Proceedings of the 2009 Winter Simulation Conference
M. D. Rossetti, R. R. Hill, B. Johansson, A. Dunkin and R. G. Ingalls, eds.

TOWARDS ONTOLOGY-DRIVEN INTEROPERABILITY FOR SIMULATION-BASED APPLICATIONS

Perakath Benjamin
Kumar Akella

1408 University Drive East
Knowledge Based Systems, Inc.
College Station, TX 77840, USA

ABSTRACT

This paper describes an ontology-driven approach for the design and execution of interoperable simulation applications. The premise of the research described in this paper is that ontologies provide the foundation for determining semantic interoperability and information exchange requirements. Two fundamental problems are inherent to simulation application integration: (i) Semantic Inaccessibility: caused by the failure to explicitly specify the semantic content of the information contained within each simulation subsystem; and (ii) Logical Disconnectedness: caused by the failure to explicitly represent constraints between the information managed by the different simulation subsystems. The paper will (i) describe the technical challenges that motivate our research; (ii) describe how ontologies enable effective simulation application mediation and interoperability; (iii) describe an automation approach for ontology creation from simulation system description data; and (iv) present a real world application example to illustrate the practical benefits of the solution ideas.

1 MOTIVATIONS AND BACKGROUND

1.1 Background and Problem Description

One of the key motivations for distributed, federated simulation modeling is to allow for the decomposition of the target system into smaller, more manageable components, and to distribute the model development effort among different organizations/functional groups. Once the component simulation models have been developed, there is a need for mechanisms to assemble a simulation model of the entire target system in a manner in which the “whole (system) = sum of its components/ parts.” Past and ongoing investments in distributed simulation and the High Level Architecture (HLA) have been motivated significantly by this idea. The idea of distributed simulation is also critical to the success of Simulation Based Acquisition (SBA) with large organizations such as the Department of Defense (DoD) and the National Air and Space Administration (NASA). In spite of the significant federal investments for developing science-based frameworks, architectures, methods, and tools to facilitate distributed and federated simulations, significant technical and pragmatic voids still remain.

There are two related problems with regard to simulation interoperability: *translatability* and *composability*. Translatability means that a simulation model in one simulation tool or application can be *easily* and *correctly* imported into another simulation without loss of information. Because there are a myriad of simulation tools and applications, each with its own format, and, to boot, various versions of the format, it is extremely difficult for these applications to share scenario data. Incorrectness of translated scenario data and the loss of information due to language limitations are two translatability problems. Composability, on the other hand, is the capability to select and assemble simulation components in various combinations into simulation systems that satisfy specific user requirements (Petty and Weisel 2003). The components to be composed/assembled are often drawn from a repository/library. The components may vary from “legacy code and data” to well-defined software components/models with “standards-based” protocols (such as the HLA protocols) and Application Programming Interfaces (API’s). Composability enables users to combine and recombine, and configure and reconfigure, components in numerous ways to satisfy diverse needs and requirements.

In a distributed, federated simulation environment, the problems of translatability and composability are related because (i) composability of simulation modules may require translatability of modules due to specific environments under which the modules may be run in the federate, and (b) translation of a simulation module that is required to run in a different environment renders existing compositions (that include the original module) invalid.

978-1-4244-5771-7/09/$26.00 ©2009 IEEE

There are two forms of interoperability: *syntactic* and *semantic*. Syntactic interoperability deals with the interoperability of implementation details such as parameter passing mechanisms, external data accesses, and timing mechanisms. Semantic interoperability, on the other hand, deals with the validity and usefulness of translated/composed simulation models.

1.2 Related Work

An event graph methodology was used to design and execute component-based Discrete Event Simulation models, where Discrete Event Simulation models are specified as components [also called Listener Event Graph Objects (LEGOS)], which are designed to work in conjunction and facilitate the composability of models (Buss 2007). Each LEGO is configured to hold its own distinct parameters and is responsible for the events and state transitions that modify its state variables and produce its state trajectories. These LEGOS are loosely connected by "listener patterns" to facilitate communication in a framework. The listener component and the event source component are associated with each listener pattern, where the listening component registers interest in another component's events and waits for the other component to execute the event. A distributed simulation based approach for supply chain interoperability testing has been studied to investigate the testing of standards and the compliance of applications to the standards through simulation (Jain 2007). This work is motivated by the Virtual Manufacturing Environment project at the National Institute for Standards Technology (NIST) to help U.S. manufacturers with technical guidance and testing support to interoperate in the heterogeneous manufacturing world. A prototype was built to demonstrate the idea of interoperability for an automotive industry application involving supply chain simulation and assembly plant simulation. C. A. Boer surveyed COTS simulation package vendors and simulation experts from defense and commercial industries to understand issues relating to the distributed simulation achievable through HLA (Boer 2008). The three factors inhibiting simulation practitioners from designing distributed simulation models are (i) monetary cost – the purchase price of existing interoperation tools or the costs of developing such a solution; (ii) time – additional time for designing and developing models in a distributed way and applying interoperability solutions; and (iii) quality – added value of distributed simulation is countered by shortcomings in the quality of the resulting model.

The role of ontologies and their axiomatization is explored to achieve dynamic composability of disparate simulation models (Yilmaz and S. Paspuleti 2005; and Yilmaz 2006). This research indicates that principled designs are needed to facilitate interoperability and composability between simulation models that are dynamically extensible. Dynamic composability is feasible through the use and implementation of the intelligent brokering and matchmaking features of advanced model bases. Similar work by (Kokla 2006) analyzed and compared relevant integration approaches through the adoption of an ontological framework and described guidelines for achieving semantic integration of geographic ontologies. This work summarizes alternative strategies for ontology integration, including the following: (i) conforming to a single global ontology; (ii) manual ad-hoc mappings by codifying the rules for mapping concepts between ontologies; (iii) intuitive mappings based on shallow semantic (lexical) information; (iv) intuitive mappings based on explication characteristics (attributes, relations, constraints, etc.); (v) direct mappings based on deep semantics (using Natural Language Processing techniques); and (vi) determination of compound similarity measures (through comparison of terms/concepts, relations/properties/attributes, and instances).

The use of ontologies to codify knowledge and facilitate the development of simulation models is presented in (Silver 2007). A technique establishing relationships between domain ontologies and a modeling ontology and then using the relationships to instantiate a simulation model as ontology instances is discussed. Ontology instances are transformed into XML-based standards, which are translated into executable models for various software packages. The implementation of existing simulation models as federated objects to achieve systems of systems integration is discussed in (Kewley 2008), and the federation development process is detailed from information exchange through entity representation, model development, and data collection.

The use of OWL for representing object-oriented descriptions to support distributed representations of data, behaviors, descriptions of units and objects to be simulated, and scenarios with initial conditions is described in (Lacy 2004). (Fishwick 2004) provides an overview of the application of semantic web technology to Modeling and Simulation. Applying semantic web technology to simulation will facilitate data exchange and interoperability, annotations of simulation resources available on the Web using RDF, and the creation of ontologies for modeling and simulation.

2 STRATEGIES FOR ONTOLOGY-DRIVEN SIMULATION INTEROPERABILITY

Conceptually, simulation interoperability may be divided into two broad categories: (i) Design Time Interoperability and (ii) Run Time Interoperability.

Design Time Interoperability

This category refers to cases in which interoperability is explicitly engineered into the simulation application from the start. Design time interoperability implies that interoperability is an important consideration through the simulation development life cycle – including problem identification, requirements definition, conceptual model design, detailed model design, verification, validation, experiment design, experiment execution, and post simulation analyses.

Run Time Interoperability

This category refers to cases in which interoperability is implemented after the fact with simulation models and applications that were developed independently. Often, run time interoperability is done with legacy model and application data; the models and applications in these situations are developed without prior intent to interoperate.

Figure 1 shows our overall approach for accomplishing simulation interoperability that accounts for both the above types of situations.

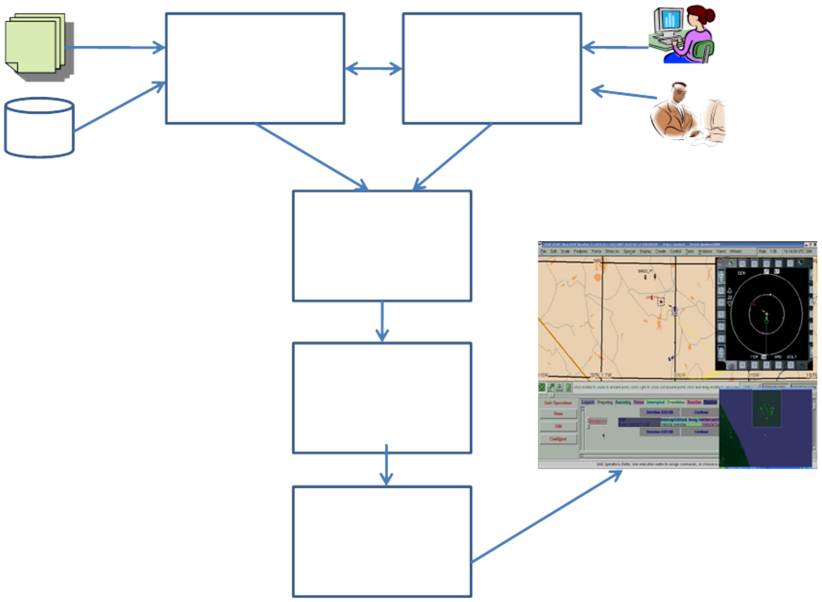


Figure 1: Ontology-driven simulation interoperability approach

2.1 Extract Ontology Models

This activity automatically extracts the elements of ontology models from multi-source data. Our method allows for ontology extraction from both unstructured and structured data types. Note that automatic ontology extraction is very relevant to run time interoperability situations, often requiring the need to generate ontology models from legacy simulation system descriptions (requirements documents, design documents, test documents, etc.).

2.2 Elicit Ontology Models

This activity acquires ontology information from human subject matter experts. Automated assistance is often provided to increase the efficiency of this activity. Often, human input is also required to validate and refine the ontology elements that are produced using the automated ontology extraction activity (Figure 1).

2.3 Perform Ontology Mappings

This activity identifies and details pair-wise correspondences between elements of the different simulation system ontology models. Multiple techniques are used to determine the inter-ontology mappings. The resulting ontology mappings are used to derive the simulation interoperability information flow requirements.

2.4 Determine Information Exchange Requirements

This activity uses the results of the ontology mapping activity to identify the information flow requirements between the different simulation systems within the simulation federation. This task is non-trivial, as information exchanges are often specified between models and data at multiple levels of abstraction. Our approach provides techniques for multi-resolution simulation integration, as described in (Benjamin et al. 2006). The information exchange requirements that are produced by this activity drive the design and implementation of inter-simulation mediation and translation mechanisms.

2.5 Execute Information Exchanges

This activity designs and implements the information exchange mechanisms between the different simulation systems/tools within the simulation federation. The inter-simulation information exchange mechanisms may be implemented in different forms, including (i) inter-simulation mediation services and (ii) inter-simulation translators.

3 A METHOD FOR ONTOLOGY-DRIVEN SIMULATION INTEROPERABILITY

This section describes a method for facilitating simulation interoperability in situations in which ontology models of component simulations *are not available a priori*. The key to the success of this approach is the ability to automatically extract ontology models from descriptive data relevant to the simulation applications. The activities of this method are described in the following paragraphs.

3.1 Perform Ontology Extraction

This activity involves extracting the elements of ontology models from text data sources. The goal is to determine ontologies of the simulation tools and/or the simulation application domains that need to exchange information with each other. The text data that are used to derive these ontologies include (i) requirements documents, (ii) design documents, and (iii) source code. Ontology extraction is accomplished using (i) Text Mining-based Methods, (ii) Statistics-based Methods, and (iii) Natural Language Processing (NLP)-based Methods. The text mining-based ontology extraction method is shown in Figure 2.

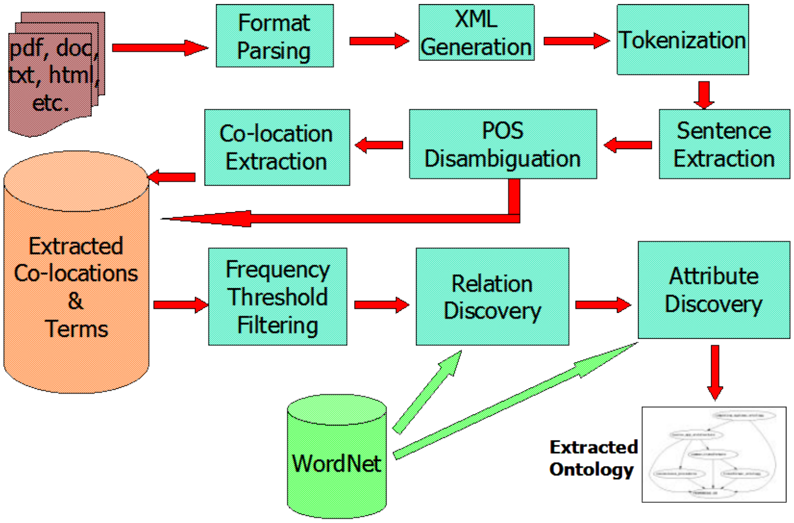


Figure 2: Ontology extraction process

3.1.1 Format Parsing and XML Generation

Raw text data retrieved from multiple data sources and formats is first pre-processed. The format types currently supported include TXT, DOC, RTF, PDF, and HTML. Once the contents from the raw text corpora (one or more files) have been processed, a “standard” XML-formatted file is generated.

3.1.2 Tokenization and Sentence Extraction

The format conversion activity is followed by multiple lexical processing tasks, including (i) tokenization and (ii) paragraph and sentence recognition. The purpose of tokenization is to separate the text stream into ‘words’ or tokens. Here a word, or token, is a word or number in the traditional sense, punctuation marks, and other items that may prove useful in extracting the semantics. Thus, a word is a “string of alphanumeric characters with a space on either side.” Paragraph boundary recognition is performed using a heuristic approach (identification of Carriage Returns, Line Feeds, Vertical Tabs, etc.).

Sentence boundary identification is accomplished using a Hidden Markov Modeling (HMM) approach. The HMM uses contextual information about the following features: (i) the probability of a part of speech of the surrounding words (one word to the left and one word to the right of the punctuation mark); (ii) abbreviation or an honorific (Dr., Mr., etc.) to the left of the punctuation mark; and (iii) the number of words between consequent sentence boundary candidates.

3.1.3 Part of Speech Disambiguation

First, we identify the Parts of Speech (POS) for the tokens in the document. The WordNet® lexical database is used to assist with this step. Note that a given word may have (i) multiple POS uses (e.g., “bomb” may be used as a noun or a verb) and (ii) multiple meanings for each POS use [e.g., the noun use of “launch” may be “take-off” (for instance, for missiles at Kennedy Space Center) or “presentation” (for instance, a sales briefing)]. POS tagging helps with the initial disambiguation relative to multiple meanings of a term, and with interpreting the roles of the terms in relationships (e.g., noun – noun relations vs. noun –verb relations). POS disambiguation narrows the scope of the possible meanings of a term. We have designed and implemented a Hidden Markov Model (HMM)-based approach for POS disambiguation.

3.1.4 Collocation Extraction

A collocation is an expression consisting of two or more words that correspond to some conventional way of communicating a thought. Collocations are characterized by limited compositionality. An expression is compositional if the meaning of the expression can be predicted from the meaning of the parts. Collocations are not fully compositional in that there is usually an element of meaning added to the combination. All two-place collocations of the types adjective-noun, noun-noun, noun-preposition-noun (proper nouns are treated as nouns) are extracted from the text. Here are examples of collocations: (i) weapons of mass destruction (noun-preposition-noun-noun) and (ii) cease fire agreement (noun-noun-noun).

3.1.5 Frequency Threshold Filtering

Once we have extracted preliminary lexical semantic content from the text corpora, the next objective is to extract concepts that are candidates for inclusion within an ontology. Initially, candidate ontology concepts include (i) words that have been tagged as nouns (e.g., mission, missile, etc.) and (ii) collocations (e.g., missile launch, radar system, etc.). Because the number of candidate concepts may potentially be large, the ontology extraction algorithm takes a “threshold frequency” as an argument. Concepts occurring with less than the specified threshold frequency are filtered out. Generally, the threshold value of 2 or 3 seems to provide good results (based on initial experimental evaluations). Filtering out low-frequency concepts is also a way to improve the efficiency of the algorithm.

3.1.6 Relation Discovery

Once the concepts have been filtered, candidate relationships between the concepts are identified using the WordNet® lexical database. WordNet® is an online lexical database whose design was inspired by current psycholinguistic theories of human lexical memory. English nouns, verbs, adjectives and adverbs are organized into synonym sets, each representing one underlying lexical concept. Different relations link the synonym sets. Different semantic relationships link the synonym sets as shown in the example in Figure 3.

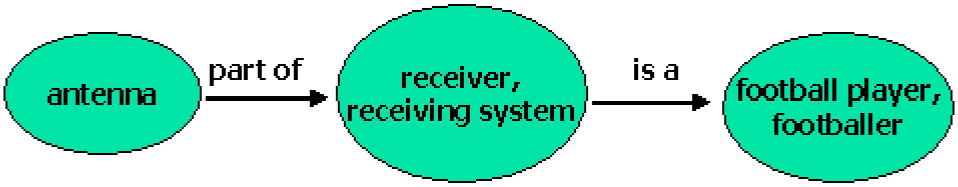


Figure 3: Example of relationships in WordNet®

The relation discovery module iterates for each concept ("source concept") and explores WordNet® for other concepts related to it. If any of the related concepts ("target concepts") from WordNet® exist in the pool of concepts extracted from the domain text, then a relationship is discovered between the source concept and the target concept, and the type of relationship as defined by WordNet®. Example relationships extracted from WordNet® include Is_A, Part_Of, Causes, Entails, and Member_Of. Figure 4 shows an example ontology fragment extracted from a document that describes Air-to-Air Combat operations.

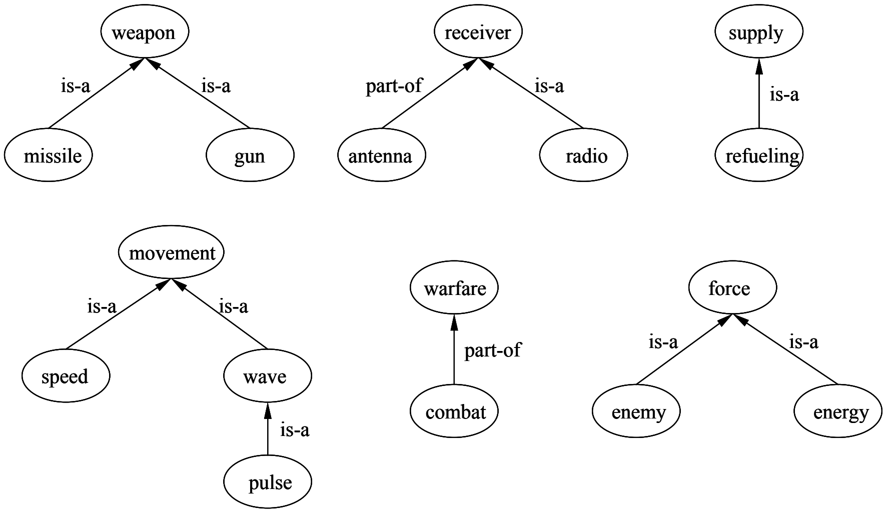
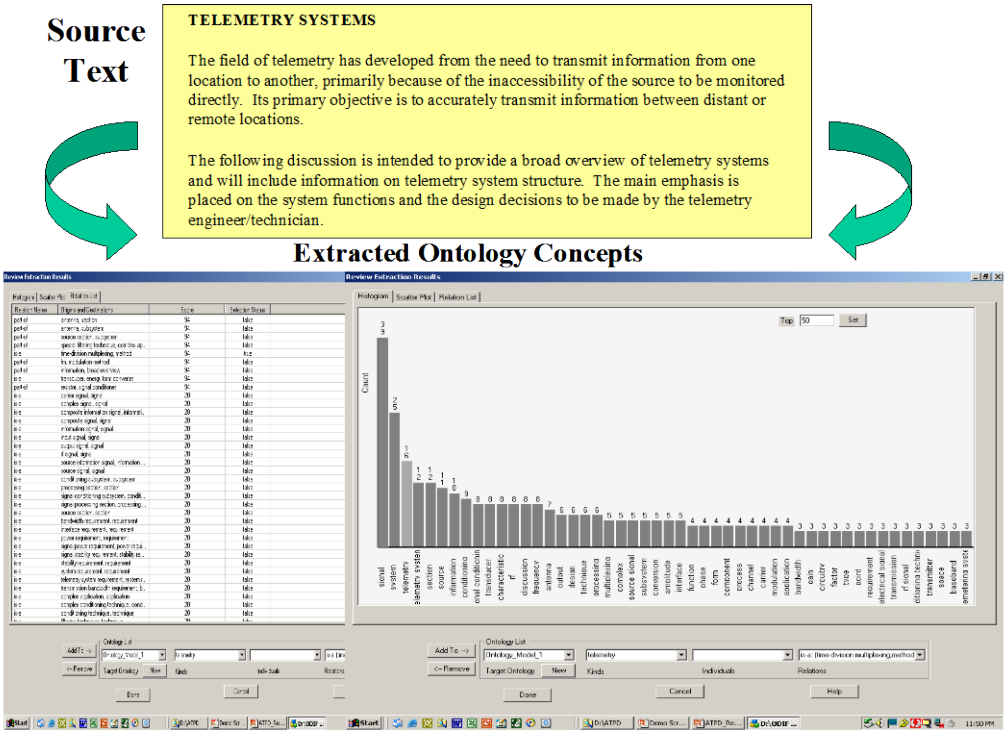


Figure 4: Example ontology fragment extracted using the text mining-based approach

3.1.7 Attribute Discovery

WordNet® is also used to discover the attributes associated with the extracted concepts. The WordNet® database contains a relationship called Is_An_Attribute_Of. In the attribute extraction step, the tokens that have been tagged as adjectives are considered as potential (candidate) attributes. The attribute discovery algorithm iterates for each concept (source concept) and explores the attribute relationship in WordNet®. If any related attribute found in WordNet® exists in the pool of attributes extracted from the domain text, it is reported as an attribute of that source concept.

Figure 5 shows an example ontology extraction User Interface display.



Source Text

TELEMETRY SYSTEMS

The field of telemetry has developed from the need to transmit information from one location to another, primarily because of the inaccessibility of the source to be monitored directly. Its primary objective is to accurately transmit information between distant or remote locations.

The following discussion is intended to provide a broad overview of telemetry systems and will include information on telemetry system structure. The main emphasis is placed on the system functions and the design decisions to be made by the telemetry engineer/technician.

Extracted Ontology Concepts

Figure 5: Example ontology extraction user interface

3.2 Perform Ontology Mappings

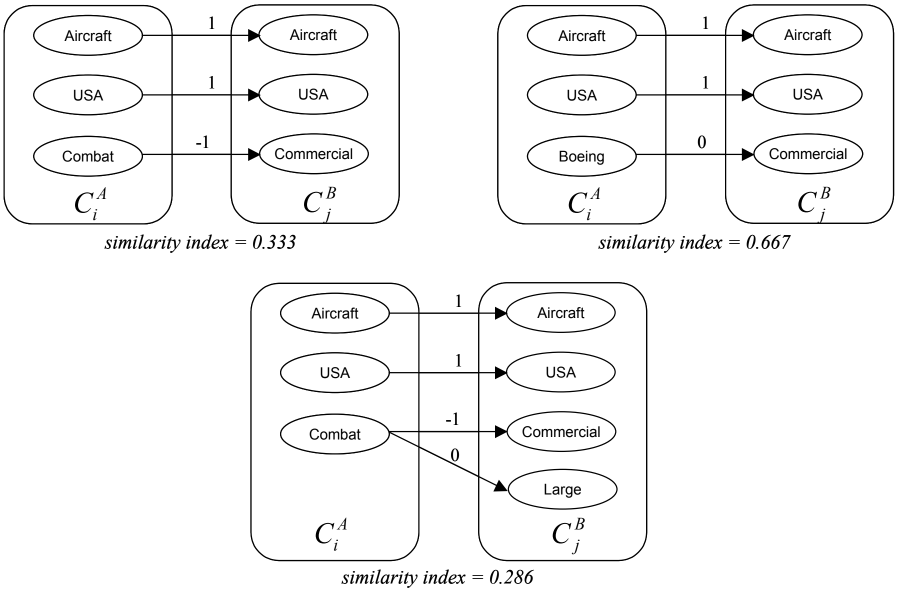
This activity involves determining correspondences between the ontologies of the different simulation tools/application domains (Patki, Benjamin, and Mayer 2007). Automation support is provided to assist this mapping activity. Multiple techniques are used to automatically determine ontology mappings, including: (i) Terminology: This analysis finds similarities between concepts based on the words used in naming the concept; (ii) Feature Based: This analysis finds similarities between concepts based on the attributes and attribute values; (iii) Topological or Graph Theory: This analysis finds similarities between concepts based on the graph structure (i.e., based on relationships the concept has with other concepts); and (iv) Semantics: This analysis finds similarities between concepts based on semantic analysis of "domain discourses or descriptions."

To provide a flavor for these methods, the *Terminology-based method* is based on the comparison of *names* of concepts, concept-attributes, inter-concept relationships, etc. When two ontologies are being compared for terminology-based similarity, each concept in one ontology must be compared with each concept in another. A concept can be represented by more than one *word*. These words need to be pre-processed to screen for synonyms. Pre-processing involves grouping together words with the same meaning and replacing them with the most representative word, thus making sure that the *words* representing the concept are distinct. When comparing two concepts, the method must compare each word representing one concept against each word in the other concept. For example, if the i^th^ concept in ontology A, C_i^A, has n_i words and the j^th^ concept in ontology B, C_j^B, has m_j words, then there can be n_i*m_j comparisons. A score of 1 is assigned if the words are synonyms, -1 if antonyms and 0 if neither. The similarity index between C_i^A and C_j^B can be given as

$$s_1(i,j) = \frac{2\sum_{k=1}^{n_i}\sum_{l=1}^{m_j} s_{kl}}{n_i + m_j} \qquad (1)$$

The valid range for $s_1(i,j)$ is –1 to +1. The subscript on *s* indicates that this is the first of four similarity indices (terminology-based).

A few example results from the application of the above method are shown in Figure 6.



similarity index = 0.333

similarity index = 0.667

similarity index = 0.286

Figure 6: Illustration of terminology-based concept comparison

Figure 7 depicts an example user interface that shows how ontology mappings may be visualized.

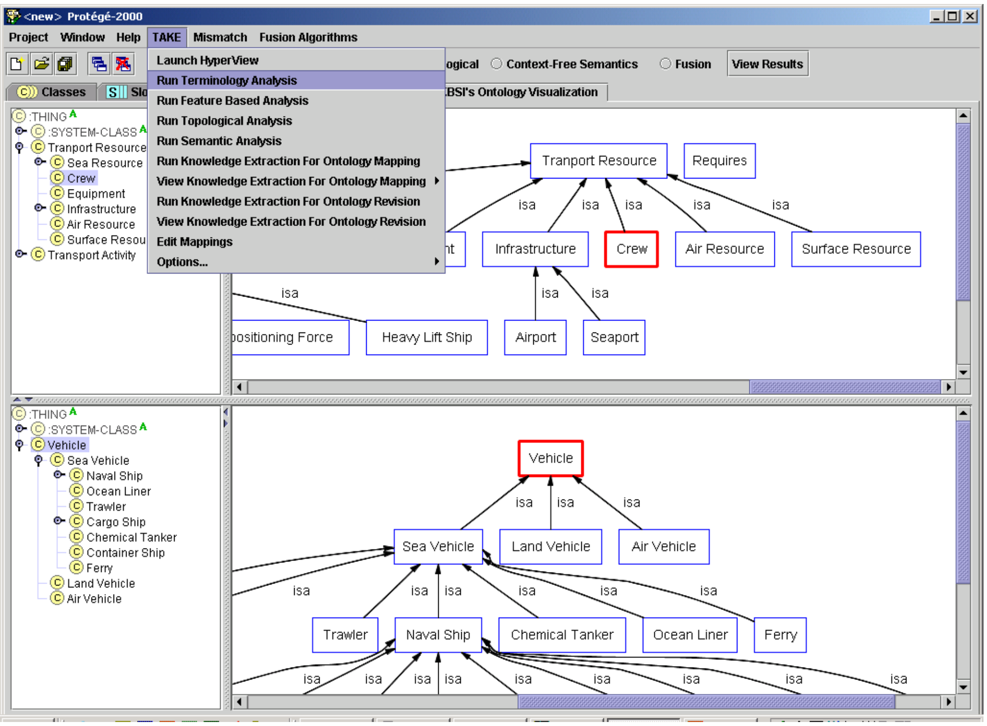


Figure 7: Visualization interface for ontology mappings

3.3 Use Ontology Mappings to Determine Information Exchange Requirements

The identification and analysis of ontology mappings provides the enabling mechanism for facilitating information sharing and integration requirements between the component simulation models/applications. Ontology mappings may be grouped into two broad categories (Klein 2002):

1. Conceptualization Mappings, and
2. Explication Mappings.

Conceptualization mappings account for differences in *granularity* and the *scope* of different ontologies (for example, a general ontology of space transportation resources vs. an ontology of resources used for range system reconfiguration).

Explication mappings are of three types: (i) Terminology, (ii) Modeling Style, and (iii) Encoding.

Terminology mappings account for the ambiguity of natural language and the naming conventions used by different organizations. The most common terminology mappings are used with synonyms (using different words to refer to the same concept) and homonyms (the difference in meaning of the same word used in different contexts).

Modeling Style mappings are used to account for modeling paradigm differences (for example, using temporal logic based on time points vs. time intervals) and modeling conventions (for example, to allow for the distinction between types and instances, allowing for the representation of relational properties such as transitivity and symmetry).

Encoding mappings are used to account for differences in formatting conventions (date formats) and differences in scale (units of measure differences, etc.).

3.4 Ontology-Driven Translation

The idea of ontology-driven translation uses *ontologies as the foundation* for translating information from one simulation application to another. Ontologies are used to formally describe the metamodels of the source and target applications. Mappings are then defined between concepts across the ontologies of the source and target applications. These mappings identify and formally define equivalences between concepts or sets of concepts in the source and target ontologies. These mappings are then used to implement translation rules for the translator. Because translation rules obtained this way are based on mappings across the metamodels of the source and target applications, they are independent of the domain of the models to be translated and are, therefore, valid for any model irrespective of its domain. Figure 8 illustrates the principle of ontology-driven translator generation. As seen in this figure, we use the notion of a "Neutral Simulation Language (NSL)" that is used as an interlingua to represent simulation information between multiple simulation tools/applications.

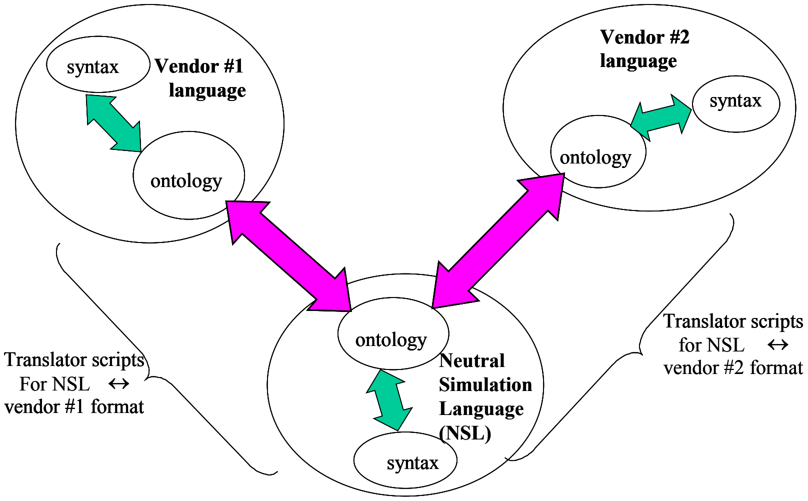
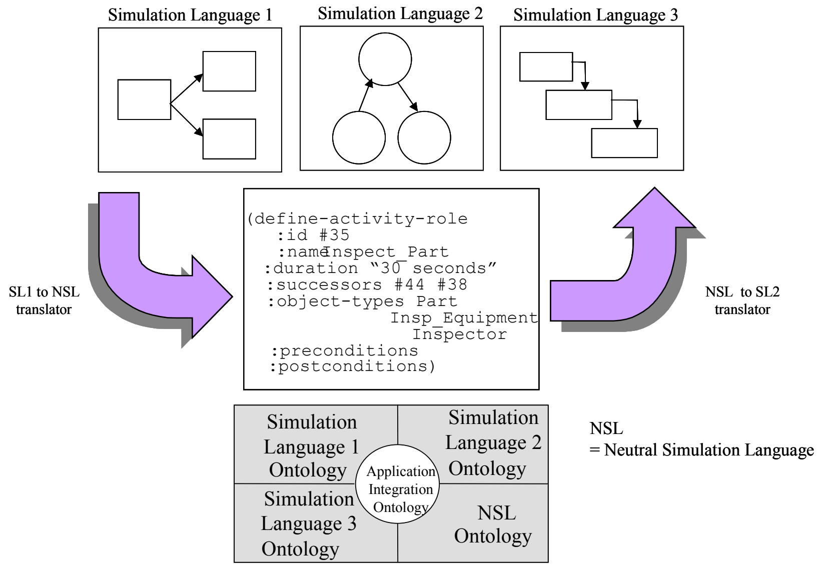

syntax
Vendor #1 language
ontology
Vendor #2 language
syntax
ontology
ontology
Neutral Simulation Language (NSL)
syntax
Translator scripts For NSL ↔ vendor #1 format
Translator scripts for NSL ↔ vendor #2 format

Figure 8: The ontology-driven translator generation concept

The idea of automatically generating translator code that can be easily modified, compiled, tested, and deployed shifts the focus of translator development on *modeling* and *mapping* the source/target specification and not on developing translator code. In that sense, it will be similar to using Computer-Assisted Software Engineering (CASE) tools to develop object-oriented software by modeling the system, say with UML, and automatically generating the target code. The main advantage of this approach is that a change in one of the vendor formats would not require programming of any sort. The model would simply be changed, and the translator code would be re-generated to reflect the changes. While, ideally, there would be no programming necessary, in reality only a part of the translator would be automatically generated and some amount of tweaking of the generated code would be necessary.

Figure 9 illustrates the above idea in greater detail, showing an example activity specification in a neutral simulation language developed by KBSI as part of ongoing R&D work (Benjamin et al. 2005).


Simulation Language 1
Simulation Language 2
Simulation Language 3
SL1 to NSL translator
NSL to SL2 translator
Simulation Language 1 Ontology
Simulation Language 2 Ontology
Application Integration Ontology
Simulation Language 3 Ontology
NSL Ontology
NSL
= Neutral Simulation Language

```
(define-activity-role
   :id #35
   :nameInspect_Part
  :duration "30 seconds"
  :successors #44 #38
  :object-types Part
               Insp_Equipment
                 Inspector
   :preconditions
   :postconditions)
```

Figure 9: Translating between simulation-based tools/applications using a neutral simulation language

4 APPLICATION EXAMPLE

This section outlines an application example that illustrates the ontology-driven simulation interoperability approach described in the previous section. We validated, through experimentation, the use of our ontology-driven approach to enable interoperability among three military combat simulation systems: (i) Enhanced Air-to-air and Air-to-Ground Linked Environment Simulation (EAAGLES) (Hodson, Gehl, and Baldwin 2006), (ii) Joint Semi Automated Forces (JSAF) <http://www.jfcom.mil/about/fact_jsaf.html>, and (iii) Radar Model Development and Analysis System (AMBER)(SAIC 2005).

EAAGLES was used to simulate the role of the training pilot with cockpit display, heads up display (HUD), and out-of-the-window (OTW) (terrain and sky view). The JSAF simulation system was used to provide a bird's eye view of the battlefield and aggressor force with autopilot capability (built-in intelligence of assigned tasks: intercept, escort, attack, etc). Aircraft information is streamed to an AMBER radar simulation model through the distributed interactive simulation/high level architecture (DIS/HLA) interface. Figure 10 illustrates the integrated architecture for this example scenario. Ontologies of EAAGLES, JSAF, and AMBER were developed using automated ontology extraction methods. Our ontology comparison methods were employed to determine the mappings among these systems. Next, we used ontology driven translation methods to facilitate the implementation of the information exchange mechanisms as part of a federated simulation experiment.

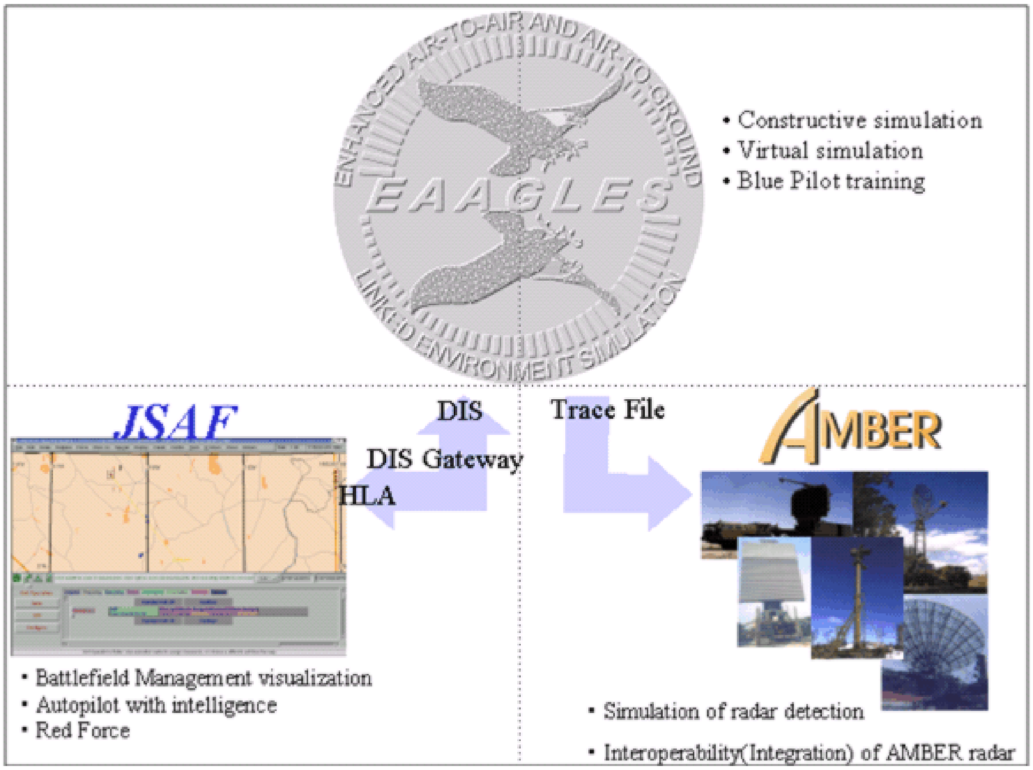

ENHANCED AIR-TO-AIR AND AIR-TO-GROUND
EAAGLES
LINKED ENVIRONMENT SIMULATION
• Constructive simulation
• Virtual simulation
• Blue Pilot training
JSAF
DIS
DIS Gateway
HLA
Trace File
AMBER
• Battlefield Management visualization
• Autopilot with intelligence
• Red Force
• Simulation of radar detection
• Interoperability(Integration) of AMBER radar

Figure 10: Simulation interoperability with EAAGLES, JSAF, and AMBER simulation systems

Communication between EAAGLES and JSAF is mainly through a DIS gateway and HLA with defined entity types. Once the entity types are defined, instances of these entities will be dynamically generated in both EAAGLES and JSAF for the federated exercise. Each side (Blue and Red) is allowed to shoot down aircraft. The entire process, from firing munitions to destroying the targets, may be seen from both (Blue and Red) sides. A trace file is used to perform verification with AMBER radar models. Information in the trace file currently contains different types of information, including position, velocity, orientation, angle of attack, side slip angle, and indicated airspeed. Whenever radar contact is established, area of interest (AOI) azimuth, AOI elevation, frequency, wavelength, pulse width, range slang, range ground, true azimuth, and relative azimuth are recorded.

5 SUMMARY AND CONCLUSION

The ideas presented in this paper are based on the hypothesis that ontology models provide a sound foundation for determining semantic interoperability and information exchange requirements for simulation based applications. After summarizing the problems that motivate the research, the paper (i) outlines the important role of ontologies in enabling simulation application mediation and interoperability, and (ii) describes a structured methodology for ontology-driven simulation interoperability. Finally, the paper outlines an example application that illustrates the pragmatic value of the ideas presented in the paper. We conclude that ontologies play a critical enabling role for the design, execution, and maintenance of simulation-based applications. We anticipate that the emerging advances in semantic technology research will lead to the increased use of ontology-based approaches to address the daunting challenges of simulation interoperability.

REFERENCES

Benjamin, P., K. Akella, K. Malek, and R. Fernandes. 2005. An Ontology-Driven Framework for Process-Oriented Applications. In *Proceedings of the 2005 Winter Simulation Conference*, eds. M. E. Kuhl, N. M. Steiger, F. B. Armstrong, and J. A. Joines, 2355–2363. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Benjamin, P., M. Graul, M. Patki, and M. Erraguntla. 2006. A Framework for Modeling and Simulation at Multiple Levels of Abstraction. In *Proceedings of SPIE, Enabling Technologies for Simulation Science X, Vol.6227*.

Boer, C. A., A. Bruin, and A.Verbraeck. 2008. Distributed simulation in industry – a survey part 3 – the HLA standard in industry. In *Proceedings of the 2008 Winter Simulation Conference*, eds. S. J. Mason, R. R. Hill, L. Monch, O. Rose, T. Jefferson, J. W. Fowler, 1094–1102. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Buss, A. and C. Blais. 2007. Composability and component-based discrete event simulation. In *Proceedings of the 2007 Winter Simulation Conference*, eds. S. G. Henderson, B. Biller, M. M. Hsieh, J. Shortle, J. D. Tew, and R. R. Barton, 694–702. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Fishwick, P. A. and J. A. Miller. 2004. Ontologies for Modeling and Simulation: Issues and Approaches. In *Proceedings of the 2004 Winter Simulation Conference*, eds. R. G. Ingalls, M. D. Rossetti, J. S. Smith, and B. A. Peters, 259–264. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Hodson, D., D. Gehl, and R. Baldwin. 2006. Building Distributed Simulations Utilizing the EAAGLES Framework. *Interservice/Industry Training, Simulation, and Education Conference (I/ITSEC)* 2006, Paper No. 2628.

Jain, S., F. Riddick, A. Craens, and D. Kibira. 2007. Distributed simulation for interoperability testing along the supply chain. In *Proceedings of the 2007 Winter Simulation Conference*, eds. S. G. Henderson, B. Biller, M. H. Hsieh, J. Shortle, J. D. Tew, and R. R. Barton, 1044–1052. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Kewley, R., J. Cook, N. Goerger, D. Henderson, and E. Teague. 2008. Federated simulations for systems of systems integration. In *Proceedings of the 2008 Winter Simulation Conference*, eds. S. J. Mason, R. R. Hill, L. Monch, O. Rose, T. Jefferson, and J. W. Fowler, 1121–1129. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Klein, M., 2002. Combining and relating ontologies: an analysis of problems and solutions. workshop on ontologies and information sharing. In *Proceedings of IJCAI*, 53–62.

Kokla, M. 2006. Guidelines on Geographic Ontology Integration. In *Proceedings of the ISPRS Technical Commission II Symposium*.

Lacy, L. and W. Gerber. 2004. Potential Modeling and Simulation Applications of the Web Ontology Language – OWL. In *Proceedings of the 2004 Winter Simulation Conference*, eds. R. G. Ingalls, M. D. Rossetti, J. S. Smith, and B. A. Peters, 265–270. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Patki, M. S., P. Benjamin, and R. Mayer. 2007. Using Ontology Comparison Methods for Simulation Composition. In *The Proceedings of the 2007 SPIE Conference,* eds. K. Schum, and D. A. Trevisani, Proc. SPIE Vol. 6564, 656408.

SAIC. 2005. Science Applications Intl. Corp. AMBER Users Manual, *Amber Radar Analysis System Users Manual*, Version 2.0, Oct. 2005.

Silver, G. A., O. A. Hassan, and J. A. Miller. 2007. From Domain Ontologies to Modeling Ontologies to Executable Simulation Models. In *Proceedings of the 2007 Winter Simulation Conference*, eds. S. G. Henderson, B. Biller, M. H. Hsieh, J. Shortle, J. D. Tew, and R. R. Barton, 1108–1117. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Yilmaz, L. 2006. On Improving Dynamic Composability via Ontology-driven Introspective Agent Architectures. In *Proceedings of the World Multi-Conference on Systemics, Cybernetics and Informatics*.

Yilmaz, L. and S. Paspuleti. 2005. Toward a Meta-Level Framework for Agent-Supported Interoperation of Defense Simulations. In *The Journal of Defense Modeling and Simulation* 2: 161–175.

AUTHOR BIOGRAPHIES

PERAKATH BENJAMIN, as Vice President for Research and Development at KBSI, directs and manages the research and development activities of the company. In that role, he manages and actively participates in the analysis, design, and development of advanced solutions for clients including the United States Department of Defense, NASA, and large corporations in the private sector. Several of the advanced technology solutions that were developed under his supervision are being applied extensively throughout industry and government. Dr. Benjamin has a Ph.D. in Industrial Engineering from Texas A&M (1991). Prior to becoming a vice president at KBSI, he was a visiting assistant professor of Industrial Engineering and lead researcher at the Knowledge Based Systems Laboratory, Texas A&M University. Dr. Benjamin has over 40 publications and book articles. He has delivered simulation and semantic technology consulting and training to multiple large organizations over the past 15 years. His email is <pbenjamin@kbsi.com>.

KUMAR V. AKELLA, a researcher at Knowledge Based Systems, Inc. (KBSI), received a Ph.D. in Mechanical Engineering from Texas A&M University. Dr. Akella's areas of expertise include simulation modeling, the design of experiments, numerical modeling, data mining, and text mining. His current work at KBSI includes simulation modeling and analysis, planning and scheduling, enterprise modeling, computer based training, and ontology development. His email is <kakella@kbsi.com>.