Proceedings of the 2004 Winter Simulation Conference
R .G. Ingalls, M. D. Rossetti, J. S. Smith, and B. A. Peters, eds

EXPLORING AGENT-SUPPORTED SIMULATION BROKERING ON THE SEMANTIC WEB: FOUNDATIONS FOR A DYNAMIC COMPOSABILITY APPROACH

Levent Yilmaz

M&SNet: Auburn Modeling & Simulation Laboratory
Department of Computer Science and Engineering
Auburn University
Auburn, AL 36849, U.S.A.

Tuncer I. Ören

M&SNet: Ottawa Center of the MISS
School of Information Technology and Engineering
(SITE), University of Ottawa
Ottawa, ON K1N 6N5, CANADA

ABSTRACT

Federated simulations address the need for interoperability, as well as the improvement of reuse and composability. The focal goal in a federated simulation is to facilitate composable simulations by standardizing interfaces to assure technical interoperability among disparate simulations. Yet, existing federated simulation infrastructures neither facilitate substantive interoperability nor are dynamically extensible. Emergent web services technologies hold out the potential to significantly improve the development of interoperable, extensible, and dynamically composable federations. As such, recent initiatives (i.e., XMSF) are urging the use of open standards that can be applied within an extensible framework for next generation modeling and simulation applications. We discuss how the realization of multimodel and multisimulation formalisms in terms of semantic web and agent technologies may bring new vistas to demonstrate runtime model discovery, instantiation, composition, and interoperation.

1 INTRODUCTION

An increasingly important trend in the engineering of complex systems is the design of component integration standards. Such standards define rules of interaction and shared communication infrastructures that permit composition of systems out of independently developed components (Davis and Anderson 2003). One problem with these standards is that it is often difficult to understand exactly what they require and provide (i.e., import and export features), and to recognize their substantive properties (Dahmann, Calvin, and Weatherly 1999). To reduce the cost of developing complex simulations and to facilitate the process of building complex collaborative simulation systems, interoperability between disparate models is of paramount importance. Moreover, to make such a system highly extensible, individual federates, which could reside on the same or distributed hosts, should be able to freely join and leave a federation without full knowledge of its peer federates. Simply put, an ideal simulation system should allow for quick and flexible assembly of a complex simulation out of independently developed simulations on demand and at the same time allow the participant simulations to have maximum independence.

1.1 Toward Open and Extensible Simulations with Dynamic Model and Simulation Updating

The need for interoperability and improvement of reuse and composition of existing simulation models is addressed by the concept of a federated simulation system. In federated simulation, the focal goal is to facilitate composable simulations within a system of systems context by standardizing interfaces to facilitate technical interoperability of disparate simulations. Autonomous simulators follow or conform to a specific set of protocols that facilitate seamless data and information exchange. In a federated simulation system, a group of simulation models residing on different machines attached to the network, called federation, collaborate with each other to accomplish a common task of simulating a complex real world system or phenomena.

While the federated simulation concept, in its current form, improves reusability through interface and infrastructure standardization, the integration decision of a federation still takes place at FOM design time. Yet, most realistic problems require scenario evolution that calls for flexibility and adaptivity in simulations with capabilities to locate, discover, and instantiate new federates consistent with emergent conditions. This can be addressed only by open and extensible protocols that can find, match, and infer properties of the services provided by the candidate federate. As such, recent initiatives such as Extensible Modeling and Simulation Framework (XMSF) are targeting the development of web-scale federated simulation technology (Brutzman et al. 2002). XMSF is based on a set of Web-based technologies that can be applied within an extensible framework to enable new generation of modeling & simulation (M&S) applications to emerge and interoperate (Mikalsen and Rouvellou 2002, White and Pullen 2003). Concomitantly, Agent-Based Environment for Linking Simulations (ABELS) is another federated infrastructure that uses software agents to allow simulations to enter and exit a virtual simulation "cloud" of heterogeneous resources (Murphy et al. 2003). The framework uses a limited form of brokering and service matchmaking to facilitate loosely-coupled interactions among disparate simulations. Unfortunately, the lack of machine processable formal annotations describing the behavior, assumptions and obligations of federates is a fundamental roadblock, as such information pertains to (1) finding and matching candidate models, (2) infer limits on the use and interpretation of federates, and (3) perform run-time mediation and facilitation (i.e., translation) among disparate federates. To facilitate formal composability as envisioned here, advances in the following areas are needed:

- **Formalisms:** There is a need for formalisms that form the basis of annotating models with profiles that include assumptions, objectives, and constraints. Such information should be amenable to inference needed to identify and qualify components.
- **Ontologies:** Ontologies (i.e., DAML, DAML-S) based on a specification formalism need to be utilized to capture various facets of models and simulators to describe the kinds of composability information needed in a given problem domain. Yet, the underlying formalism must be general enough to accommodate various problem domains.
- **Profiles:** Each federate model needs to be annotated with a schema that describes the services they provide in terms of domain-specific ontologies. Such service ontologies may include (1) declarative advertisements of model properties and capabilities, in the form to be used for automatic federate discovery, qualification, and instantiation (2) declarative APIs of federates for execution, and (3) declarative specification of the assumptions and obligations of federates and their capabilities to infer the consequences of their use during automatic run-time federate composition.
- **Tools:** To aid the instantiation and configuration of federates, tools are necessary to perform inference and make run-time decisions about composing candidate federate models.

1.2 Limitations on Dynamic Extensibility of Existing Federated Simulation Architectures

There are various limitations of existing federated simulations that can be alleviated with the emergent service-oriented technologies such as semantic web services (Berners-Lee, Hendler, and Lassila 2001) and agent technologies that can exploit the semantic web.

In most realistic phenomena the nature of the problem changes as the simulation unfolds. Initial parameters, as well as models can be irrelevant under emergent conditions. Discontinuities in the behavior of simulation are often the case within realistic joint military exercises, where tactical behavior changes based on observed battlefield conditions. To perform online what-if analysis based on the new emergent and observed conditions, one ideally needs to discover, locate, and join new federates on the fly to continue experimentation. Consider, for instance, the dynamics of military conflicts in general. The trajectory of a realistic conflict scenario is never fixed due to changing attitudes, emotions, and motives between adversarial parties. Dealing with such uncertainty in complex simulation is paramount for realistically modeling and analyzing complex multi-stage phenomena. Adaptivity in simulations and scenarios (not just parameter spaces), is necessary to deal with emergent conditions for evolving systems in a flexible manner. Yet, the current interoperability solutions are not capable of assuring consistent run-time dynamic composability. For instance, Singhal and Zyda (1999, p. 282) argue

> "However, HLA does not go all the way toward supporting dynamically composable simulations and universal reuse. Federation development is static, meaning that the object model and information exchanges must be completed before the simulation run begins."

Given these observations, we explore the potential of web services and emergent semantic web initiatives in deploying extensible federations that can facilitate dynamic run-time simulation composition. Semantic web formalisms, ontology and profile specification approaches, along with agent-based processing capabilities are discussed to explore the utility of service-oriented architectures for federated simulation.

2 DYNAMIC MODEL AND SIMULATION UPDATE

The challenge in dynamic model updating in federated simulation involves issues in substituting a new federate or replacing an existing one without taking the federation offline.

2.1 The Need for Dynamic Model/Simulation Updating

Developing an adaptive and extensible simulation infrastructure is important at least for the following reasons:

- For most realistic phenomena, the nature of the problem changes as the simulation unfolds. Initial parameters, as well as models can be irrelevant under emergent conditions. Relevant models

need to be identified and instantiated to continue exploration. Manual exploration is not cost effective and realistic within a large system state space.
- In exploratory analysis (Bankes 1993), our knowledge about the problem being studied may not be captured by any single model or experiment. Instead, the available knowledge is viewed as being contained in the collection of ensemble of models that are plausible given what is known and what is learned during experimentation.
- Dealing with uncertainty is paramount to analyzing complex evolving phenomena. Adaptivity in simulations and scenarios is necessary to deal with emergent conditions for evolving systems in a flexible manner. As simulations of complex phenomena are more and more used to aid intuition, dynamic runtime simulation composition helps id-entifying solutions that are flexible, adaptive, and robust.
- In symbiotic simulation, which is raised as a grand challenge (Fujimoto et al. 2002), information about the actual performance and accuracy characteristics of the system are acquired during actual simulation rather than before.
- In a recent IEEE Spectrum article (Guizzo 2003) on UrbanSim, which is an advanced city simulation software that helps urban planners look decades ahead, experts conclude as follows: "…. ideally, stakeholders should be able to see computer-generated streetscape animations of **different** scenarios—what it would be like to walk in the neighborhood or commute to work, for example. It would be easier then for people to visualize the effects of **alternative** development policies and projects…."

2.2 Multimodels

Many real-world phenomena can not be modeled by one single model; rather they require the use of a set of complementary models that together are able to describe plausible processes (Ören 2001). A *multimodel* is a modular model that subsumes multiple submodels that together constitute the behavior of a complex multi-phased process. Similar to the concept of encapsulation in object oriented paradigm of software engineering, a multimodel encapsulates several aspects of reality (i.e., submodels) in one model. In special cases, only one aspect of reality can exist at a given time (to be represented by an appropriate submodel) and transitions can occur from one submodel to another one under monitored conditions. However, more than one submodel can also be active at a given time to represent several aspects of a system.

2.3 Multisimulation

Multisimulation (or multisim, for short) is simulation of several aspects of reality in a study. It includes simulation with multimodels, simulation with multiaspect (Ören 2001) models, and simulation with multistage models. *Simulation with multimodels* allows computational experimentation with several aspects of reality; however, each aspect and the transition from one aspect to another one are considered separately. (In special cases, multimodels can be metamorphic models or evolutionary models). Simulation with multiaspect models (or *multiaspect simulation*) allows computational experimentation with more than one aspect of reality simultaneously. This type of multisimulation is a novel way to perceive and experiment with several aspects of reality as well as exploring conditions affecting transitions between several aspects of systems. While exploring the transitions, one can also analyze the effects of encouraging and hindering transition conditions. Simulation with multistage models allows branching of a simulation study into several simulation studies; each branch allowing to experiment with a new model under similar or novel scenarios. *Multistage simulation*, similar to multiaspect simulation, is a type of multisimulation and can allow a novel way to perceive and experiment with several aspects of reality as well as exploring conditions affecting transitions between several aspects of systems, especially social systems.

Multisimulation gaming is a simulation gaming where at decision points, simulation updates may include decisions on the branching of simulation studies as well as selection of models/submodels and associated parameters and experimental conditions to be used in subsequent stages of simulations. In multisimulation gaming, at the interruption of the simulation, one can induce new submodel(s) (and not just data as it is customary in simulation gaming, in general) based on the assessment of the situation so far. This very realistic possibility is not explored yet. Introduction of new submodels and/or models by manual or automated means are complementary possibilities for multisimulation and need to be explored. Multisimulation with multimodels, multiaspect models or with multistage models needs mechanisms to decide when and under what conditions to replace existing models with a successor or alternative.

2.4 Requirements for Simulation Updating

Given the basic conceptual definitions above, the following five conditions in Table 1 present the basic requirements (Litmus test) for dynamic model replacement in federated simulation.

3 SIMULATION ON THE SEMANTIC WEB

In this section we discuss how semantic web services and agent technology can provide foundational facilities for model discovery, location, retrieval, composition, and interoperation.

Table 1: Requirements for Dynamic Federate Update

Requirement	Objective
Discovery, Location, Activation	Federate (re)placement must be initiated, either internally by the federation or externally. New federates must be discovered, located, and activated (pulled) for inclusion in the federation.
Integrity	The consistency of federates undergoing (re)placement needs to be preserved. The event scheduling and simulation protocol need to be restricted or regulated to facilitate interleaving of federate replacement activities with the simulation events.
Instantiation	The new federate must be dynamically loaded and linked into the run-time environment of the federation. This requires new model and simulator decoupling strategies that avoid persistent connections. Also, the intricate details of complex federate construction process should be as independent of the federation as possible to enable flexible update
State construction	The state of a federate must be constructed or at least continue from a specific state after an update operation. This requires externalization through abstraction, state saving, transmission, and reconstruction after the update operation
Rebinding	Once a federate model is loaded and linked to the run-time environment, the federation needs to be bound to the new model (federate).

3.1 Semantic Web Services

Federates can be primitive web services in the sense that they can represent an atomic component. For instance, a sensor model may return at certain time intervals specific threat observation data. Or services represented by a federate can be complex, composed of multiple primitive or complex services. OWL-S (OWL Services Coalition 2004) is meant to support both categories of services, but complex services have provided the primary motivations for the features of the language. An extensible federated simulation framework requires incorporating federates as needed by the federation management protocols. The following discussion gives an idea of the kinds of tasks one can expect from such metaprotocols for web-based federated simulation with the use of OWL-S.

3.2 Automatic Federate Discovery

Automatic federate discovery involves the automatic location of simulations that provide a particular service and adhere to requested constraints. For example, a federation may need to locate a simulation that provides aircraft with specific interaction constraints to be deployed against an emergent threat. Currently, this can be performed by an operator who might use a repository to find a model, read its SOM template to determine whether it satisfies the interaction and attribute constraints, and instantiate the federate. Notice that a federate needs to join to the federation and register itself to make its interactions and attributes visible. With OWL-S markup of federates, the information necessary for service discovery could be specified as computer-interpretable semantic markup at federate repositories, and a service registry or ontology-enhanced search engine could be used to locate federates automatically. Alternatively, a federate could proactively advertise itself in OWL-S with a service registry, also called advertisement, so that requesters can find it when they query the registry. Thus, OWL-S based specification of federates must provide declarative advertisements of service properties and capabilities that can be used for automatic federate discovery.

3.3 Automatic Federate Instantiation

Automatic federate instantiation involves the invocation and execution of a qualified federate. OWL-S markup of services provides a declarative, computer-interpretable API for executing these interaction capabilities within the scope of its protocol. A software agent should be able to interpret the markup to understand what input is necessary to instantiate the federate, what parameters to use in its construction, and how to use its services.

3.4 Automatic Federation Composition and Interoperation

This task involves the automatic selection, composition and interoperation of appropriate federates to address an emergent task. For instance, given a threat determined by a sensor federate or observations of adversarial behavior in the battlefield, an operator in the loop may generate a new tactical plan. The plan may require a fighter aircraft, ground to air missiles with specific ranges, or other components that were not originally foreseen in the original federation design. With OWL-S markup of federates, the information necessary to select and compose services can be encoded and advertised to be brokered by intelligent agents with ontology processing capabilities. Based on the results of a service matchmaking process, a set of federates can be qualified and selected for instantiation. The matchmaking process entails brokering mechanisms that match dynamic runtime requests to the profiles of simulation services. Software can be written to manipulate these representations, together with a specification of the objectives of the task, to support the reconfiguration automatically. Thus, OWL-S based specification of federates must provide declarative specifications of the prerequisites and consequences of individual service use that are necessary for automatic service composition and interoperation.

4 AGENT-BASED SIMULATION BROKERING FOR EXTENSIBLE FEDERATIONS

The semantic markup of federates would enable a wide variety of agent technologies that can facilitate realization of the dynamic model location, discovery, and activation requirement. Section 3 presented the general concepts for the plausibility of dynamic federation updating over the semantic web. The rest of the paper explores dynamic model discovery and location problem, which is the first core requirement listed in Table 1.

4.1 Agent-Based Dynamic Simulation Service Matchmaking Schemes

As the size and number of online model and simulation service repositories increase, locating relevant simulation services at run-time as well as design-time could significantly affect the viability of dynamic simulation updating. Furthermore, existing federation design methodologies assume that model producers are passive, forcing consumers (federation integrators) to drive the process. This imposes several limitations: (1) Federation developers must be aware of or locate relevant simulation models on demand. Today's collaboration networks include large number of contractors that develop and constantly refine existing models. As such, discovering and pulling relevant resources is difficult and inefficient. This would be a significant concern, in particular, when dynamic update mechanisms are realized. (2) Model providers currently have no effective way to contribute their efforts and promote their models for potential utilization. After a federate joins to a federation, there is no means by which a federation can be notified to facilitate runtime update. Brokering can follow many different specific modes as shown in Figure 1.

Agent-based matchmaking protocols could play a critical role to alleviate the limitations presented above. Matchmaking can be defined as a cooperative partnership between a federation and available simulation services available on the semantic web. We suggest using a facilitator agent to enable various brokering protocols among a federation and simulation services. Simulation service providers take an active role in promoting their services by advertising their simulation capabilities to the facilitator. Peteet, Murphy, and Wilson (2003) present a distributed brokering system to facilitate matchmaking among distributed simulation services. A broker in their conceptualization is a central coordinator that exists as a separate agent. The agent encapsulates shared information in the form of knowledge-sharing and transformation protocols as well as a specific brokering algorithm. Yet, their use of JINI architecture imposes a specific mode of brokering that can have limitations on possible means and modes of federation updates.

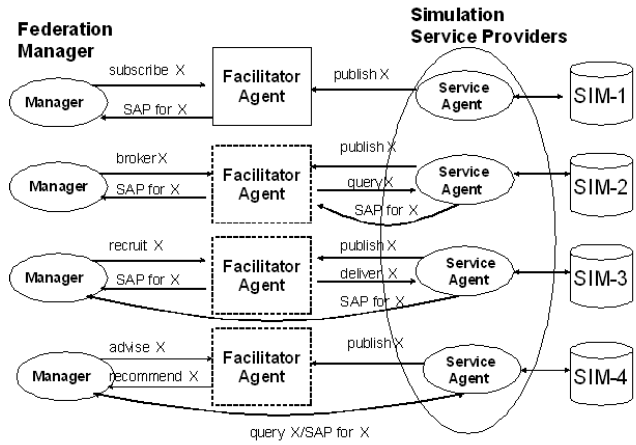

Figure 1: Matchmaking Protocols

The facilitator can enable subscription of a federation manager (or designer) to certain types of simulations during federation design time. As agents associated with simulation service providers publish their models in the advertisement database located within the facilitator, the facilitator can act as an intermediary, forwarding requests to service agents and sending replies to federation managers. This is similar to the approach advocated in (Peteet et al. 2003). Yet, the federation can also request from the facilitator to recommend a simulation service capable of generating the desired behavior with the provision that subsequent replies will be sent directly to the federation. Such a recruitment protocol will decrease the unnecessary load on facilitator agents. The subscription of a federation to certain state information about remote simulation services would enable its notification when simulation scenarios change. Such a facility would especially be critical in dynamic federation update scenarios, where the conditions for update are predicated on the state changes within remote simulation service or the actual system. A useful extension, as shown in Figure 1, would be the utilization of facilitator as online model recommender, where under emergent conditions federation manager consults with the facilitator to receive recommendation for proper models that are consistent with the observed conditions.

4.2 Realizing Simulation Brokering Over Semantic Web

In this section we describe a strategy within an ongoing research for realizing simulation matchmaking strategies over the semantic grid that combines grid research with semantic web constructs. Given a scenario specification that includes federate profiles (attribute and interaction properties), a registry component within a facilitator agent enables internet wide registration of available simulation services. Conventional Universal Description Discovery and Integration (UDDI) facility can be used for registration and location purposes. Yet, UDDI provides a poor search facility based on simple keyword lookups. We suggest adding a new semantic layer to facilitate matching ontology-based model profiles to federation queries. As demand for new federates arises, due to new emerging scenarios, queries for new simulation models can be instantiated autonomously via the federation or manually by operators The architecture of the facilitator constitutes a communication module, matchmaking subsystem that uses simulation service profiles to infer the degree of relevance of a model with respect to a given query, and registries to locate the physical address of the simulation service for invocation. The architectural representation, shown in Figure 2, assumes similarity analysis initiated by an operator at the simulation interruption time to find federates that are relevant to emergent tactical plans or demands. The architecture presents a simplified illustration of the data flow during a matchmaking session. For the sake of brevity, we only discuss measuring concept similarity as a metric for determining the relevance. For concept-based similarity analysis, the federation protocol sends to the facilitator the required concept (i.e., *jammer* federate in a *deception systems* taxonomy) along with a reference to a service taxonomy. The communication module of the facilitator interprets the message interchange format as a query and forwards it to the matchmaker to facilitate similarity analysis based on the available service profiles.

5 MATCHING FEDERATE CAPABILITIES USING MODEL TAXONOMIES

While primitive keyword matching can be applied to assure syntactic matching between keyword, the approach could be overly conservative. Same concepts may be defined by different variables and keywords. Furthermore, inexact matches may be sufficient for qualifying a simulation service. For instance, considering the ontology depicted in Figure 3, a federation that requires a warplane may find a bomber or fighter plane sufficient for its domain of applicability. As such our approach extends the basic syntactic keyword filtering and variable matching strategy (Murphy et al. 2003) to concept similarity analysis to facilitate inexact semantic matching.

OWL-S supports the need for semantic representation of simulation services through its tight connection to DAML+OIL that supports subsumption reasoning based on taxonomies of concepts. As such, the service category component of a service *profile* includes a reference to a

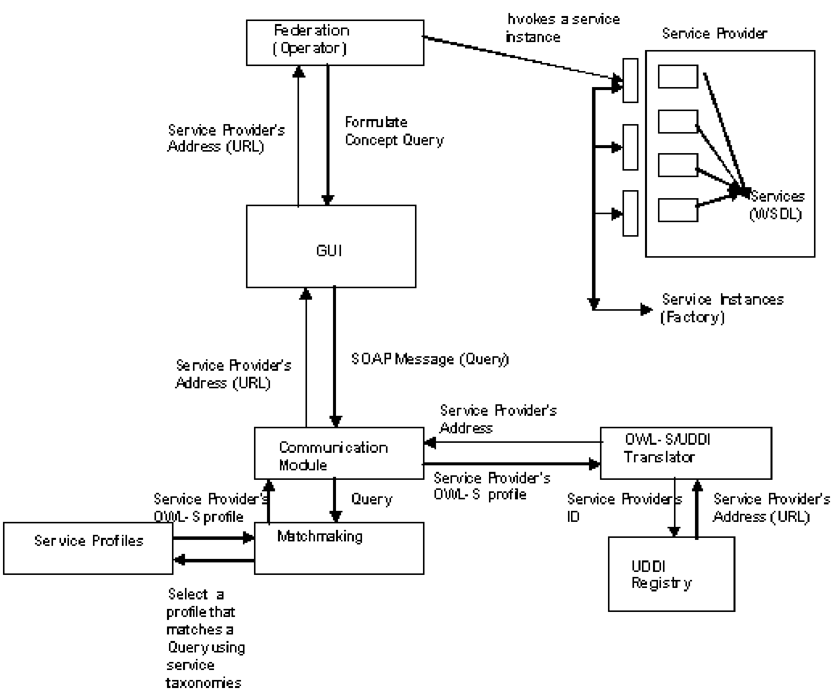

Figure 2: Components of a Facilitator Agent

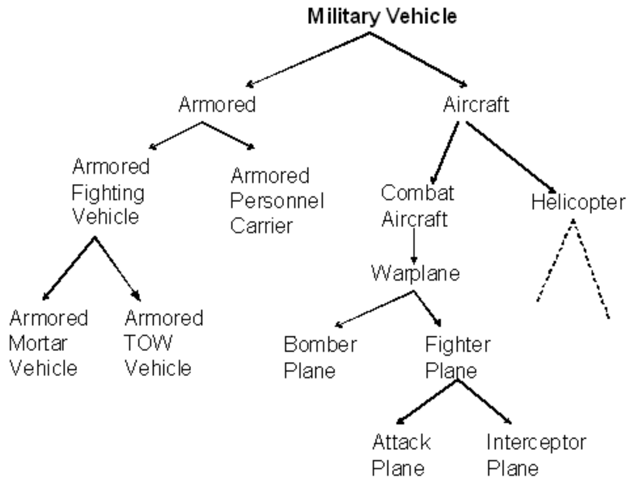

Figure 3: A Partial Vehicle Ontology

schema, along with the category name (concept) by which the federate can be classified. DAML+OIL subsumption reasoning capability can be exploited to perform concept similarity analysis to locate relevant simulation services for the emergent requirements. The matching engine within the facilitator agent should support flexible matching between the federate advertisements and requests on the basis of available ontologies.

To increase the efficiency of selecting a unique candidate, the matchmaking engine needs to minimize falsepositives. The accuracy of a matchmaking scheme depends on the extent to which false negatives are avoided. Furthermore, a matchmaking algorithm should be sufficiently efficient to avoid excessive delays during run-time federate discovery and location. Consider, for instance, the partial view of the military vehicle ontology, as shown in Figure 3. Suppose that the federation requires a new vehicle of type *combatAircraft* based on the presented military vehicle taxonomy. The conceptual context of *combatAircraft* model within the taxonomy (i.e., ancestors and descendants in the taxonomy tree) suggests plausible models for qualification, when an exact match can not be found. More specifically, consulting with the advertisement database, the matching engine needs to determine the relevance of existing federates based on their profile definition. Each profile definition designates the concept that the federate supports, along with the service category (i.e., ontology) it is indexed into. Note that a federate can be classified under several taxonomies. Therefore, the query needs to explicitly define the ontology that will provide the basis for similarity analysis. Based on the location of the query concept (Q) and advertised federate (A) within the ontology, there are four cases to consider:

1. If Q=A then the query and advertisement are equivalent and there is an exact match between the concepts.
2. If A is a subtype of Q then the federate is substitutable for the requested concept.
3. If Q is a subtype of A then the advertised federate can be used with slight modifications to perform the desired tasks.
4. If there is no subsumption relation between the advertised concept A and Q then the query fails.

The degree of match can simply be defined as the ratio of the length of the minimum path between Q and A to the height of the taxonomy (classification) tree. The smaller the ratio, the more relevant is the federate under examination. This simple brokering mechanism can be extended to perform more sophisticated analysis on the preconditions and postconditions of simulation capabilities defined in terms of simulation service profiles. Each service entails functions that take parameters and changes the state of an object in a federation. OWL-S provides facilities to associate precondition and postcondition predicates for functional services. Hence, matchmaking can take place at a lower level of fidelity. By representing pre and postconditions in terms of concepts (i.e., a federate interaction requires concept X as input and generates concept Y as output), the similarity analysis approach can be utilized. A weaker precondition implies a more general concept. That is, (1) if the precondition of an advertised capability is associated with a more general concept (i.e., warplane) than the precondition of the query (i.e., fighter plane) and (2) if the postcondition of the capability is associated with a more specific concept (i.e., hit armored mortar vehicle), while the postcondition of the query is related to an equivalent or more general concept (i.e., hit armored vehicle), then the capability can be selected for use.

6 CONCLUSIONS

The focal goal in federated simulation is to facilitate composable simulations within a system of systems context by standardizing interfaces to facilitate interoperability. In this paper, it is observed and argued that the lack of machine processable formal annotations describing the behavior, assumptions and obligations of federates is a fundamental roadblock, as such information pertains to (1) finding and matching candidate models, (2) inferring limits on the use and interpretation of federates, and (3) performing run-time mediation and facilitation (i.e., translation) among disparate federates.

Our second observation is the inherent limitation of existing federated simulation environments in supporting dynamic model and simulation updating. HLA federation development, for instance, requires complete specification of object models and information exchanges before the simulation run begins. At runtime, federates may enter and leave the simulation at will, as long as they conform to the predefined object model. Given these observation, we explored the potential of web services and emergent semantic web initiatives in deploying open and extensible federations that can facilitate dynamic run-time simulation composition.

REFERENCES

Bankes S. 1993. Exploratory Modeling for Policy Analysis. Operations Research, Vol 4. No. 3, 435-449.

Berners-Lee T., J. Hendler, and O. Lassila 2001. The Semantic Web. Scientific American, May Issue. Available online via `http://www.sciam.com /article.cfm?chanID=sa006&colID=1&articleID=00048144-10D2-1C70-84A9809EC588EF21` [accessed August 1, 2004].

Brutzman D., K. J. Morse, M. Pullen, and M. Zyda 2002. Extensible Modeling and Simulation Framework (XMSF): Challenges for Web-Based Modeling and Simulation, Interim Technical Report. Naval Postgraduate School, Monterey, California. Available online via `http://www.movesinstitute.org/xmsf/workshop/XmsfWorkshopReportDraft.pdf` [accessed August 1, 2004].

Dahmann J., J.O. Calvin, and R. Weatherly 1999. A reusable architecture for simulations. Communications of the ACM, Vol. 42, No.9, 79-84.

Davis K. P. and A. R. Anderson 2003. Improving the Composability of Department of Defense Models and Simulations, RAND.

Fujimoto R., D. Lunceford, E. Page, and A. Uhrmacher 2002. Technical Report of the Dagstuhl-Seminar Grand Challenges for Modeling and Simulation.

Guizzo E. 2003. Software and the City. *IEEE Spectrum*, Available online via `http://www.spectrum.ieee.org/WEBONLY/wonews/dec03/1203urb.html` [accessed August 1, 2004].

Mikalsen T. and I. Rouvellou 2002. Transactional attitudes: Reliable composition of autonomous Web services, IBM Watson Research Center, Available on-line via `http://xml.coverpages.org/ni2002-04-03-a.html` [accessed February 10, 2004].

Murphy P. J., A. Mills-Tettey, L. F. Wilson, Greg Johnston, B. Xie 2003. Demonstrating the ABELS System Using Real-World Scenarios, In Proceedings of the SAINT 2003.

OWL Services Coalition 2004. OWL-S: Semantic Mark-up for Web Services. Available online via `http:// www.daml,org/services/owl-s/1.0/owl-s.html` [accessed April 04, 2004].

Ören T.I. 2001. Towards a Modelling Formalism for Conflict Management, In: Discrete Event Modeling and Simulation: A Tapestry of Systems and AI-based Theories and Methodologies, H.S. Sarjoughian and F.E. Cellier (eds.), Springer-Verlag, New York, pp. 93-106.

Peteet J, John P. Murphy, and Linda F. Wilson 2003. Matchmaking in the ABELS System for Linking Distributed Simulations. In Proceedings of the 15TH European Simulation Symposium.

Singhal S. and M. Zyda 1999. Networked Virtual Environment: Design and Implementation, Boston, Addison-Wesley.

White L. E and J. M. Pullen 2003. Adapting Legacy Computational Software for XMSF, Simulation Interoperability Workshop (SIW), Fall 2003, Orlando, Florida.

AUTHOR BIOGRAPHIES

LEVENT YILMAZ is Assistant Professor of Computer Science and Engineering in the College of Engineering at Auburn University and co-founder of the Auburn Modeling and Simulation Laboratory of the M&SNet. Dr. Yilmaz received his Ph.D. and M.S. degrees from Virginia Polytechnic Institute and State University. His research interests are (1) advancing the theory and methodology of modeling and simulation to explore understanding and managing conflicts and (2) agent-directed simulation. He has published in the areas of agent-directed simulation, conflict modeling and simulation, V&V, and reuse. He currently serves as the associate chair for the Theory and Methodology Council of the Society for Computer Simulation International. He is a member of ACM, IEEE, Society for Computer Simulation International, and Upsilon Pi Epsilon. *URL:* `<http://www.eng.auburn.edu/~yilmaz>`

TUNCER ÖREN is a professor emeritus of computer science at the School of Information Technology and Engineering of the University of Ottawa (Canada). He has contributed to: (1) the advancement of modeling and simulation methodologies; (2) the synergy of: simulation, system theories, cybernetics, artificial intelligence, and software engineering; (3) the reliability issues in modeling and simulation; and (4) ethics in simulation. Currently member of the Board of Directors, AVP on Ethics, and Director of the McLeod Modeling and Simulation Network of the Society for Modeling and Simulation International. Over 330 publications; some translated in Chinese and German. Contributions in over 300 conferences and seminars held in 28 countries. Invitations from United Nations; sponsorship from NATO; and fellowships or sponsorships in several countries. Over 20 Who's Who citations. "Information Age Award" -from the Turkish Ministry of Culture (1991) and plaques and certificates of appreciation from organizations including NATO, Atomic Energy of Canada and ACM. *URL:* `<http://www.site.uottawa.ca/~oren/>`