Service Oriented Simulation using Web Ontology

Abstract: COTS simulation packages (CSPs) have proved popular in a wider industrial setting. Reuse of simulation component models by collaborating organisations or divisions is restricted however by the same semantic issues that restrict the inter-organisation use of other software services. Semantic models, in the form of ontology, utilized by a web service based discovery and deployment architecture provides one approach to support simulation model reuse. Semantic interoperation is achieved using domain grounded simulation component ontology to identify reusable components and subsequently loaded into a CSP, modified according to the requirements of the new model, and locally or remotely executed. The work is based on a health service simulation that addresses the transportation of blood. The ontology engineering framework and discovery architecture provide a novel approach to inter-organisation simulation, uncovering domain semantics and providing a less intrusive mechanism for component reuse. The resulting web of component models and simulation execution environments present a nascent approach to simulation grids.

Keywords: COTS Simulation, Web Services, Ontology, Model Integration, Semantic Web.

1. Introduction

Modeling and experimenting with business systems is well supported by Commercial-off-the-shelf (COTS) simulation packages (CSPs) and offers an interactive and visual model development environment. Industrial simulation practitioners extensively use CSPs such as Simul8, Witness, AnyLogic, AutoMod and Arena to model their simulations. These packages allow reuse of standard simulation components like workstations, queues, conveyors, resources etc. and thereby provide the building blocks that facilitate the creation of larger models. As models grow larger and more complex the prospect of simulation model reuse is appealing as it has the potential to reduce the time and cost incurred in developing future models (benefiting from the experience embedded within existing models). In addition to reuse, the simulation owner is able to separate development and user groups, allowing models to be developed and validated by one group and then used to specify simulations by another group (Bortscheller & Saulnier, 1992). In this paper we look at the discovery and import of CSP-created models across organizational boundaries in the context of supply chains, enabling the development and deployment of model components in collaborating organisations. In its current form, the approach does not allow model information hiding between enterprises and contrasts with the *distributed simulation* approach to model reuse which allows an organisation to hide model specific information and data from the other participants. A short discussion on supply chains and the distributed simulation approach follows with additional detail provided in Section 2.

Supply Chain Management (SCM) consists of a series of tasks – such as manufacturing, transport and distribution - undertaken by organisations who aim to deliver products to their customers. Simulation of the supply chain can identify manufacturing bottlenecks; resources required for timely delivery, adequate stock levels for distribution etc. and help improve the performance of the underlying supply chain. From a simulation perspective, each organisation forms part of the supply chain and develops models in order to simulate their part of the supply chain using CSPs (Fujimoto, 2000). Assuming that the necessary individual simulation components are made available the question is how do we link them together? Distributed simulation offers one such solution. Distributed simulation can be defined as the distribution of the execution of a single run of a simulation program across multiple processors (Taylor et al., 2001). It allows each organisation to run its model in its own site (thereby encapsulating model details within the organisation itself) and participating with other sites through information exchange using distributed simulation middleware (Fujimoto, 1999). Boer et al. (2002), Mertins et al. (2000) and Mustafee & Taylor (2006) are examples of successful distributed simulation using CSPs. There is a growing body of research dedicated to creating distributed simulation with CSPs and the High Level Architecture (HLA), the IEEE 1516 standard for distributed simulation. In an attempt to unify this research COTS Simulation Package Interoperability Product Development Group (CSPI-PDG), a Simulation Interoperability Standards Organisation (SISO) standardization group began operating in October 2004 (http://www.sisostds.org/), producing reference models in 2010.

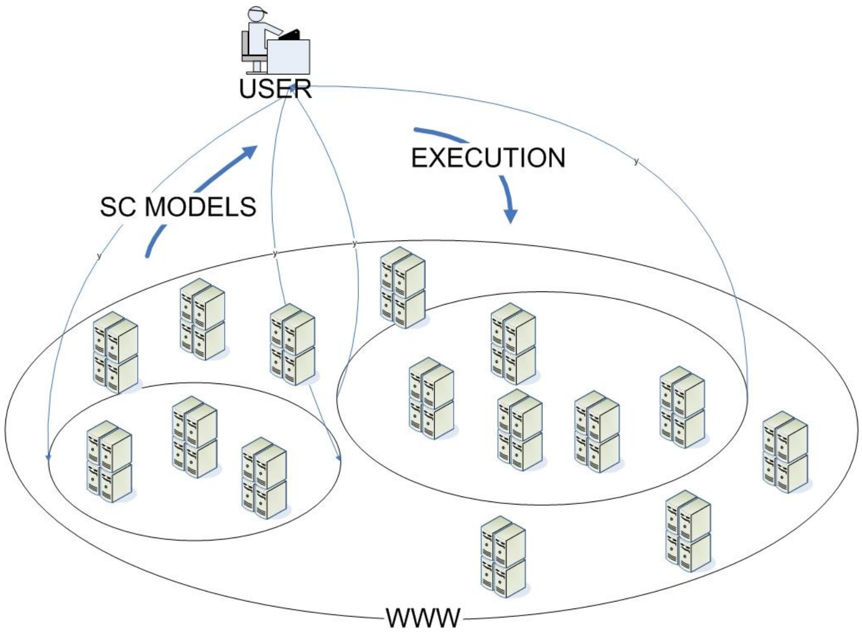


Figure 1 - Simulation Vision

Our vision is a web of simulation component (SC) models and execution environments that are accessible to the practitioner (Figure 1). New models are selected from best-practice components and deployed on CSP hosting hardware. In order to realise such a vision the user must first be able to identify suitable components. Current representations of web components are predominantly syntactic in nature lacking the fundamental semantic underpinning required to support discovery on the emerging semantic web (Bell et al., 2005). Semantic models, in the form of ontology, utilized by web service discovery and deployment architectures provide one approach to support simulation model reuse. Improved component reuse through ontological model use has already been proposed in simulation (Miller et al., 2006). Importantly however, this has focused on the simulation type and not the domain being modelled. A further concern when considering COTS Simulation packages is that intrusive activities are not possible. Packaged software of this type allows only import or export capabilities. The tools of the semantic web provide a means to construct external description of the CSP models. This external description, or ontology, can then be used to support the reuse of simulation components. Consider a scenario where a large multinational organisation uses CSPs to model many of its business activities. Two human processes (interactions) are undertaken when a simulation is required – the creation of the model from parts and its subsequent execution. In order to fully utilise the capabilities within the organisation we propose that *model parts* can be reused more effectively, better utilising the codified expertise within distinct models. In order to support the reuse, methods for describing the models and semantic discovery are proposed. The system supports the discovery of specific model components and their loading into a local or remote COTS simulation package. Semantic interoperation is achieved through the use of a simulation component ontology to identify required components at varying levels of granularity (including both abstract and specialized components). The ontology is derived from existing CSP Simulation Components and is contrasted to current simulation ontology.

Evolutionary construction of domain grounded SC ontology, a central theme of this paper, improves semantic discovery of SCs. In addition, when combined with hard simulation semantics (such as state etc.), concepts from both vocabularies provide improved matching precision. The paper is organised as follows. Section 2 presents a summary of pertinent literature including a summary of semantic web and ontologies. Section 3 describes the DESC ontology, including the process undertaken to engineer it. Section 4 covers the software tools that use the DESC ontology – the semantic search and component integration software. Evaluation of research artifacts is carried out in Section

5 and the paper concludes with a summary of the work presented and ideas for future development.

2. Related Literature

The reuse of software artifacts has been a frequent topic around software engineering; with the people, process and technology implications (and modeling support) highlighted by Jacobson et al. (1997) and more recently with standardization of components within software product lines (Bosch 2002). Designing for reuse, although a good idea is not directly relevant to this research where a heterogeneous world of existing models is already assumed – requiring a technical focus on existing artifacts. Business process model reuse (including specific model fragments) has similarities to this research however. Markovic et al. (2008) present an approach that makes use of business process ontology (containing business process language including process goals and roles). The key differentiator with regard to this research is the ontology being utilized and its development (i.e. a focus on the language of the simulation, business and technical domains as opposed to specific modeling languages). Consequently, this research focuses on the semantic web approach and its applicability to the description and discovery of simulation components. Therefore, two communities of research are relevant to the work being undertaken and presented here are: (1) Semantic web services and (2) the current approaches to description and reuse of components in simulation. Both provide an insight into the decoupling of component models from their execution environment and are used for both discovery and synthesis. Semantic search has been applied with a common reliance on knowledge – referred to as service ontology. Ontology itself is a specification of a representational vocabulary for a shared domain of discourse – with definitions of classes, relations, functions, and other objects (Gruber, 1993). It is an explicit specification of a conceptualization. The term is borrowed from philosophy, where an Ontology is a systematic account of existence (Gruber, 1993). In borrowing the term ontology and placing it into an engineering discipline, two distinct usage types emerge in the creation of these specifications: The theoretic (deductive) approach and the pragmatic (inductive approach) (Geerts & McCarthy, 1999). It is the pragmatic approach that is adopted in this paper – focusing on the engineering of knowledge from existing CSP models.

The semantic web provides structured knowledge and reasoning about a web of models and the grid promises a vision of CSPs that are able to execute newly discovered models. The semantic web (Berners-Lee et al., 2001) aims to uncover knowledge about domains so as to better support discovery, integration and understanding of resident objects. Semantic web services (SWS) refine this vision (McIllraith et al. 2001) making web services "computer-interpretable, use apparent, and agent-ready" (p.46). With a web of services comes the need to describe explicitly and in a form able to be read by computer.

Current intersections between web services and the semantic web have delivered a diverse body of research. The agent community (Gibbins et al., 2003; Martin et al.,1999; McIllraith et al., 2001) has recognized the benefit of ontology if computer-to-computer web architectures are to be achieved. Combining service and domain ontology is seen as a key to achieving service synthesis (Chen et al., 2003). Work on service ontology is currently around OWL-S and WSMO groups (with service annotation being carried out by WSDL-S and USDL groups) (Verma & Sheth, 2007). Recognizing the original work and subsequent progress by the DAML Consortium and others, attention has moved from the ontology languages to specific application to services. A discussion of semantic web services would not be complete without coverage of the OWL-S upper ontology model (WSMO is similar in a number of areas). The OWL-S high level model describes the relationship between the differing service decompositions (see Figure 2) (Ankolekar et al., 2001; Chen et al., 2003). A resource provides a service that is represented by the ServiceProfile, described by the ServiceModel and supported by the ServiceGrounding. Generally, the profile describes the service in a high level way (enough to discover the service), the model describes the detail of how it works and can be used to: (1) perform more in-depth analysis of whether the service meets a need, (2) to compose service descriptions from multiple services to perform a specific task, (3) during enactment, to co-ordinate activities from participants and (4) to monitor execution (Ankolekar et al., 2001). The service grounding details practical access and has converged with WSDL.

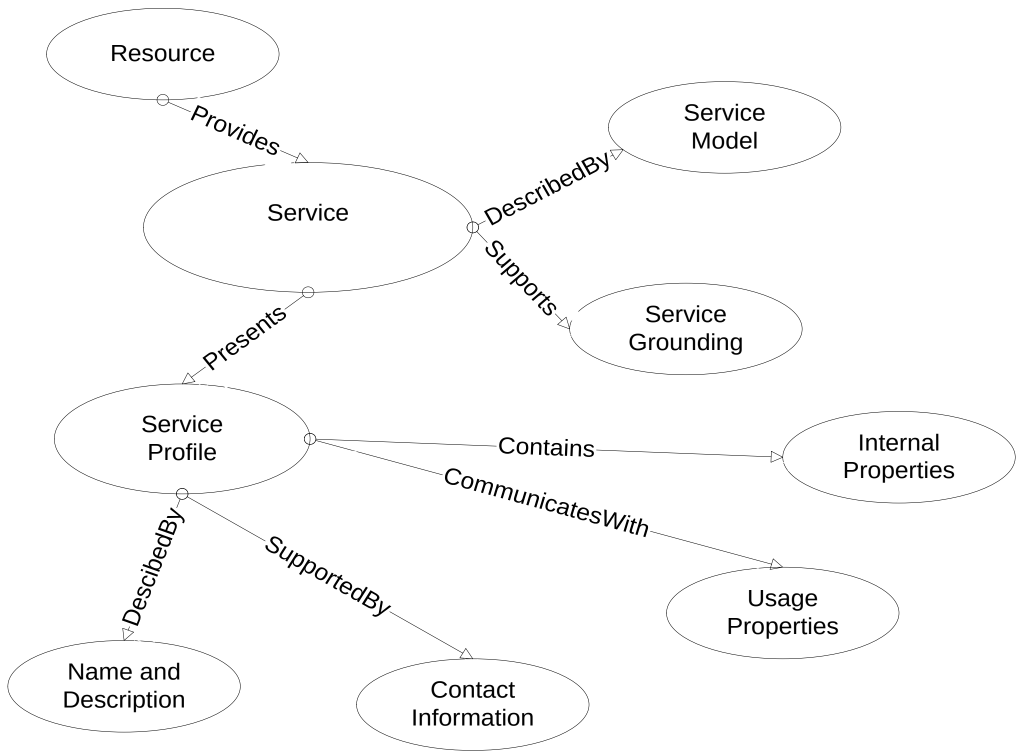


Figure 2 - OWL-S Upper Ontology

OWL-S (and WSMO) (Lara et al., 2004) provide generalized models for describing services. Others have identified the need for specialized common concepts within a web service context (Cardoso & Sheth, 2003; Dahmann & Morse, 1998; Lara et al., 2004; Paolucci et al., 2002), one example being quality of service. These concepts represent glue homogenizing a wealth of asymmetrically described web resources. New issues become pertinent in a semantic web of “great number of small ontological components consisting largely of pointers to each other” (Hendler, 2001, p.31). This semantic web service environment, with recognition of the need to combine service and domain ontology, warrants research that identifies practical approaches for practitioners to combine the service ontology with existing or new domain ontology. The foremost question in semantic service orientation is how best this should be undertaken in the context of simulation.

Transporting this vision to a simulation environment with a web of simulation components has several challenges. Combining distributed SCs models into a new model requires that they are first discovered. Consequently, explicit, computer readable knowledge is required for such search tasks. Knowledge in the form of ontologies has already been applied to simulation (Fishwick & Miller, 2004) with work by the University of Florida on simulation translation and University of Georgia on a taxonomy of simulation objects called DeMO. DeMO provides a precise description of simulation models with hard semantics. In order to realize a vision for SCs similar to that of SWS requires that the domain being simulated is represented explicitly (an OWL ontology – W3C, 2005). The DeMO ontology (Fishwick & Miller, 2004) is an upper ontology that details events, activities and processes. Hard semantics work perfectly if all stakeholders adopt the single model. If this is not the case, and with only the CSP SCs, a transformation directly to such a model will likely miss tacit domain concepts that may help any subsequent SC search activity.

The eXtensible Modeling and Simulation Framework (XMSF) is defined as a set of composable standards, profiles and recommended practices for web-based modeling and simulation. XMSF prescribes the use of ontologies for the definition, approval and interoperability of complimentary taxonomies that may be applied across multiple simulation domains (Bhatt et al, 2004). In military modeling and simulation, the study of ontology is recognized as important in developing techniques that would allow semantic interoperability between simulation systems and to this effect ontology of C2IEDM

(Command and Control Information Exchange Data Model) has been created to further studies on enabling interchange of data between two or more systems (Tolk and Turnitsa, 2004). Work is also underway that creates an ontology for physics which would represent physics-based model semantics in modeling and simulation. The intension is to capture the concepts of physical theories in a formal language so as to support various forms of automated processing that are currently not supported (Collins, 2004). Ontology for the representation of synthetic environment have also been proposed (Bhatt et al., 2004) - sedOnto (Synthetic Environment Data Representation Ontology). Finally, ongoing work is looking into establishing an ontology for the Battle Management Language (BML), an unambiguous language to command and control forces and equipment (Tolk & Blais, 2005).

Current approaches to distributed simulation rely on tightly coupled SCs. The proposed looser approach to reuse warrants a basic understanding of current distributed simulation approaches. Distributed simulation has been defined as the distribution of the execution of a single run of a simulation program across multiple processors (Fujimoto, 1999, 2003). In 2000, the IEEE published a standard approach to distributed simulation called the IEEE 1516 standard *The High Level Architecture* (HLA) (IEEE, 2000, 2003) (updated in 2006 – see Figure 3). In the HLA, a distributed simulation is called a federation, and each individual simulator is referred to as a federate. HLA Runtime Infrastructure (RTI) software provides services to federates in a manner which is comparable to the way a distributed operating system provides services to applications (US DOD, 1999). These RTI services enable federates to communicate with one another, as well as to control and manage the simulation. The HLA is composed of four parts: a set of compliance rules (IEEE 1516, 2000), the Object Model Template (OMT) (IEEE 1516.2, 2000), the Federate Interface Specification (FIS) (IEEE 1516.1, 2000), and the Federate Development Process (FEDEP) (IEEE 1516.3, 2003).

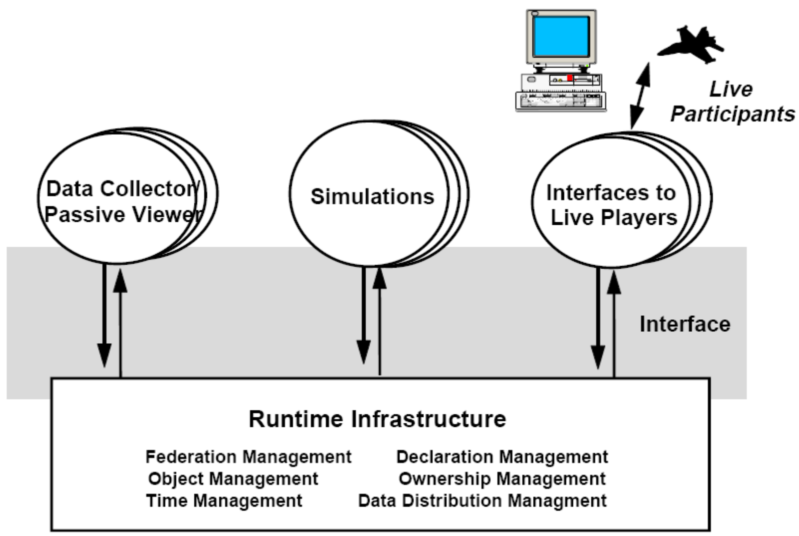


Figure 3 - Functional view of an HLA federation (from Dahmann & Morse, 1998)

The rules are a set of ten basic conventions that define the responsibilities of federates and their relationship with the RTI. Of these, five rules relate to the federation and five to the federate. These rules relate to HLA OMT representation of federation and federate objects, transfer of object ownership among federates, federates' management of local time, among others.

The FIS is an application interface standard for HLA distributed simulation middleware which defines how federates interact within the federation, and is implemented by an RTI (i.e. federates communicate with one another via an RTI). FIS organises the communication between federates and the RTI into six different service groups (US DOD, 1999).

- Federation Management: RTI Calls for creation and deletion of federation; joining and resigning of federates from the federation; and creation and realization of synchronization points.
- Declaration Management: Calls pertaining to publication and subscription of interaction class (interaction classes describe events and comprise of parameters) and object class (object classes describe persistent objects and comprise of attributes).
- Object Management: Calls that relate to sending and receiving interactions, updating object class attributes. Also services that relate to instance registration and instance updates on the object producers' side, and instance discovery and instance reflection on the object consumers' side.
- Ownership management: RTI calls for divesting or acquiring ownership of object and / or individual object attributes. It also supports ownership queries.
- Time Management: RTI calls required to implement time management mechanisms and to advance the federate simulation clock.
- Data Distribution Management: RTI calls for advance RTI routing of data.

A distributed simulation approach to model reusability faces a number of challenges. Firstly, a lack of widespread demand for distributed simulation in industry has meant that the CSP vendors have not currently incorporated distributed simulation support into their products. Consequently, the organisations that want to use this approach do not have readymade solutions. Secondly, research projects that aim to create CSP based distributed simulation do not have access to product source code and are limited to using the functionality offered by the specific vendor. Thirdly, execution time of a distributed simulation can sometimes be slower than standalone simulation, typically because of overly granular parallelization and associated network overheads (Gan et al., 2005). In order to progress, these issues have to be resolved before industry can fully benefit from the application of CSP based distributed simulation. In the meantime it is worth investigating alternative approaches that enable tactical supply chain simulation across organisational boundaries. Our discovery and import approach to model reuse, in the context of CSPs, offers one such alternative to existing distributed simulation. By discovery we mean that individual simulation models (or model parts), which are created by organisations to model their activity in the supply chain, are identified from within an inter-organisational repository (or Web) of models. The selected models are then loaded into a CSP, modified according to the requirements of the new model and executed. We believe that our approach at enabling CSP based supply chain simulation has a lighter touch with fewer technical barriers. It also requires minimal CSP vendor intervention when compared to the distributed approach.

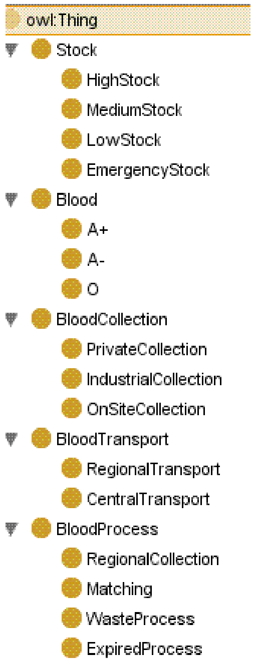
3. Simulation Component Ontology

3.1 Requirement for Semantic Search

The globalisation of many organisations and industries often result in a fragmentation of the heterogeneous knowledge produced by resident domain experts. In order to synthesize the most appropriate knowledge in a model, the best available model parts must first be found. Typically, these will come from a number of domain experts or ad-hoc selection from a local model repository. Syntactic or taxonomic approaches, e.g. list of concepts or components, limit the precision at which SCs can be related to the domain (e.g. relating to physical entities or recognizable processes), due in part to a tendency to generalize. For example, components that may not fit neatly into their prescribed category or overly general synonyms may be used to describe components.

3.2 DESC Ontology

The Discrete Event Simulation Component (DESC) ontology resulted from two distinct research activities: (1) The transformation of CSP models into OWL and Resource Description Framework (RDF) ontology files and (2) semantic search scenarios being carried out against the OWL files. Snapshots of DeMO and DESC ontologies are presented in figures 4 and 5. The differences are apparent with DeMO focusing on the component properties and DESC on the component in relation to the domain (including the components technical and contextual specification). Links between the two models are achieved through referencing the DeMO Model, Component, Concept or Mechanism from a DESC:SimulationConcept when it relates to a specific model component. Practically, the DeMO ontology is imported into Protégé in order to use its classes as properties of the DESC ontology (for example, when describing a business concept that is a specific *state* or *activity* in the simulation) and also imported into the component descriptions themselves. The DESC components are described using Turtle as RDF triples (see http://www.w3.org/TeamSubmission/turtle/) as this provides an end-user means to add and amend description. For example, the vocabulary for component description is detailed in the DESC OWL ontology with each simulation component (e.g. *exampleComponent1* in Figure 4) described using this (and DeMO) language in RDF. RDF triples can be seen in the description of two example components 1 and 2 – one focusing on the business domain and the other on the technical. In this example, two triples state that, *exampleComponent1* uses blood of type A+.




```
@prefix DESC:     <http://site/ontology/DESC_Blood.owl#> .
@prefix DeMO:     <http://site/ontology/DeMO.owl#> .

:exampleComponent1
DESC:SimulationConcept DeMO:p1:EventOrientedModel .
DESC:SimulationConcept DESC:NHS .
DeMO:ModelConcept "UseBlood" .
DESC:Blood "A+" ;
.
.
:exampleComponent2
DESC:ComponentName "TransportBlood" .
DESC:RegionalTransport "Areal" .
DESC:ComponentLocation "URI of Model fragment" .
DESC:ComponentSource "Simul8" .
DESC:ComponentVersion "1.5" .
DESC:ComponentSourceModel "Model1:v1.4:12/01/2008" .
DESC:ComponentContext DESC:BloodTransport ;
```

Figure 4 - DESC-Blood Service Domain and DESC-Blood Service Component Ontology

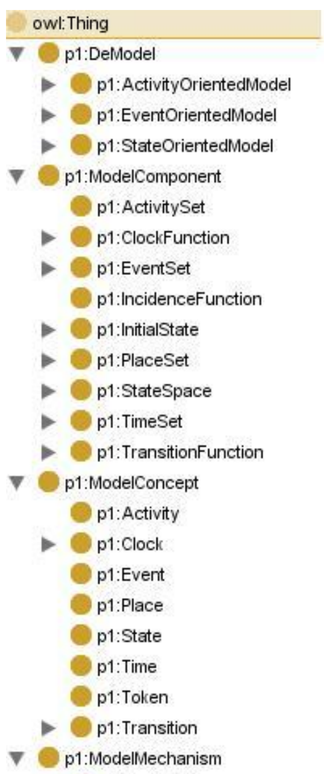


Figure 5 - DeMO Ontology Structure

The ontology was created using the Protégé tool from Stamford University (with OWL plugins) (http://protege.stanford.edu/). A decision was made to ground the ontology in existing SCs as opposed to using particular service ontology such as OWL-S or WSMO. Approaching the modeling in this way allows evolution and integration of underlying concepts described in a number of existing models. It should be noted that the SCs are modeled within the DESC ontology and reference external ontology (e.g. DeMO for simulation specifics and others). The DESC ontology is focused on the domain, both business and technology specific domain concepts. DeMO and DESC provide a robust means to describe *simulation components* using simulation, domain and technology specific language in a relatively unconstrained manner. This relative freedom in description (subclassing DESC:SimulationConcept in the most part) enables greater flexibility when searching.

3.3 Ontology Engineering

A number of activities were carried out in order to transform CSP models into ontological form – namely OWL and RDF files. The process included the decoupling of the SCs from the model by placing discrete component models into a web based component library (URI accessible). The activities carried out, in framework form, are detailed in Table 1. The framework evolved as each CSP model was deconstructed and transformed into ontology classes (including relations to dependent or related classes). Realization of the need for a DESC ontology resulted from this process – which included the adoption of DeMO for hard component semantics.

Activities	Description	Impact

Component Extraction	Specific components are extracted to form distinct models. These are stored in the DESC library (a standard web server).	CSP models SC Models
Component Typing	A new class is added to the OWL ontology to represent the SC. Similar classes are grouped under a type.	OWL Classes
Component Dependency Models	Extended DeMO properties are used to define dependencies between services. E.g. StateDependency. Reference DeMO concepts when describing business properties (e.g. Matching of blood has a DeMO state property defining the result). New classes and properties are created for previously implied activities etc. (e.g. BloodTransport is created from an analysis of the various transportation activities). The RDF description is driven in part from the component library – using the derived OWL concepts in DeMO and DESC.	OWL Properties New OWL Classes and properties implied from the model (both process and physical entity) New RDF component ontology
Ontology Testing	The finalized ontology is loaded into the SEDI4G server and several search tasks are undertaken.	DESC-BloodService OWL & RDF Ontology files

Table 1 - Process for deriving semantic content from CSP Models

The ontology engineering process resulted in DESC-BloodService OWL and RDF files (seen in Figure 2). Searching the ontology resulted in more components being returned as concept inferencing was able to traverse the concept tree and return additional suitable candidates (e.g. various blood transportation alternatives). The process undertaken to engineer the domain simulation ontology provides the basis for subsequent modelers to reference and extend the domain ontology; thus achieving richer search results and evolving large component ontology. The ontology engineering process systematically analyses the CSP model, of which Figure 6 is an example.

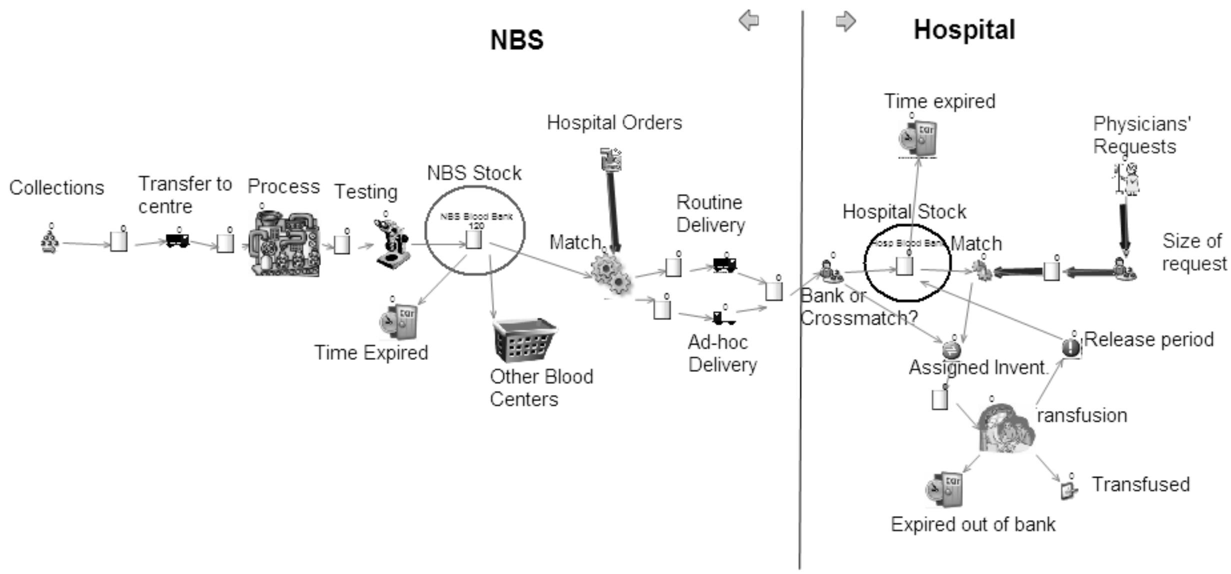


Figure 6 –Simul8 Health Service Model

4. Discovery and Import of Simulation Components

Our *discovery and import* approach aimed at CSP model reuse enables us to (1) semantically search for the desired simulation models and (2) parse and import the identified models into a simulation package. For our demo application we have used the CSP Simul8. Simul8 enables users to rapidly construct accurate, flexible and robust simulations using an easy-to-use visual modeling interface (Bell et al., 2007). However, our discovery and import architecture has the potential to support any CSP that allows an external program to perform basic operations such as opening the CSP and loading a model through its Component Object Model (COM) or XML import interface. COM is a Microsoft technology that allows different software components to communicate with each other by means of interfaces (Gray et al., 1998). The discovery component of our architecture (described in section 4.1) can be used with very little change to support other

CSPs. The parse and import component, however, would require implementation of a CSP specific parser (described in section 4.2).

4.1 Design of Component Discovery System

The component discovery system is an extension of the SEDI4G architecture (Bell & Ludwig, 2005). Extending the application to support SC descriptions as well as grid services required only minor configuration changes to support the new OWL DESC ontology. The semantic discovery system shown is Figure 7 comprises a set of web services.

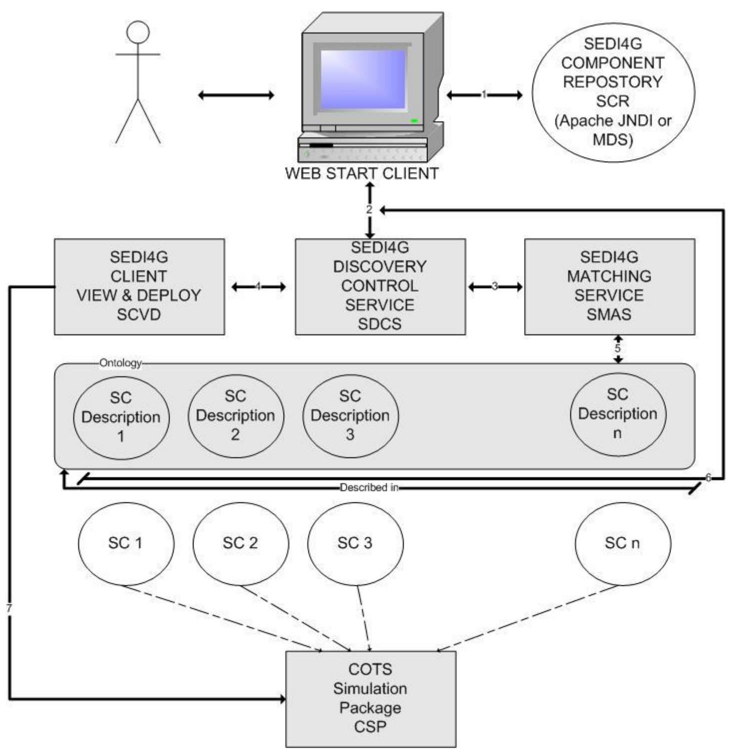


Figure 7 – Component Discovery Architecture

The discovery process begins by identifying the web services and ontology required to carry out semantic search. The choices are directed by the ontology size and service placement on the network (represented by the grey flexible services and data in Figure 7).

Thus, Step 1 involves the selection of which discovery control service (SDCS), knowledge base and matching service best fit the user requirement – specified as text strings. This information is sent to SDCS together with the search parameters (2). SDCS then calls the knowledge base (KB) using the matching service (SMAS) (itself based on OWLJessKB (http://edge.cs.drexel.edu/assemblies/software/owljesskb/)) (3) that in turn loads the KB and rules (5). The matching is carried out and returned to SDCS for use in one of the client components (4). The SDCS service can optionally provide the resource properties, the dynamic state of each service, alongside the service choices (6). Finally the returned components are displayed in a web start client (SCSV holding the component options on the server side) allowing selected components to be deployed into the CSP. The deployment is simple in nature, loading server side XML into the CSP. A more robust solution would provide transformation capabilities similar to those undertaken at the University of Florida (Fishwick & Miller, 2004).

The matching algorithm is semantic and uses an ontology and a reasoning engine. The assumption in this paper is that an ontology is a catalogue of the types of "things"; derived from existing simulation models and including simulation, domain and technology specific elements. Types in the ontology represent the predicates, word meanings, or concept and relation types of the language when used to discuss topics in the domain (Bell & Ludwig, 2005) – in this paper these are SCs.

To summarize, the matching algorithm comprises two steps; the initialization of the knowledge base and the search. During the initialization phase the ontology is loaded, transforming ontological classes into facts that have rules applied using the Rete algorithm (Forgy, 1982). During the search inferences are made from the facts (using Jess queries) identifying semantically matched SCs. For example, when searching for a component to simulate a blood collection – several alternatives are returned that model different processes and relate to different locations etc.

4.2 Design and Operation of the CSP Model Parser and Importer

The discovery architecture detailed in the previous section is used by the CSP Model Parser and Importer (CMPI) software to conduct a semantic search for existing models. The search is conducted by calling a web service defined in the component discovery architecture, which takes a search string as a parameter and returns an enumeration of uniquely identified name (URN) and corresponding unique resource identifier (URI) for each model returned by the matching algorithm. CMPI then provides the user an option to (1) download the models into the local system for inspection or (2) import it directly into the new model being built through reuse of the discovered components. In case the user chooses option (1) the model can be loaded into the local system. The file downloaded is an XML representation of the Simul8 model which was discovered. If the user chooses option (2) the URN is passed as a parameter to yet another web service which returns the XML representation of the model as a SOAP attachment. The nature of this web service is synchronous and this allows the CMPI to block further execution of the code until the XML file has been received.

The merging of the existing model parts (reusing the discovered model components) into new models requires a CSP specific parsing operation. For example, a model fragment exported using Simul8 will need to be transformed in order import into another package. Typically, both of the component models will be XML based. In this research we employ a crude parsing mechanism that transforms and combines a small sample of model formats. The result being a newly generated XML file is loaded into the CSP and the user is presented with a new model to work on. It should be noted that the text parsing mechanism is heavily dependent on Simul8 specific knowledge and has yet to be fully perfected. It is envisaged that (in future work) knowledge about packages could be described using ontology in order to automate this transformation process – seamlessly integrating model parts from a number of source systems.

Two alternative CMPI implementations were carried out: (1) Servlet based and (2) COM based. The Servlet approach provides remote access to the CSP – deploying an XML file for upload into Simul8 on a remote machine. The COM version of CMP software is written in Java and it uses the Simul8 COM interface to interact with a local Simul8 instantiation using Java Native Technology (Sun, 2003). CMPI invokes web service calls to communicate with the component discovery system. It also includes a CSP specific parser component which, as has been discussed in the previous paragraph, can be considered optional. The architecture and dependencies of CMPI is shown in Figure 8.

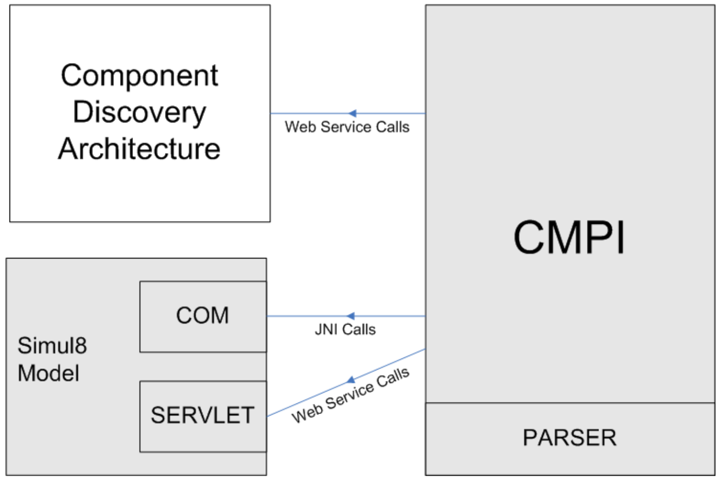


Figure 8 - Architecture of dependencies of CMPI

Alternative approaches to loading the discovered SC model provide a flexible execution environment, supporting: (a) search, deployment and execution locally and (b) search locally and discovery/execution remotely. The remote approach provides opportunities for Grid-enabling the simulation environment. The grid aspects of the execution are beyond the scope of this paper and form part of ongoing research at Brunel.

5. Evaluation of research artifacts

The ontology files that resulted from the interpretation of a health related simulation model were deployed into the semantic discovery architecture. All OWL files were placed on the same web server. The DESC ontology now references one or more domain ontology. The impact of this on the design is that the initial phase of ontology selection is more complex, with a larger choice of varying content. With more OWL files the search has three options: (1) Load all ontologies and search them, (2) Load only those ontology referenced by the component ontology or (3) Load the ontology references by a particular property of the a service ontology (e.g. a particular model type specified using DeMO ontology). Re-tests using SEDI4G and a number of search scenarios identified performance decreases when moving from (1) to (2) or (2) to (3). The heuristics for filtering the search space are not part of this research, but may provide a useful direction for further research. Performance is line with previous (non simulation) research into service discovery (Bell & Ludwig, 2005). Topology decisions apply when deploying component models, as well discovery services, as opportunities for network optimization exist through the co-existence of service and ontology. The combined effectiveness of distributed semantic search and ontology engineering presented in this design needs discussion in terms of literature derived requirements (Paolucci et al., 2002; Trastour et al., 2001).

Requirement	Fulfillment
High degree of flexibility and expressiveness	Expressiveness is achieved through the combination of three distinct ontology types, covering: the overall simulation, simulation components and the domain. Describing SCs precisely, in relation to other SCs and in relation to the domain improves expressiveness. Existing approaches are either precise (DeMO) or as is the case with grid discovery – taxonomy focused. A high degree of flexibility is achieved through distributing the discovery components, the SC models and the execution environments across the network.
Expression of semi-structured data	The ontological approach supports semi structured data in that partial description of

	component models are allowed. The interpretation process supports evolution in structure as new relations are exposed – adding structure to existing objects. Support for this drove in part the design of the RDF based component ontology – allowing simplified end-use description.
Support for types and subsumption	Domain analysis clearly supports typing – particular the sub-classification of specializations. The result is a rich, deep branch that is able to benefit for subsumption approaches to matching.
Ability to express constraints	Limited work on constraints is included in the framework. The 'object' approach to ontology engineering allows constraints to be represented as ontological classes – and these are specialised as analysis progresses, e.g. HighSpeed Transport contains an implied constraint.
False Positives and Negatives should be minimized	Complex simulation models do not produce a small, compact language to describe SCs (unlike high performance computing where CPU Usage is well understood). Precise description of SCs in relation to the domain and each other support the need for precision. Observing the design artifacts and source models directed the focus on ontology engineering (proving more effective than a singular for on search algorithm optimization). The precision in component selection is increased as more terms are included in the search string, although reducing terms further supports SC browsing.
The algorithm should encourage	Honesty is out of the scope of this investigation.

honesty	The diversity of description is supported through clear transformation and grounding from source services. Subsequent description of services using untruths is not restricted. The approach does however support systematic analysis of the represented domain (in model form).
Matching should be efficient	The distributed nature of the SEDI4G system is more efficient than centralised discovery architecture. Performance supports efficiency claims; especially through the use of heterogeneous network of ontology and search components. The ability to deploy models across a number of execution environments provides run-time efficiency.

Table 2 - SC Discovery Requirements

Comparing the system (and ontology engineering frameworks) to current enterprise discovery (e.g. UDDI) the approach presented in this paper meets more of the requirements for matchmaking. In contrast, a comparison to HLA is more problematic with the proposed approach attempting to radically shift ideas of integration and reuse to support on-the-fly reuse. None the less, it is worth emphasizing that a semantic web service based approach to simulation model reuse has proved to be both less invasive and relatively easy to architect (as loosely couple infrastructural add-ons to existing simulation tools).

6. Conclusion

The paper presents a novel approach to CSP model reuse using a simulation component ontology and semantic search architecture – decoupling both model parts and execution platforms. Approaching the modeling of simulation components directs focus on the domain being modeled – allowing for the explicit description of simulation components in domain language. In relating each component to a typed collection and each other enables the search process to better identify likely semantic matches when users search for existing models to reuse. A COTS simulation package (Simul8) was used with models being transformed into OWL ontologies and then used by a web service based semantic search and component deployment architecture. The research has demonstrated: (1) a new, lighter approach to CSP model reuse and (2) the benefits of semantic search to this field of research.

6. References

Ankolekar, A., Burstein, M., Hobbs, J.R., Lassila, O., Martin, D., McDermott, D., McIlraith, S.A., Narayanan, S., Paolucci, M., Payne, T. & Sycara, K. 2001, "DAML-S: Semantic Markup For Web Services", *International Semantic Web Working Symposium (SWWS),* Springer-Verlag, pp. 348.

Bell, D., de Cesare, S. & Lycett, M. 2005, "Semantic Transformation of Web Services", *OnTheMove 2005 (SWWS 2005 Workshop),* Springer LNCS 3762, pp. 856.

Bell, D., de Cesare, S., Lycett, M., Mustafee, N. & Taylor, S.J.E. 2007, "Semantic Web Service Architecture for Simulation Model Reuse", *Proceedings of the 11th IEEE International Symposium on Distributed Simulation and Real-Time Applications,* pp. 129-136.

Bell, D. & Ludwig, S.A. 2005, "Grid Service Discovery in the Financial Markets Sector", *Journal of Computing and Information Technology,* vol. 13, no. 4, pp. 265-170.

Berners-Lee, T., Hendler, J. & Lassila, O. 2001, "The Semantic Web", *Scientific American,* vol. 284, no. 5, pp. 34-43.

Bhatt, M., Rahayu, W. & Sterling, G. 2004,"sedOnto: A Web Enabled Ontology for Synthetic Environment Representation Based on the SEDRIS Specification", *Fall Simulation Interoperability Workshop*.

Boer, C.A., Verbraeck, A. & Veeke, H.P.M. 2002, "Distributed simulation of complex systems: Application in container handling", *European Simulation Interoperability Workshop*.

Boon Ping Gan, Yoke, M., Low, H., Xiaoguang Wang & Turner, S.J. 2005, "Using manufacturing process flow for time synchronization in HLA-based simulation", *Ninth IEEE International Symposium on Distributed Simulation and Real-Time Applications,* pp. 148.

Bortscheller, B.J. & Saulnier, E.T. 1992, "Model reusability in a graphical simulation package", *WSC '92: Proceedings of the 24th conference on Winter simulation*ACM Press, pp. 764.

Cardoso, J. & Sheth, A. 2003, "Semantic e-workflow composition", *Journal of Intelligent Information Systems,* vol. 21, no. 3, pp. 191-225.

Chen, L., Shadbolt, N.R., Goble, C., Tao, F., Cox, S.J., Puleston, C. & Smart, P. 2003, "Towards a Knowledge-based Approach to Semantic Service Composition", *Second International Semantic Web Conference (ISWC2003)*.

Collins, J.B. 2004,"Standardizing an Ontology of Physics for Modeling and Simulation", *Fall Simulation Interoperability Workshop*.

Concannon, K.H., Hunter, K.I. & Tremble, J.M. 2003, "Dynamic scheduling II: SIMUL8-planner simulation-based planning and scheduling", *WSC '03: Proceedings of the 35th conference on Winter simulation*Winter Simulation Conference, pp. 1488.

Curbera, F., Duftler, M., Khalaf, R., Nagy, W., Mukhi, N. & Weerawarana, S. 2002, "Unraveling the Web services Web - An introduction to SOAP, WSDL, and UDDI", *IEEE Internet Computing,* vol. 6, no. 2, pp. 86-93.

Dahmann, J.S. & Morse, K.L. 1998, "High level architecture for simulation: An Update", *Proceedings of the 2nd International Workshop on Distributed Interactive Simulation and Real-Time Applications,* pp. 32-40.

Fishwick, P.A. & Miller, J.A. 2004, "Ontologies for modeling and simulation: Issues and approaches", *Proceedings of 2004 Winter Simulation Conference*, pp. 259-264.

Forgy, C.L. 1982, "Rete: A Fast Algorithm for the Many Pattern/Many Object Pattern Match Problems", *Artificial Intelligence,* vol. 19, pp. 17-37.

Fujimoto, R.M. 2003, "Distributed simulation systems", *Proceedings of the 2003 Winter Simulation Conference,* pp. 124.

Fujimoto, R.M. 2000, *Parallel and Distributed Simulation Systems*, John Wiley & Sons.

Gan, B.P., Liu, L., Jain, S., Turner, S.J., Cai, W. & Hsu, W. 2000, "Manufacturing supply chain management: distributed supply chain simulation across enterprise boundaries", *WSC '00: Proceedings of the 32nd conference on Winter simulation*Society for Computer Simulation International, pp. 1245.

Geerts, G. & McCarthy, W.E. 1999, "An accounting object infrastructure for knowledge-based enterprise models", *IEEE Intelligent Systems & Their Applications,* vol. 7.

Gibbins, N., Harris, S. & Shadbolt, N. 2003, "Agent-based semantic web services" in *Proceedings of the 12th international conference on World Wide Web* ACM Press, Budapest, Hungary, pp. 710-717.

Gray, D.N., Hotchkiss, J., LaForge, S., Shalit, A. & Weinberg, T. 1998, "Modern languages and Microsoft's component object model", *Communications of the ACM,* vol. 41, no. 5, pp. 55-65.

Gruber, T.R. 1993, "A translation approach to portable ontology specifications", *Knowledge Acquisition,* vol. 5, no. 2, pp. 199-220.

Hendler, J. 2001, "Agents and the Semantic Web", *Intelligent Systems, IEEE [see also IEEE Intelligent Systems and Their Applications]*, vol. 16, no. 2TY - JOUR, pp. 30-37.

IEEE 1516, 2000, "IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA)," New York, NY: Institute of Electrical and Electronics Engineers.

IEEE 1516.0, 2000, "IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) - Rules," New York, NY: Institute of Electrical and Electronics Engineers.

IEEE 1516.2, 2000, "IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) - Object Model Template (OMT) Specification," New York, NY: Institute of Electrical and Electronics Engineers.

IEEE 1516.1, 2000, "IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) - Federate Interface Specification," New York, NY: Institute of Electrical and Electronics Engineers.

IEEE 1516.3, 2003, "IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) - Federate Development Process (FEDEP)," New York, NY: Institute of Electrical and Electronics Engineers.

Jacobson, I., Griss, M. & Jonsson, P. 1997, *Software reuse: architecture, process and organization for business success,* ACM Press/Addison-Wesley Publishing Co. New York, NY, USA.

Lara, R., Roman, D., Polleres, A. & Fensel, D. 2004, "A Conceptual Comparison of WSMO and OWL-S", *Web Services: European Conference, ECOWS 2004*, pp. 254.

Markovic, I., Pereira, A.C. & Stojanovic, N. 2008, "A framework for querying in business process modelling", *In Proceedings of the Multikonferenz Wirtschaftsinformatik (MKWI), München, Germany*, .

Martin, D., Cheyer, A.J. & Moran, D.B. 1999, "The Open Agent Architecture: A Framework for Building distribted Software Systems", *Applied Artificial Intelligence,* vol. 13, no. 1-2, pp. 91-128.

McIlraith, S.A., Son, T.C. & Zeng, H.L. 2001, "Semantic Web services", *IEEE Intelligent Systems,* vol. 16, no. 2, pp. 46-53.

Mertins, K., Rabe, M. & Jaekel, F. 2000, "Neutral template libraries for efficient distributed simulation within a manufacturing system engineering platform", *WSC '00: Proceedings of the 32nd conference on Winter simulation*Society for Computer Simulation International, pp. 1549.

Miller, J.A., Fishwick, P.A., Baramidze, G. & Sheth, A.P. 2006, *Ontologies for Modeling and Simulation: An Extensible Framework*, Technical Report #UGA-CS-LSDIS-TR-06-011, Department of Computer Science, University of Georgia, Athens, Georgia, pp. 1-47.

Mustafee, N. & Taylor, S.J.E. 2006, "Investigating Distributed Simulation with COTS Simulation Packages: Experiences with Simul8 and the HLA", *2006 Operational Research Society Simulation Workshop (SW06)*, pp. 33.

Paolucci, M., Kawamura, T., Payne, T.R. & Sycara, K. 2002, "Semantic matching of Web services capabilities" in *Semantic Web - Iswc 2002* SPRINGER-VERLAG BERLIN, Berlin, pp. 333-347.

Sun Microsystems JNI, Online: http://java.sun.com/j2se/1.4.2/docs/guide/jni/, Sun Microsystems Limited Java Native Interface, 2003.

Taylor, S.J.E., Sudra, R., Janahan, T., Tan, G. & Ladbrook, J. 2001, "Towards COTS distributed simulation using GRIDS", *WSC '01: Proceedings of the 33nd conference on Winter simulation*, IEEE Computer Society, pp. 1372.

Tolk, A. & Blais, C. 2005,"Taxonomies, Ontologies, and Battle Management Languages – Recommendations for the Coalition BML Study Group, Spring Simulation Interoperability Workshop", *Spring Simulation Interoperability Workshop (SIW).*

Tolk, A. & Turnitsa, C. 2005, "Ontology of the C2IEDM - Further studies to enable Semantic Interoperability", *Fall Simulation Interoperability Workshop (SIW).*

Tosic, V., Esfandiari, B., Pagurek, B. & Patel, K. 2002, "On requirements for ontologies in management of Web Services" in *Web Services, E-Business, and the Semantic Web* SPRINGER-VERLAG BERLIN, Berlin, pp. 237-247.

Trastour, D., Bartolini, C. & Gonzalez-Castillo, J. 2001 , *A Semantic Web Approach to Service Description for Matchmaking of Services.* [Homepage of Hewlett-Packard Laboratories], [Online]. Available: http://www.hpl.hp.com/techreports/2001/HPL-2001-183.pdf [2006, .

US Department of Defense Modeling and Simulation Office, 1999, "High Level Architecture Run-Time Infrastructure RTI 1.3-Next Generation Programmer's Guide (version 5)," US DoD M&S Office.

Verma, K. & Sheth, A. 2007, "Semantically Annotating a Web Service", *IEEE Internet Computing*, vol. 11, pp. 83-85.

W3C, 2005, "Web Ontology Language," Online: http://www.w3.org/2001/sw/WebOnt/.