Proceedings of the 2007 Winter Simulation Conference
S. G. Henderson, B. Biller, M.-H. Hsieh, J. Shortle, J. D. Tew, and R. R. Barton, eds.

CONCEPTUAL MODELING OF INFORMATION EXCHANGE REQUIREMENTS BASED ON ONTOLOGICAL MEANS

Andreas Tolk

Engineering Management and Systems Engineering
Old Dominion University
Norfolk, VA, U.S.A.

Charles D. Turnitsa

Virginia Modeling Analysis & Simulation Center
Old Dominion University
Suffolk, VA, U.S.A.

ABSTRACT

Unambiguous definition of the information exchanged between distributed systems is a necessary requirement for simulation system interoperability. The ontological spectrum categorizes ontological means and shows the various degrees of support from recovery and discovery to reasoning. These means are highly applicable in support of data engineering to define information exchange requirements and therefore can increase interoperability. If the systems interfaces and resulting information exchange requirements are captured using the appropriate metadata, these ontological means can be furthermore applied to unambiguously identify exchangeable information. This allows systems to exchange information based on self-organizing principles using what they can exchange and not on mandated specifications of what they should exchange.

1 THE CURRENT PARADIGM

The current standardized solutions for simulation system interoperability follow the paradigm to identify a set of data elements in an agreed format with an agreed meaning to share information for initialization and execution. This is the case for Distributed Interactive Simulation (DIS) systems specified by IEEE1278, as they use well defined *Protocol Data Units* (PDU) that are standardized regarding syntax and semantics in the standard. Although the High Level Architecture (HLA) specified by IEEE1516 is more flexible, the Object Model Template (OMT) is used as the common syntax and the meaning of all object and interaction classes within the *Federation Object Model* (FOM) is defined federation specific in the Federation Development and Execution Process (FEDEP) and documented in the FOM Lexicon as well as in the federation agreements. In both cases, HLA and DIS, the information exchange model used is a model of its own resulting from consensus between the participating systems. Every system agrees to map their information exchange to these information exchange model. Some FOMs are even standardized, such as the Real-time Platform Reference FOM (RPR-FOM).

This paradigm is not unique to Modeling and Simulation (M&S). Every "black box" solution uses an interchange format that is well defined in syntax and semantics. The operational systems in the military domain exchange well defined messages using the *Message Text Format*. Alternatively, *Data Replication Mechanisms* are used to replicate pre-defined sections of databases using *Information Exchange Data Models*. Another alternative is the use of *Tactical Data Links*, such as the binary Link messages used by missile defense systems and aircrafts.

Overall, the importance of data has been recognized over the recent years. Data are perceived as a valuable resource, as the quality of operational supports is always determined by the quality of the underlying algorithms and their implementation as well as by the quality of obtainable data. Consequently, authoritative data sources and repositories are currently established in various organizations relying on such support, e.g., the Joint Training and Joint Experimentation Directorates of the Joint Forces Command. *"It is all about Data!"* became a slogan.

2 THE SPECIAL ROLE OF M&S

While the authors generally welcome the new intensified focus on data, the special role of M&S must be recognized in order not to fall short when necessary metadata structures in support of these processes are defined. The M&S community distinguishes between models and simulations. *Models* are a purposeful abstraction of reality, capturing constraints and assumptions resulting in a conceptualization of the problem to be solved, the environment in which it has to be solved, and relevant actors, their behavior, and the relationships of interest to solve the problem. *Simulations* are implementations of models executable over time, which means that they allow what-if-analysis, evaluation of alternative approaches, etc.

The modeling side of M&S is often identified as the conceptual model on which the system is built. Robinson (2006) defines this as *"a non-software specific description of the simulation model that is to be developed, describing the objectives, inputs, outputs, content, assumptions and* *simplifications of the model."* As Robinson (2006) furthermore points out, there is a significant need to agree on how to do this and capture this information technically. In the worst case, conceptual models exist only implicitly in the mindset of the developers and are not captured at all. In many cases, the conceptual model is captured as prose in a very loose and nebulous form. However, as Davis and Anderson (2003) point out, conceptual models should be documented based on engineering methods enabling their interpretation and evaluation by other engineers, or in Davis and Anderson's words a *"fully specified but implementation independent model."* Only if the conceptual model is captured using agreed to sufficient technical means, it is possible to evaluate if two systems are aligned conceptually or not.

The simulation side of M&S was the focus of interoperability work of the recent past. Both current solutions, IEEE1278 as well as IEEE1516, target the *interoperation of systems*, not the *alignment of concepts*. The current standards treat simulation systems as black boxes that exchange data. In particular in the domain of web-based simulation, resulting shortcomings become clearly visible, as shown in Pullen et al. (2004). Yilmaz (2004) comes to the same conclusions when he shows that the introspection of simulation systems needs to be specified in order to support the meaningful composition of systems. Hofmann (2004) shows as well that simulation interoperability is not sufficient, but that information exchanged must take different semantics and pragmatics into account, that are captured in Common Conceptual Models.

These observations motivate the use of different names and definitions for the different layers of interoperation, as proposed by Page at al. (2004), although they envisioned these aspects to define dimensions, not layers as proposed in this paper.

- *Interoperability* deals with the software and implementation details of interoperation, including exchange of data elements based on a common data interpretation, which can be mapped to the levels of syntactic and semantic interoperability. Here we are on the simulation side of M&S, how the models are actually implemented and executed.
- *Composability* addresses the alignment of issues on the modeling level. The underlying models are meaningful abstractions of reality used for the conceptualization being implemented by the resulting simulation systems. Composability deals with the Contextualized Introspective of Common Conceptual Models.

In summary, current research findings support the recommendation to speak of *Interoperability of Simulations* versus *Composability of Models*. For the later, information exchange between systems must be specified as contextualized information. Technology driven and interoperability specific solutions are not sufficient to support composability. Semantics and pragmatics need to be captured as well. The means of the ontological spectrum, as documented by Obrst (2006) and others can support this requirement.

3 THE ONTOLOGICAL SPECTRUM

The ontology spectrum was introduced by Daconta et al. (2003) and describes a range of semantic models of increasing expressiveness and complexity. In one of the earlier reports Mizoguchi and Ikeda (1996) stated that ontology is an important area of research and application for various areas in information science where the *specific and unambiguous meaning of data* needs to be captured. Entities within a domain can be understood in terms of their conceptual meaning as well as their relationships to each other. In the semiotic trichotomy developed by Charles Morris, Rudolph Carnap, and C. S. Peirce in the 1930s, syntax addresses the formal relations of signs to one another, semantics the relation of signs to what they denote, and pragmatics the relation of signs to their users and interpreters. Levinson (1983) describes their work as well as applications of their findings.

This aspect leads to distinguishing among the *real world referent*, the *concept* that represents this referent in the model, and the *entity* that implements the concept in an application. These three views describe what is modeled (world view), how it is modeled (conceptual view) and how it is implemented (systems view). This layered categorization capturing the difference between idea, concept, and implementation is comparable to using an operational view and a systems view to describe system architectures or to distinguish between the logical view and the physical view in data models. Alternative views of the trichotomy include the distinction between the *concept* itself and its *role* in an application. All categorizations allow separating the views on several layers and identifying differences and commonalities in a formally specified way. This trichotomy and its implications for model composition and system interoperability will be a topic of the next section.

Without doubt, ontology has many facets and we can only deal with a small fraction. Within this paper, we focus on an application driven approach and try to contribute to answering the question: *How can ontology help us to overcome the challenges of M&S composability and interoperability?* Therefore, we use the following working definition for ontology: *Ontologies are formalizations of specifications of conceptualizations.* All three parts of this definition are important for applications:

- The objective of ontologies is to document the *conceptualization*, which is another word for the result of the modeling process.
- This is done in a *specified way*, which means the application of engineering methods guided by rules and methods.
- The result is *formalized*, which means that machines and computers can not only read the result, but also

make sense out of it in the context of their applications.

As formulated in Tolk and Blais (2005) for practical applications: *"If a formal specification concisely and unambiguously defines concepts such that anyone interested in the specified domain can consistently understand the concept's meaning and its suitable use, then that specification is an ontology."* As with composability and interoperability, there is no cookie-cutter function that can be used as a yardstick for all applications. A layered approach is necessary which was introduced as the ontological spectrum. We use a simplified view on the spectrum. Interested readers are referred to Daconta et al. (2003) and Obrst (2006) for detailed versions. The following categories are used in this paper: controlled vocabularies, thesauri, taxonomies, ontologies, and logical models. Figure 1 shows how these categories lead from weak semantics to strong semantics using increasingly metadata to capture the information required.

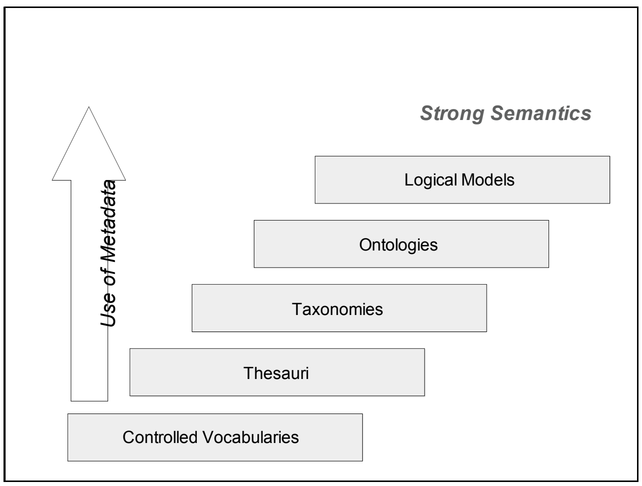

Figure 1: Simplified Ontological Spectrum

The categories start on the simple definition level where common vocabularies are defined. They lead to logical models that represent the respective conceptualization in a formal way including business rules and all layers of the trichotomy in a formal way.

- Dictionaries and glossaries are lists of *controlled vocabularies* and are among the weak semantics in the ontological spectrum. All allowed terms and their meanings are completely enumerated, well-defined and controlled by a common registration authority. Sometimes, additional information, such as pronunciations, etymologies, and variant spellings, are given or cross-references are included, but the overall structure is a flat list. Furthermore, these lists enumerate terms, not underlying concepts.
- *Thesauri* are controlled vocabularies arranged in a known order and structured so that equivalence, homographic, hierarchical, and associative relationships among terms are displayed clearly and identified by standardized relationship indicators. The primary purpose of thesauri is to facilitate retrieval of documents and achieve consistency in the indexing of written or otherwise recorded documents and other items. As with controlled vocabularies the focus is on the terms, not the underlying concepts, but a structure is introduced.
- *Taxonomies* are tree structures of classifications for a given set of objects. At the top of these structures are single classifications, which are the root nodes that apply to all objects. Nodes below these roots are more specific classifications that apply to subsets of the total set of classified objects. The main purpose is the classification of terms. The higher a term, the more universal it is; that means that leaves are the most specific terms of taxonomies. Taxonomies are the first form reflecting the idea of concepts. Also, the tree structure allows to capture first ideas of trichotomy.
- *Ontologies* formulate an exhaustive and rigorous conceptual schema within a given domain. Although most practical examples are typically hierarchical data structures containing all the relevant entities, they are not necessarily trees. In addition to entities, ontologies contain relationships and rules, such as theorems and regulations, within those domains. Ontologies capture the meaning of the underlying concepts and handle the trichotomy explicitly. In practice it is agreed that ontologies should contain at a minimum not only a hierarchy of concepts organized by subsumption relations, but also other 'semantic relations' that specify how one concept is related to another.
- *Logical models* are representing semantically the strongest methods of the ontological spectrum. Description logic, first order logic, and modal logic belong to this group. Furthermore, logical models can be separated into frame-based and axiomatic models. Frame-based models use an associated-node structure representing the logical expressions. Entity classes, attributes, properties, relations/associations, and constraints/rules are in the center. Axiomatic approaches make axioms and rules explicit, which means that they use languages exposing logical expressions.
  Furthermore, a very close relation to means of artificial intelligence in general and knowledge representation in particular becomes obvious. The work on Sowa (2000) is used as an example for these close relations.

The next section will show how these ideas can be used to contribute to composability of models and interoperability of simulations, focusing particularly on those aspects that are currently not captured by standardized means in IEEE1278 and IEEE1516: using trichotomic principles of ontology to model concepts, properties, primitives, and rules, multi-resolution challenges, and capturing pragmatic and dynamic challenges during runtime-exchange of information.

4 APPLYING ONTOLOGICAL MEANS FOR CONCEPTUAL MODELING IN SUPPORT OF COMPOSABILITY AND INTEROPERABILITY

As pointed out before, models are purposeful abstractions of reality. Every simulation is based on a model. *If a simulation is supporting a domain successfully, the underlying model and its abstraction make sense in the realm of this domain.* If an application models a concept in a very special role, this may reflect the add-on benefit of this model to a federation. This special role may nowhere else be reflected. This special abstraction is the essence of each model and makes up the value. The trichotomy captured by ontology reflects this. The current standards are missing these dimensions. This allows to share concepts, but now their roles in the applications; they allow to share concepts, but not the implementation details of entities; they allow sharing semantics (meaning of data), but not pragmatics (intention). Oren et al. (2007) distinguish between various categories of understandings, which are

- Lexical understanding (recognizing the symbols and tokens),
- Syntactical understanding (recognizing the elements describing an entity),
- Morphological understanding (recognizing the structure o the properties describing the underlying concepts, attributes of the entities),
- Semantic understanding (recognizing the meaning of entities and their relations), and
- Pragmatic understanding (recognizing the intention behind using the identified entities).

Tolk et al. (2006) show that very similar structures are needed on the metadata level in order to support composable M&S services. The Levels of Conceptual Interoperability Model (LCIM) distinguishes the following levels

- Technical Interoperability (exchanging bits and bytes),
- Syntactic Interoperability (using common protocols, exchanging data),
- Semantic Interoperability (using common data models, understanding meaning),
- Pragmatic Interoperability (using common workflow models, understanding the intent of sending and context of using of data);
- Dynamic Interoperability (using common execution models, understanding the effects of exchanging data in the sending and receiving services), and
- Conceptual Interoperability (using common conceptual models, being aware of abstractions, constraints, concepts, relations, and roles).

Wache et al. (2001) already showed that the trichotomy aspects captured by ontology are needed to integrate information (data in context). Gnägi et al. (2006) used these concepts to set up conceptual models for data centric applications. In the next subsections, we will extend this idea to show aspects of conceptual models for M&S, which are model-centric solutions and as such more complex than data-centric solutions. However, as the exchange of data within the models is a necessary requirement, we will start with these ideas.

4.1 Concepts, Properties, and Primitives

In the case of a taxonomy as defined above under the ontological spectrum, as well in more expressive ontological artifacts from the spectrum, it is likely that the formal organization of all ontological-entities in the ontology (concepts and properties) will be organized based on the conceptual meaning of those ontological-entities. This meaning is rooted in the trichotomy relationship of concept to ontological-entity, and makes it possible for the definition of the meaning within a system-entity (entities and attributes) to be made explicit. However, in the ontological artifact, it becomes necessary to quantify the product of this relationship, in order to export it to a system-entity.

An ontological model based on conceptual meaning of its ontological-entities will most likely structure those ontological-entities into an acyclic directed graph, showing parent-child relations between concepts. What differentiates this from a true tree structure is that a child ontological-entity may have multiple parents, inheriting conceptual meaning from all of them. This accommodates the organization of the relationship, but still does not cover the quantification of the conceptual difference between ontological-entities. For that purpose, primitives of meaning, or simply primitives, are relied on.

Primitives are within the community that will be using the universally accepted ideas representing one aspect of meaning in the definition of an ontological-entity. Each ontological-entity necessarily inherits all of the primitives of its parent ontological-entities, and has at least one new primitive ontological-entity introduced that provides the conceptual difference between it and its parents. In this way, each ontological-entity in the ontology can be differentiated from all other ontological-entities by having its primitives enumerated. Additionally, each ontological-entity that is a child, and has siblings (defined as having the same parents), must be differentiated from each other by having a difference in conceptual meaning – a difference that is measured by having distinct primitives from each other. Primitives are exposed as properties in the context of a special world-view. While properties represent the interpretation of a primitive in a special (system specific) view, the primitive captures the general view and con be exposed in several views differently.

As the defined ontological-entities of the ontology representation provide the definition of system-entities within an application, those entities are defined within an application by not only a term, but also by a collection of associated properties, which have values. These properties each have their origin in a primitive. When comparing the system-entities of different systems, for purposes of data engineering, having an understanding of the primitives that each systems ontology defines for a system-entity makes the engineering process easier, as differences in meaning become readily apparent.

It is likely, of course, that system-entities will have identity through the exposure of different values for the properties that they exhibit. It appears to be not possible that two system-entities, based on the same ontological-entity for meaning, can have identical property-values. There seems to be a requirement that some difference must exist, even if it is only in the application of a different value to the name property for the system-entity. In this way we can see that all ontological-entities are made distinct and discrete from each other by the complete enumeration of their respective primitives, and all system-entities are made distinct and discrete from each other by the complete enumeration of their respective property-values.

The definition of primitives, property-values, properties, system-entities, and ontological entities and their relationships allow to capture all necessary information to identify if two system-entities are clearly expressing the same underlying concepts or if they are homonyms (same or similar word with different meaning) or synonyms (different words with same or similar meaning). Furthermore, the degree of similarity can be determined by the degree of overlapping properties and primitives. Figure 2 shows examples for these ideas.

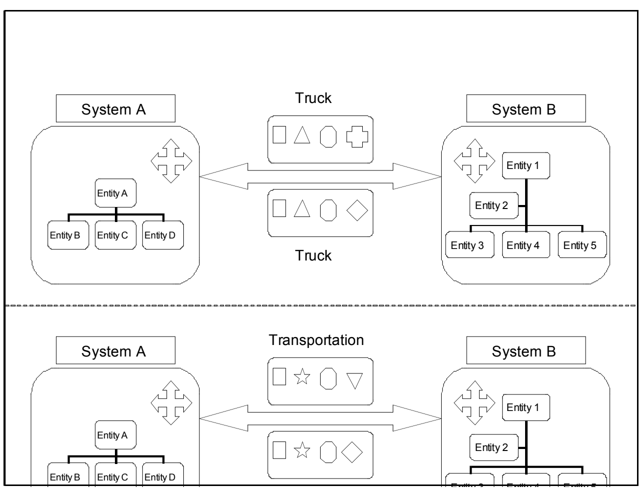

Figure 2: Examples of Alignment

The upper part of the figure shows two system-entities both called "Truck," but the exposed properties are not identical (although sufficiently similar to justify calling both entities truck). This can be used to identify homonyms. The lower part shows two system entities that exposed very similar properties, but use different terms. This is an example for a synonym. The next subsection will extend these principles to a general application.

4.2 Multi-resolution Challenges

Multi-resolution modeling is a domain of special interest to ontology-driven interoperability and composability, as the application of ontological means contribute to solving challenges in this domain. In general, multi-resolution modeling copes with the challenge when two simulation systems represent the same concept on different levels of resolution, with a different scope, are if the expose different structures. Many recommended solutions focus on the different levels of resolution. While in the low resolution level only a few attributes represent the state of the implementation of the concept, on the high resolution level several entities and respective attributes represent the same concept. Reynolds et al. (1997) introduced the idea of Multi-Resolution Entities (MRE) to serve as mediators between the different levels of resolution. Figure 3 shows the principles of an MRE.

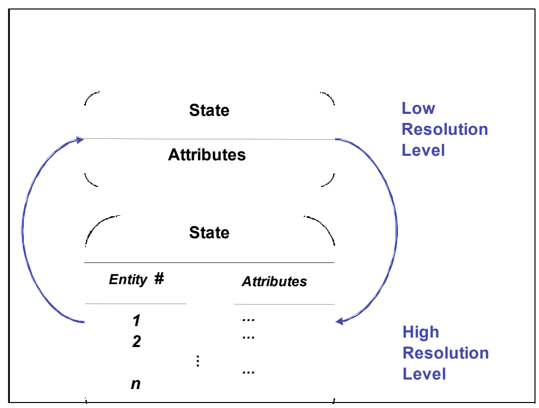

Figure 3: Multi-Resolution Entity

The MRE serves two purposes: (a) it documents the mapping between the low resolution entity attributes to the high-resolution entities and attributes, and (b) it serves as "memory" for the information that normally gets lost in the process of aggregation.

As pointed out before, ontological means allow to model various forms of trichotomy explicitly, in particular real world references, concepts, and entities, as well as properties, property values, and primitives; and they are machine understandable. This ability can be used to extend the ideas of MREs into Trichotomic MREs (TMREs) which allows a new way to compose simulations into a system-of-systems. Each model comes with data reflecting the purposeful abstraction of reality on which it was built. These data implement the entities reflecting the concepts that are derived from the real-world reference. Each entity is a collection of characterizing attributes with a set of valid attribute values. In addition to these characterizing attributes, non-characterizing attributes are possible. The later comprise non-characterizing and optional information. The entities and their attributes are the finest elements of information in the simulation. In addition, entities can be related with other entities to build higher elements of information. The relation between to entities can be characterized by additional attributes that are not part of any of the single entities but make only sense for the relation. Hierarchical information – such as "is superior of" – makes no sense within a single entity but only within the relation between two entities. A shown in section 4.1, all this information can be caught in machine understandable form.

We propose the use of TMRE to capture all these data for every participating system. The TMRE documents each system to the highest degree currently possible, as pragmatics, morphology, semantics, and syntax are captured and can be mapped to each other on the respective levels. Homonyms and synonyms are identified. Even if concepts are only reflected in primitives without explicitly modeling the concepts, these relations can be captured in the trichotomic structure within TMREs. Figure 4 exemplifies these principles by showing the different ontological artifacts and their use within the TMRE.

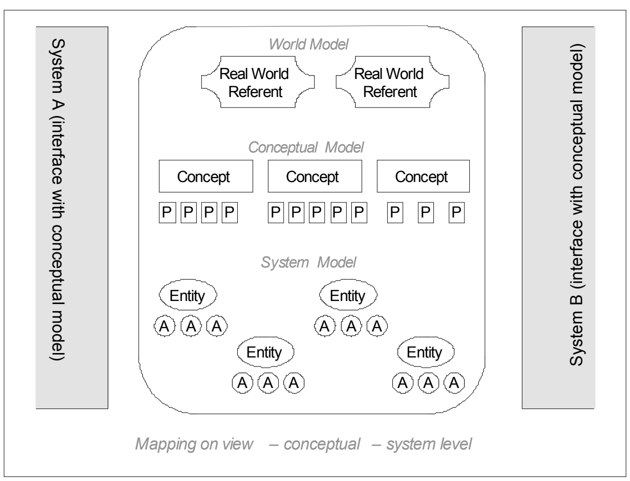

Figure 4: Trichotomic Multi-Resolution Elements

TMREs allow the highest degree of alignment between systems possible, as down to the level of primitives alignment can be based on data engineering principles and no interpretation gaps are left to implementation decisions of system developers based on only partially appropriate federation object models or protocol data units. They allow to document if two models use the same abstraction in their world views (real world referents), if they use the same concepts to reflect their world view (concepts and parameters), and how the implement this ideas (entities and attributes). The idea of primitives allows in addition to reflect abstractions of other world views even if only attributes and/or properties are used to reflect them (see section 4.1 for details on this principle). As mapping is possible and documented in machine readable form on all levels, TMREs can capture the degree of information exchange possible between all systems.

4.3 Pragmatic and Dynamic Challenges

Among the current challenges is to specify languages to be used between composable services. Examples are Banerji et al. (2002) for the general case. Tolk et al. (2006) extend this idea and apply a common reference model for military applications. Sudnikovich et al. (2004) envision a *Battle Management Language* (BML) to allow the unambiguous communication of tasks and report at runtime to connect command and control systems, simulation systems, and robotic systems conducting military operations.

There is no general solution established. Arsanjani (2005) envisions that service oriented architectures will develop from remote services over service adaptors and service proxies to virtual providers and service integrators. The final state is an enterprise service bus on which services are mediated to speak a common language enabling the composition. The Enterprise Service Bus (ESB) structure is shown in Figure 5.

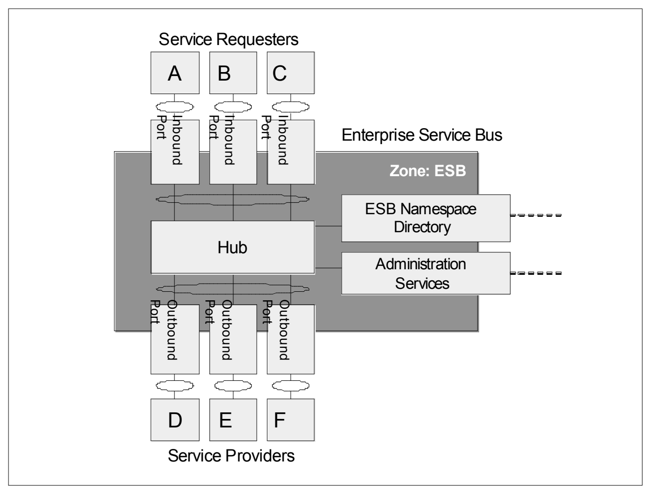

Figure 5: Enterprise Service Bus (ESB)

The solution is based on similar principles as identified for federated databases in Spaccapietra et al. (1992), but it adds the component of orchestration of services. However, all solutions agree that a common language is necessary, which requires to apply the principles explained before.

The requirements for a language that can be used to describe services and their information exchange requirements and information exchange capabilities for selection, choreography, and orchestration comprise the following:

1. The description must be parseable and computable.
2. The description must be complete, consistent, and unambiguous.

The reason for these requirements is that the receiving services are unlikely humans who can deal with incompleteness, inconsistency, and ambiguity. It is possible that strong Artificial Intelligence (AI) will support necessary interpretations or transformations. Such a language is also more than interesting for software agents, as shown in Oren et al. (2007).

For such a language to be used during runtime, more requirements than coped with so far are needed. The internal rules of the implementing systems regarding consistency and completeness of data as well as the rules regarding the information exchange requests and capabilities within the federations need to be satisfied for every single information exchange. If the structure of the receiving system requires a well defined set of entities and each entity requires a set of properties, a sending system can only than share the information if all the additionally required data is delivered as well. The TMREs can support to identify such constraints, but the implementing services need to satisfy the constraints.

It is not only possible but likely that for real-world applications the data grow rapidly and the resulting sentences of the language become extreme verbose. Often, a lot of this information is redundant when compared with earlier messages. The necessity for completeness requires, e.g., that the type information of communicated objects is communicated as well, as the type may have changed. In order to cope with this challenge, it is possible to implement a sort of memory with the services or in form of persistent data for the federation (such as a central database as described by Sudnikovich et al., 2004). This results easily in bottlenecks and is only of limited use in distributed systems. Memory within the services adds to the complexity of their implementation and may reduce the efficiency of their implementation (including mobility, etc.). Alternatively, the ideas of patterns and frames as known from AI can help. Karp (1993) introduces the application of patterns and frames in connection information management tasks similar to those described here. This approach defines patterns of typical data that can be assumed to be applicable in a given context (captured by data describing the pragmatics). Only if the observed data differs from the assumed data it needs to be communicated. Such an approach allows to reduce the information amount to be exchanged without violating the requirement for completeness.

4.4 Self-organizing Information Exchange

As all information exchange requirements and capabilities are captured in machine-understandable form, the principles of agent-mediated integration are supported. Intelligent software agents can use the information captured regarding concepts and properties and represented world referents and their implementation as entities and attributes – plus primitive, relations and rules – to identify matches. The conceptual models are not "flat" like most current approaches, but have several dimensions allowing to apply advance reasoning algorithms to identified transformations and mediations between systems with minimal human interaction. So far, only very simple problems have been addressed using automatic procedures, see, among others, Su et al. (2001), but with a semantically rich conceptual model more options become possible.

Current approaches do not address these problems, as it is assumed that the participating systems somehow fulfill all constraints. How this is accomplished is often left to the system programmer. All these constraints are supported by artifacts captured in the ontological spectrum reaching from weak to strong semantics and now can at least be captured in a formal conceptual model. Nonetheless, additional research is necessary and many gaps in the Body of Knowledge are still open. We hope to contribute constructively to the discussion with the paper.

5 SUMMARY

Conceptual modeling using the artifacts of the ontological spectrum will help to close several gaps identified in Robinson (2006), particularly by introducing common artifacts to capture concepts in a computable way. Conceptual models based on trichotomical principles captured in ontologies are in particular supportive to deal with challenges of multi-resolution modeling in a service oriented context. They enable the definition of a language based on the real information exchange request and information exchange capabilities of participating systems.

It should be pointed out that if the operationally required information exchange is captured, this view should be used as an additional system to unambiguously identify referents, concepts and properties, and entities and attributes. Instead of assuming that systems somehow will reflect these data, the means described in this paper can capture what systems can really exchange and understand. The approach captured here defines a conceptual model unambiguously describing what systems *can* understand while mandating approaches describe what systems *should* understand. The latter requires composability and interoperability, but does not enable them.

ACKNOWLEDGMENTS

The underlying research efforts leading to the results documented in this paper were partly funded by the US Joint Forces Command and the US Army Test and Evaluation Center.

REFERENCES

Arsanjani, A. (2005). Toward a pattern language for Service-Oriented Architecture and Integration. IBM DeveloperWorks, SOA and Web Services

Banerji, A., Bartolini, C., Beringer, D., Chopella, V., Govindarajan, K., Karp, A., Kuno, H., Lemon, M., Pogossiants, G., Sharma, S., and Williams., S. (2002). Web Services Conversation Language (WSCL). W3C Note Mar 2002

Daconta, M., Obrst, L., and Smith, K. (2003). The Semantic Web: The Future of XML, Web Services, and Knowledge Management. John Wiley, Inc.

Davis, P.K. and Anderson, R.H. (2003). Improving the Composability of Department of Defense Models and Simulations. RAND Corporation

Gnägi, H.R., Morf, A., and Staub P. (2006). Semantic Interoperability through the Definition of Conceptual Model Transformations. Proceedings 9th AGILE Conference on Geographic Information Science, pp. 128-138

Hofmann, M. (2004). Challenges of Model Interoperation in Military Simulations. SIMULATION, Vol. 80, pp. 659-667

Karp, P.D. (1993). Frame Representation and Relational Data Bases: Alternative Information-Management Technologies for Systematics. Advanced Computer Methods for Systematic Biology: Artificial Intelligence, Database Systems, Computer Vision, pp. 560ff, Johns Hopkins University Press

Levinson, S.C. (1983). Pragmatics. Cambridge University Press, Cambridge

Mizoguchi, R., and Ikeda, M. (1996). Towards Ontology Engineering. Osaka University Technical Report AI-TR-96-1, I.S.I.R.

Obrst, L. (2006). The Ontology Spectrum and Semantic Models. MITRE Information Semantics Group, Information Discovery & Understanding, Center for Innovative Computing & Informatics

Oren, T., Ghassam-Aghaee N., and Yilmaz, L. (2007). An Ontology-based Dictionary of Understanding as a Basis for Software Agents with Understanding Abilities. Proceedings 2007 Spring Simulation Multiconference, IEEE Press

Page, E.H., Briggs, R., and Tufarolo, J.A. (2004). Toward a Family of Maturity Models for the Simulation Interconnection Problem. Proceedings IEEE Spring Simulation Interoperability Workshop, IEEE CS Press

Pullen, J.M., Brunton, R., Brutzman, D.P., Drake, D., Hieb, M.R., Morse, K.L., and Tolk, A. (2004). Using Web Services to Integrate Heterogeneous Simulations in a Grid Environment. Proceedings International Conference on Computational Science 2004: 835-847

Reynolds Jr., P.F., Natrajan A, and Srinivasan, S. (1997). Consistency Maintenance in Multiresolution Simulations. ACM Transactions on Modeling and Computer Simulation, 7(3): 368–392

Robinson S. (2006). Issues in Conceptual Modeling for Simulation: Setting a Research Agenda. 2006 OR Society 3rd Simulation Workshop, Ashorne, UK

Sowa, J. (2000). Knowledge representation: logical, philosophical, and computational foundations. Brooks Cole Publishing Co., Pacific Grove, CA

Spaccapietra, S., Parent, S. and Dupont, Y. (1992). Model Independent Assertions for Integration of Heterogeneous Schemas. Very Large Database (VLDB) Journal 1(1): 81-126

Su, H., Kuno, H. and Rundensteiner, E. (2001). Automating the transformation of XML documents. Proceeding 3rd International Workshop on Web Information and Data Management, ACM Press, pp. 68-75

Sudnikovich, W.P., Pullen, J.M., Kleiner, M.S. and Carey, S.A. (2004). Extensible Battle Management Language as a Transformation Enabler**.** SIMULATION 80: 669-680

Tolk, A., and Blais, C.L. (2005). Taxonomies, Ontologies, and Battle Management Languages – Recommendations for the Coalition BML Study Group. Proceedings IEEE Spring Simulation Interoperability Workshop, IEEE CS Press

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

Wache, H., Vogele, T., Visser, U., Stuckenschmidt, H., Schuster, G. Neumann, H., and Hübner, S. (2001). Ontology-based Integration of Information – a Survey of Existing Approaches." Proceedings of the IJCAI-Workshop Ontologies and Information Sharing Seattle, WA: 108-117

Yilmaz, L. (2004). On the Need for Contextualized Introspective Simulation Models to Improve Reuse and Composability of Defense Simulations. Journal of Defense Modeling and Simulation (JDMS) 1(3): 135-145

AUTHOR BIOGRAPHIES

ANDREAS TOLK is Associate Professor for Engineering Management and Systems Engineering of Old Dominion University's Modeling, Simulation, and Visualization Faculty. He is also a Senior Research Scientist at the Virginia Modeling Analysis and Simulation Center (VMASC). He holds a M.S. in Computer Science (1988) and a Ph.D. in Computer Science and Applied Operations Research (1995), both from the University of the Federal Armed Forces of Germany in Munich. He is a member of SCS and SISO. His e-mail address is <atolk@odu.edu>.

CHARLES D. TURNITSA is a Ph.D. candidate at the Virginia Modeling Analysis and Simulation Center (VMASC) of the Old Dominion University (ODU). He received his B.S. in Computer Science (1991) from Christopher Newport University (Newport News, Virginia), and his M.S. in Modeling & Simulation (2006) from ODU. His Ph.D. research under Tolk focuses on the domain of dynamic and fractal ontology models for M&S interoperability. His e-mail address is <cturnits@odu.edu>.