be aware of the assumptions and constraints underlying the model. This task is everything but trivial, as shown in [3a, 3b] and other publications. However, in order to support the cognitive domain of awareness, this knowledge must be captured in meta data interpretable by intelligent software agents. Yilmaz [6] evaluates these ideas of agent-mediated composition further.

3. LEVELS OF CONCEPTUAL INTEROPERABILITY

The underlying work on composability of M&S applications conducted by the authors is mainly based on military applications, in particular within 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 [7], or medical emergencies [8].

Models for Composability

The composability discussion started with Harkrider and Lunceford [9] making the case that technical integration of systems is necessary but not sufficient. Based on similar observations, Dahmann [10] distinguished between technical interoperability and substantive interoperability. Petty [11] 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 [12] and IEEE 1516 [13], are focused on the implementation level, standardization must be aimed at the modeling level to ensure interoperability between systems. Page et al. [14] 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 [15]. The current version of LCIM as depicted in Figure 1 is documented in [16].

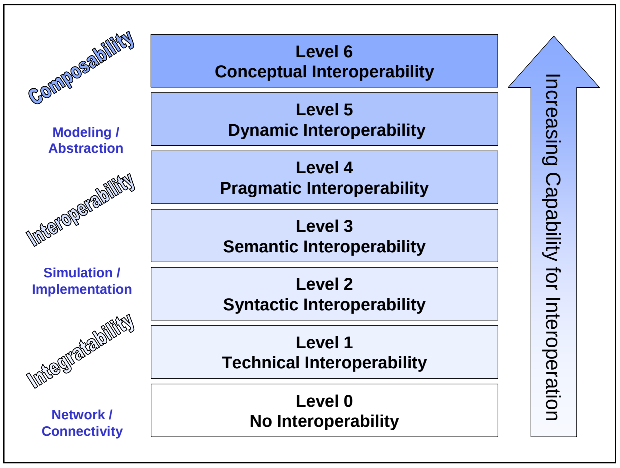

Figure 1: Levels of Conceptual Interoperability Model

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.

- 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 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 [19] 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 [20] is topic of ongoing research.

The LCIM was applied in various domains successfully and featured as a reference model in various journal contributions and book chapters. The originally intended use is described in [21]: applying the ideas to support composable M&S service for net-centric command and control applications. The Interoperability Framework for future U.S. Department of Energy solutions for the Power-Grid described in [22] uses a derivate of the model. How to apply the LCIM to align smart applications is the topic of [23]. Finally, the recent book on model and simulation-based data engineering uses the LCIM to show functionality and supported concepts of their solution [24]. The study of Carnegie Mellon University on System of Systems mentions the LCIM as one of the candidates for successful evaluation of approaches [25].

4. THE MODEL-BASED APPROACH TO SYSTEM OF SYSTEMS ENGINEERING

The following advantages of model driven approaches to reach alignment of heterogeneous decision-making processes are well known in the domain of cybernetics and informatics. They have been published in [26]. The idea is to use the first phase to understand the business processes and the supporting information technology (IT) solutions and capture them in a common model. The use of models is important, because:

- *Models help the decision makers understand the key mechanisms of an existing process.* A model provides a clear picture of acting entities, roles, relations, and tasks. This is needed to understand the processes of the allies as well as the processes of the non-military partners and vice versa.

- *Models act as the basis for creating suitable information systems that support the process*. The model comprises descriptions of process that can be used to identify necessary support. Furthermore, the sub-processes already supported by IT in the various participating organizations are displayed. This includes systems' interfaces as well as their information capability that is available information that can be delivered to other systems as well as suitable information that can be computed to deliver new insights. Therefore, the model puts the various existing systems into their place within the federated system of systems supporting the overarching operations and also serves as the requirement driver for additional IT support.

- *Models can be used to improve the current structure and operation*. By creating a common description of the overall operation, participating organizations and supporting systems, redundancies as well as bottlenecks become ob-

---

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

[2] Methods that enable such interoperability can be (documented) open source, reference implementations, or adequate documentation, such as complete UML models or DEVS models [17]. Tolk and Muguira [18] proposed an initial framework based on the LCIM merging several engineering approaches, including UML and DEVS, to insure consistent interoperation of services.

vious. Necessary changes can be identified and solutions can be derived and agreed on based on a common model.

- *Models show the structure of innovated solutions*. The model becomes the basis for a common action plan supporting radical as well as incremental changes. The desired end state and the necessary steps leading from the status quo to this end state are part of the model. The model itself becomes an important management instrument that orchestrates the necessary improvements in parallel and distributed events.

- *Models can serve as a basis to evaluate new ideas*. Models can be used to copy other structures, and evaluate processes used by other partners – or opponents – in the environment in which the operation takes place. As the model comprises the necessary detail needed to derive a conceptual or functional model of the mission space, support by M&S directly becomes possible. Respective experiments can help evaluate such future concepts. An appropriate model can be used to orchestrate respective efforts and helps create a common understanding of all participating institutions.

- *Models facilitate the identification of potential reuse of existing solutions*. Although every operation is special and unique, many processes are supported by standard solutions. Additionally, when using a common model, the identification of processes supported in other operations and that can be modified easily to support the current effort becomes feasible with minimal effort.

These arguments show that models play a pivotal role in gaining a common understanding of what processes have to be supported and which systems can help in these processes. As a product of this analysis, the supporting IT infrastructures, interfaces between systems and services, and the information that needs to be exchanged are identified.

Therefore, the first step of phase 1 entails identification of the organizations that will participate in satisfying a particular operation. This involves not only each organization, but also an understanding of what each organization is contributing to the operation, as well as what systems the organization has to support that contribution. The second step in this top-down approach is to construct a conceptual model of how each of the contributing organizations will make their contribution to the operation being discussed. Such a model will be based upon the doctrine of the contributing organizations. This model is based on the different modeling strengths described above, as it can result in not only a picture of what is expected to happen, but also provides a basis for showing how the different processes will interact with each other. This model is a conceptual model. Standard methods of systems engineering can be used in support of this task. The third and final step of phase 1 is the identification of information exchange events between the processes. While the second step resulted in conceptual models of all processes supported by each participating organization, we are now looking at the new overarching and common processes in support of the overall enterprise. In this process, the analysts identify the conceptual data domains and data element concepts needed to describe the information exchange necessary between the processes on the conceptual level.

The result of the top down approach is the conceptual understanding what information exchanges occur when, between which processes, what are the business objects into which the atomic information elements are composed or aggregated, and which organizations – and hence which supporting IT systems – contribute as source or target systems to this system-of-systems supporting enterprise wide applications.

5. ONTOLOGIES FOR COMPOSABILITY

The top-down use of models in understanding the alignment of different processes in phase 1 must be accompanied by an analysis of required information exchange between the processes resulting in data exchange requirements between the underlying IT systems and than used to compose the solutions in phase 2. The result of this activity is an ontology that is able to satisfy the information exchange requirements of the participating systems based on the concepts, relationships, and rules identified in phase 1.

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.

Enabling systems to interoperate based on a merging of each system's own ontological representation can be accomplished by a number of different methods [32]. The method suggested here is based on federating ontologies of different systems, which will allow for the exchange of meaning between the different world-views that the systems each have. This method is based, loosely, on the idea of federating databases, with the nature of ontological representation addressing some improvements to the method.

Federated Databases

Taken as the model for federating ontological representations, we take the approach of federated databases [33]. This approach is applicable when there is a requirement for an outside system to access a single data model that is representative of a merging of a number of distinct data models.

Within the world of databases, this idea of created a merged database, based on merged data models, while allowing the original components to remain distinct and intact has been accepted for some time: federated databases. The objective of such a data federation is to merge different data sources, which are – and remain – distributed, heterogeneous, and autonomous.

In order to meet this objective, Sheth and Larson introduce a new five level architecture [33], shown in Figure 2. Every application has its own data view, the external schema. They are based however, on a so-called "federated schema" being the common "data exchange" data model for all participants. Different from the conceptual schema of distributed homogeneous data bases, the federated schema only comprises the shared data elements, and doesn't deal with all details of the local autonomous data bases.

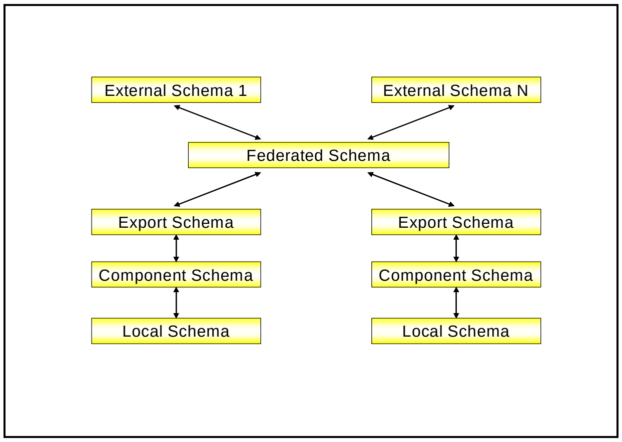

Figure 2: 5-Level-Architecture for Federated Databases

The local databases contribute to this federated schema their part via export schemata comprising the data to be shared by the local data base with other data bases. Each export schema is part of a local component schema, which is a common presentation of the data elements being comprised in the local, system dependent schema. Therefore, the five levels are external, federated, export, component, and local schemata.

The architecture shown in Figure 2 enables the evolutionary growing of the common data exchange model based on the actual information exchange request being formulated between the global applications and the local data bases. The moment, a new piece of data is needed in a global application; it becomes part of the federated schema. However, the local data bases don't have to be changed as long as that piece of data is already comprised in one of them.

In practice, the local schemata of the 5-level-architecture can be interpreted as the conceptual data model of the 3-level-architecture of the component model—the federate database. Additionally, export schema and component schema are often swapped. The reason is, that it seems not to be worth to translate all the tables of the local schema into the component schema, but only the parts of the data model that have to be used for the data to be exchanged during the federation execution.

Generally, two concepts have to be used to implement a federated database:

- Schema Transformation – The concept of schema transformation maps two data models onto each other in a semantically consistent way.
- Schema Integration – The concept of schema integration merges several different transformed schemata into a common resulting schema comprising data elements for every piece of information that is part of at least one of the original schemata.

As per Conrad [34], at least four rules have to be met to do the transformation and integration properly:

- Completeness – All concepts being comprised in one local schema must be comprised in the federated schema also.
- Correctness – All concepts in the federated schema must have either a semantically equivalent local concept or must be a new inter-schema-relation that may not be in contradiction to any local schema.
- Minimality – Logically and semantically equal concepts of the local schemata may be represented only once in the federated schema.
- Understandability – The federate schema must be logical and well documented (including semantics as well as sources, constraints, mappings, etc.) to facilitate the work of the users, as well as of the database administrators.

The problem of semantic heterogeneity was realized, in the mid-1990s as a major problem to federated databases, in that each data base is likely to have different semantic values for the objects it represents, and the relationships between those objects. An overview of this problem is described in [35]. While this problem may have existed for federated database instantiations, we will see below how an ontological representation method within a federated ontology may avoid this problem, by accomodating semantic heterogenous valuations of entities.

Federated Ontologies

Taking the approach recommended for layering federated databases, the ability to federate the worldviews of multiple distinctive ontological representations into a single ontological representation should be possible. The approach of federating ontologies is not a new one, and has been addressed several times in recent literature. One of the better known efforts was presented in [36], and proposes a layer of different ontological representations similar to the way a federated database system has a number of layered data schema.

The five layers, when applied to a federation of ontological representations, are as follows:

- Local Ontology – This is the ontological representation of each local system.
- Component Ontology – This is a transformed version of the Local Ontology, where each is represented using a similar representation method.
- Export Ontology – This is a subset of the Component Ontology, where only the subset of the Component Ontology that is relevant. Federated Ontology – This is a merge of all the Export Ontologies into single ontological representation that includes all aspects of the local ontologies.
- External Ontology – This is the portion of the Federated Ontology that might be of interest to an outside system that might have to interact with the system of systems.

As with federated database systems, each of these layers may need to have the principles of integration or transformation applied in order to derive it. In addition to integration and transformation, the method for arriving at both the Export Ontology and the External Ontology (which are possibly subsets of, respectively the Component Ontology and the Federated Ontology) requires a method of reducing the source ontology into some subset. This represents a third principle, that of subsetting.

For the discussion of a federated ontology, we will define these principles separately from how they were introduced for federated databases.

- Ontology Transformation – Transforming the ontological knowledge from one representation type into another. A current approach is presented in [37].
- Ontology Integration – This is commonly referred to as either merging or mapping of ontologies. A review and critique of many reported techniques is presented in [38].
- Ontology Sub-setting – Selecting only the ontological entities, attribution of those entities, and relationships, subject to requirements of the representation method employed, that are relevant to a sub-area of interest of the original ontology representation reduced from. Approaches, for selecting such a subset, are presented in [39] based on syntactic selection, and in [40] based on semantic selection.

The four rules of [34] – completeness, correctness, minimality, and understandability – should be equally applied to Ontology Transformation, Integration, and Sub-setting.

An effort in showing how federated ontologies may be constructed is presented in [36]. The approach described there is based on a series of source documents that may be relevant to several (two or more) ontologies. The principles of Formal Concept Analysis are applied to produce a merged structure of concepts from both ontologies, and then an algorithm (TITANIC) is applied to reduce that combined structure to a manageable new, merged ontology.

Ontological Entities

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 [27], 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 sub-sumption and a treatment of the topic can be found in [28]. The organization of all of an ontology model's entities into an interconnected graph is referred to as a taxonomical 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 primitives of meaning that the entities can exhibit. This is discussed further, below.

The type-subtype-instance relationship (of the taxonomical model) 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 enumeration of particular relation types is potentially unique for each universe of discourse [29].

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 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.

The existence of an taxonomical model that systems can reference allows for the specific identification of entities referred to during system-to-system communications [30]. A set of rules 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. Primitives of meaning, which are exhibited by entities, provide this characterization.

Primitives of Meaning: Atomic Elements of Understanding

Primitives of meaning, or just "primitives", 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

---

[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.

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 primitives to consider during the definition of the term is that primitives 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 [31]. These concepts are what we are calling primitives.

Each ontological entity has a unique collection of primitives of meaning. Within the domain that the systems in question come from, the primitives of meaning must be universally recognized and accepted. However, each system's ontological representation may have a different collection of primitives that make up the various entities it entails. This gives the different morphology of similarly named entities, and is where the defined difference may be found between the different system's world-views. As each system is a different abstraction of potentially the same reality, the difference is in which primitives of meaning each system assumes are involved in the make up of their ontological entities.

Following this reliance on primitives of meaning, if we have an ontological representation method that exposes these primitives, and this representation method is used for the Ontology Transformation between the Local Ontology and the Component Ontology, then the federated database problem of semantic heterogeneity should be solved. A trivial example which illustrates this point is given in Figure 3.

If the primitives, 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 [31]. Similarly, proper definition of the primitives 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.

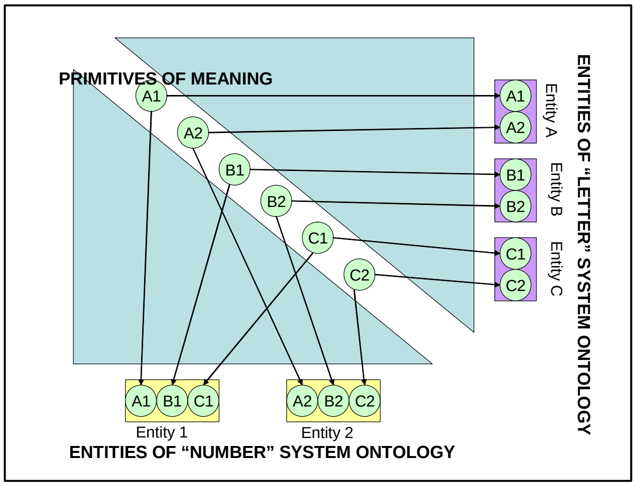

Figure 3: Primitives of Meaning

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 (1) data model of the system, in the (2) assumptions concerning the structure and meaning of that model, and in the (3) 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 primitives of meaning. 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.

The result of this phase is an ontological description of the information exchange model derived from the common conceptual model. This is more than just a data model. The ontological representation formally specifies the concepts regarding their property meanings (syntax and semantics), the contexts in which they are exchanged (pragmatics), and the business rules that need to be applied in form of axioms.

6. ONTOLGY-BASED SERVICE LANGUAGES

In principle, the ontology derived in phase 2 is sufficient to compose the contributing systems into a system of systems able to provide the IT infrastructure and services identified in the conceptual model. However, in order to support the engineers with more help, the third phase produces communication protocols and information exchange specifications applicable in the domain of service-oriented architectures.

In general, service-oriented architectures promise easier integration of functionality in the form of services into operational systems than is the case with interface-driven system-oriented approaches. However, although the Extensible Markup Language (XML) enables a new level of interoperability among heterogeneous systems, XML alone does not solve all interoperability problems users contend with when integrating services into operational systems. In addition, XML is often managed using underlying databases, which are less ambiguous than flat tag structures. But even when using data bases, the rules for accessing them appropriately need to be captured separately. Using an ontology as derived in phase 2 facilitates this process significantly: because all necessary information can be derived from one common model, the often observed inconsistencies between information exchange model and common reference model is avoided. The axioms of the ontology lead to business rules. The concepts, entities, relations, entities, and properties are mapped to table and attribute definitions, which are used to derive the XML schema.

The second advantage of this approach is that the information exchange requirements are based on the information exchange capabilities of the systems. Current practice is to define an information exchange model as a common language between the services. The model resulting at the end of phase 2, however, is based on the definition of exchangeable information identified in phase 1. In other words: the model is by design part of these systems: (a) what needs to be exchanged is part of this model, and (b) what is part of this model needs to be exchanged. A simulation system specific view of this approach has been published in [41].

7. 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 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. [42]. Tolk et al. [43] 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 [12] and IEEE 1516 [13] 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 [44] 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 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.

---

[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.

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 [45], which is at least encouraging for the application domains dealt with in this paper.

8. 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 [46]: 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 [47]. 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]. While this doesn't solve the challenge of system 0f systems engineering, as originally formulated in [48], the work contributes to potential solutions.

9. 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] M. Spiegel, P.F. Reynolds, D.C. Brogan, "A Case Study of Model Context for Simulation Composability and Reusability," **Proceedings of the IEEE Winter Simulation Conference**, IEEE CS Press, 2005

[5] R. King, "Towards Conceptual Linkage of Models and Simulations," **Proceedings IEEE Fall Simulation Interoperability Workshop**, IEEE CS Press, 2007

[6] L. Yilmaz, S. Paspuletti, "Toward a Meta-Level Framework for Agent-supported Interoperation of Defense Simulations," **Journal Defense Modeling and Simulation** 2 (3):161-175, 2005

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

[8] 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.

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

[10] 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

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

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

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

[14] E.H. Page, R. Briggs, J.A. Tufarolo, "Toward a Family of Maturity Models for the Simulation Interconnection Problem," **Proceedings IEEE Spring Simulation Interoperability Workshop**, IEEE CS Press, 2004

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

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

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

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

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

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

[21] A. Tolk, S.Y. Diallo, C. Turnitsa, L.S. Winters, "Composable M&S Web Services for Net-centric Applications," **Journal Defense Modeling and Simulation** 3 (1):27-44, 2006

[22] GridWise Architecture Council Interoperability Framework Team, "**Interoperability Context-Setting Framework**," V1.0, July 2007

[23] P. Dobrev, O. Kalaydjiev' G. Angelova, "From Conceptual Structures to Semantic Interoperability of Content," Bookchapter in **Conceptual Structures: Knowledge Architectures for Smart Applications**, pp. 192-205, Springer Verlag, 2007

[24] B.P. Zeigler, P.E. Hammonds, "**Modeling & Simulation-Based Data Engineering**: Introducing Pragmatics into Ontologies for Net-Centric Information Exchange," Academic Press, 2007

[25] E. Morris, L. Levine, C. Meyers, P. Place, D. Plakosh, "System of Systems Interoperability (SOSI),

Final Report", **Software Engineering Institute, Carnegie Mellon University**, Pittsburgh, PA, 2004

[26] A. Tolk, "Beyond Technical Interoperability - Introducing a Reference Model for Measures of Merit for Coalition Interoperability," **Proceedings of the Command and Control Research and Technology Symposium**, CCRP Press, 2004

[27] 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.

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

[29] 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

[30] 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.

[31] 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

[33] Sheth, A. P., and Larson, J. A. 1990. *Federated database systems for managing distributed, heterogeneous and autonomous databases*. ACM Computing Surveys 22: 183-236.

[34] Conrad, S. 1997. *Föderierte Datenbanksysteme*. Berlin: Springer.

[35] Colomb, R. M. 1997. *Impact of Semantic Heterogeneity on Federating Databases*. The Computer Journal 40(5):235-244.

[36] Stumme, G., and Maedche, A. 2001. *Ontology Merging for Federated Ontologies on the Semantic Web*. In Proceedings of the International Workshop for Foundations of Models for Information Integration. Viterbo, Italy.

[37] Mostowfi, F., and Fotouhi, F. 2006. *Improving Quality of Ontology: An Ontology Transformation Approach*. In Proceedings of the 22nd International Conference on Data Engineering Workshops. 61-61. Atlanta GA.

[38] Kalfoglou, Y., and Shorlemmer, M. 2003. *Ontology Mapping: The State of the Art*. The Knowledge Engineering Review. 18: 1-31

[39] Chopra, S., Parikh, R., and Wassermann, R. 2000. *Approximate belief revision*. In Proceedings of the Workshop on Language, Logic and Information, 2000.

[40] Budanitsky, A., and Hirst, G. 2001. *Semantic distance in wordnet: An experimental,application-oriented evaluation of five measures*. In Proceedings of the Workshop on WordNet and Other Lexical Resources, Pittsburgh, PA., 2001.

[32] Kalfoglou, Y., and Shorlemmer, M. 2003. *Ontology Mapping: The State of the Art*. The Knowledge Engineering Review. 18: 1-31

[41] A. Tolk, C.D. Turnitsa, "Conceptual Modeling of Information Exchange Requirements based on Ontological Means," **Proceedings of the IEEE Winter Simulation Conference**, IEEE CS Press, 2007

[42] 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

[43] 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

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

[45] 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

[46] 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

[47] 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

[48] C. Keating, R. Rogers, R. Unal, D. Dryer, A. Sousa-Poza, R. Safford, W. Peterson, G. Rabadi, "System of Systems Engineering," **Engineering Management Journal** Vol. 15 No. 3, pp. 36- September 2003