A SEMANTIC APPROACH TO SIMULATION COMPONENT IDENTIFICATION AND DISCOVERY

Marianela García Lozano[(a)], Farshad Moradi[(b)], Eneko Ibarzabal[(c)], Edward Tjörnhammar[(d)]

[(a,b,d)]FOI, Swedish Defence Research Agency, 164 90 Stockholm, Sweden
[(c)]KTH Royal Institute of Technology, 100 44 Stockholm, Sweden

[(a)]garcia@foi.se, [(b)]farshad@foi.se, [(c)]eneko@kth.se, [(d)]edward@foi.se

ABSTRACT
Development of simulations is often a costly process that consumes a lot of time and resources. An appealing approach to reduce the costs involved is to reuse and recombine existing predefined models and components through a composition process.

This process is a complex task that involves four main steps: identification, discovery, matching and composition. The focus of this work is to show how different ontologies combined with semantic querying enables more accurate identification and discovery of components in the process. We present a methodology for achieving this goal and clarification through a use case.

Our preliminary results indicate that our approach is feasible, and semantic techniques contribute to both finer descriptions and search results.

Keywords: Model composition, Semantic description, Ontology, BOM

1. INTRODUCTION

Modelling and Simulation (M&S) is a well known tool and method for e.g. exploring phenomenon that are too costly or impossible to explore in the real world. However, the development of simulations is many times an expensive process that consumes a lot of time and resources. An approach to reduce costs is to reuse existing models, mainly via recombination (Oses, Pidd and Brooks 2004).

To combine components into models a composition process of several steps is required. First, it is necessary to have an idea of what is to be simulated, i.e. to have a conceptual model of the simulation. From this, both the kinds of components and the requirements that are needed for the simulation may be identified. Then, from a set of components, viable candidates need to be located and selected. Finally, a matching of discovered components and comparison of different composition alternatives to the requirements needs to be done (Moradi, Nordvaller and Ayani 2006).

While the later steps of the process have been subject to research, the earlier steps are more or less untouched territory. In this paper we focus on these first steps i.e. the simulation description, and the identification and discovery of components.

The goal of this work has been to improve these steps by identifying and selecting components as accurately as possible i.e. only selecting relevant components and reducing the total number of composition alternatives to check. Hence, making the steps more precise and automated.

We propose a refined simulation composition process introducing semantic identification and discovery of components by *i*) coupling Simulation Reference Markup Language (SRML) documents with domain ontology information thus allowing a parser to extract more accurate information on the simulation components sought *ii*) presenting a Base Object Model (BOM) ontology which we together with a domain ontology use to describe simulation components *iii*) storing the descriptions in a semantic based distributed repository, and then retrieving viable candidates by *iv*) using SPARQL Protocol and Resource Description Framework Query Language (SPARQL - a recursive acronym) queries built from parsing the SRML documents. We have tested and evaluated our approach through a set of case studies and examples, which are presented in this paper.

1.1. Composability

One of the main challenges when composing simulation components is the issue of composability. Composability has been defined as the capability to select and assemble reusable simulation components in various combinations into simulation systems to meet user requirements (Petty 2004).

When addressing composability there are several issues to take into account. Composability checking of components needs to be performed on several aspects, syntactic, semantic and pragmatic.

- Syntactic composability focuses on the implementation aspects of each component, e.g. checking in and out parameters (Szabo and Teo 2007).
- Semantic composability addresses whether the combined computation of the components in the model is semantically valid (Petty 2004).
- Pragmatic composability addresses whether the model is meaningful with regard to the context in which it is to be used (Hofmann 2002).

As the definition suggests, syntactic composability is handled at the technical level to make sure that components have the right interfaces to interact with each other. Semantic composability however should be addressed at conceptual level. This requires precise definition and specification of components' syntax and semantics in order to capture the basic requirements for matching and synthesizing a semantically meaningful composition of those components. Here is a need for a common methodology for specification of simulation components consisting of meta-models describing the components at different levels. In order to enable automatic matching of meta-models they should be formalized and structured using ontologies. This is to avoid misunderstanding and to provide unambiguous definitions as a basis for reasoning about syntactic and semantic validity of compositions (Moradi et.al. 2009).

1.2. Relevant Techniques

The common denominator for all composition infrastructures is that they all besides the components themselves require a component repository, composition and query tools, and meta-data standards (Bartholet et.al. 2004). In this work we have utilized the following technologies, standards and tools.

The Simulation Reference Markup Language, see (SRML Web Site), is a markup language based on XML and a reference standard for representing simulations. It has enough expressiveness to model almost anything for the purpose of a simulation. SRML has been used in this work as a format for describing simulations scenarios.

The Document Object Model (DOM) is an API for XML and HTML documents. It is used to represent the document's logical structure and the document object's interaction possibilities. It is also language-independent.

The Base Object Model concept, see (BOM Web Site), has been identified and developed within High Level Architecture (HLA), see (HLA IEEE Standard), as basic components for rapid development of HLA object models. The goal with BOM is to enable reusability, interoperability and composability. It is based on the assumption that simulation and model parts may be extracted and reused as building blocks or components.

Ontologies are the corner stone of the Semantic Web initiative. Semantic Web is an attempt to simplify search and query of information on the web, and make the information understandable by computers, hence facilitating communication between them. Ontologies can be used to create a common understanding/context between components and describe, among others, entities, hierarchies, relations and attributes (Miller et.al. 2004, Chenine, Kabilan and Garcia Lozano 2006).

Ontologies are described using languages such as, Resource Description Framework (RDF) (RDF Web Site) and Web Ontology Language (OWL) (OWL Web Site). RDF is basically a directed graph data format used to represent information. With OWL there is a better possibility of defining restrictions in the properties that relate different classes and data types.

SPARQL (Editors Prud'hommeaux, Seaborne 2008) is an RDF query language that enables queries consisting of triple patterns, their conjunctions and disjunctions. Query results can be sets or RDF graphs.

Besides the components themselves one of the main parts of a composition infrastructure is a repository facility which provides management and storage of, and access to those components. SDR is a Semantic based Distributed Repository (Garcia Lozano, Moradi and Ayani 2007) developed with the aim to enable sharing of resources with an overlay architecture. Resources may be anything that can be described e.g. simulation models, components, software, hardware, etc. Discovery of these resources is a main functionality and to make discovery as precise as possible semantic techniques are utilized.

2. APPROACH

In order to achieve our objectives, presented in the previous section, we develop an automated system for semantically retrieving simulation components. We assume that: *i*) each component is represented by a BOM model, stored in a distributed repository and *ii*) the simulation scenario is described by an SRML file.

The idea is that with the information specified in an SRML file describing the entities and interactions taking part in a simulation, and by semantically describing each BOM stored in the system, the process should be able to identify and discover relevant components automatically. Three aspects are of importance here; first, how the parsing of the SRML file is done and how the components are identified, second, how BOM components can be described, such that the information given in the descriptions corresponds to the information retrieved from the SRML file, and third, how the query used to discover the sought after BOMs should be formulated.

For the purpose of enriching BOM descriptions with semantic information we have defined a BOM ontology, which is presented in the following section.

2.1. BOM Ontology

The BOM ontology is implemented to enable semantic discovery of the simulation components. It provides a means for agreeing on the meaning of the different parts of a BOM description and their relationship. Thus enabling the mapping of features described in SRML to BOM "language". The ontology is specified in OWL and describes the structure of the BOM parts, the restrictions and relationships, specifying what is and what is not allowed when describing BOMs.

A BOM description consists of four major parts; Model Identification, Conceptual Model, Model Mapping and HLA Object Model.

The *Model Identification* part keeps track of meta data information about the BOM component and is used to document key meta data about the component i.e. information such as use history, unsuccessful inclusions in simulations, description, creation date, information about how to contact the BOM's developers, application domain and purpose, etc.

The *Conceptual Model* contains information about the behavior of the component and how it actually works. It has four sub-parts, Pattern of Interplay, State Machine Model, Entity types and Event Types. Pattern of interplay contains actions, entity types and event types that take part during an interaction between two components. State Machine Model models how entity types move from one state to another via actions and conditions that must be fulfilled.

The third part is the *Model Mapping*, which describes the mapping between entity and event elements from BOMs to their counterparts in HLA's object model.

The fourth part is the *HLA Object Model* that provides the HLA OMT information like HLA object and interaction classes, their attributes and parameters, HLA data types.

The semantic information of a BOM, which is our main concern here, is contained in the two first parts. In the ontology we describe these parts together with existing relationships in OWL. The individual parts are described with classes named correspondingly. The attributes, relationships between classes and constraints have been modelled with data and object type properties, and restrictions. We have also added a central class called BOMContext which defines a BOM component. The BOMContext is related to the other classes; Identification Model, Entity Type, Event Type, Pattern of Interplay and State Machine with the hasBOMContext and isBOMContextOf object properties. For further details on the BOM Ontology, see (Ibarzabal 2008).

2.2. Methodology

The steps in our methodology are as follows:

1. We start by parsing the SRML document where we identify the simulation elements (an SRML file may contain multiple simulations). Having done that we build a DOM tree containing the document's elements, and store it in memory. A DOM tree contains a hierarchy of node objects with tags representing the object's names.
2. In this step we extract the information required to build the query from the DOM tree. The step begins by selecting and iterating through the simulation objects. In the XML domain an SRML Simulation description, Item and Property corresponds to Document, Element and Attribute. When processing each simulation we retrieve the ItemClass elements and their attributes such as Name and other values.
3. To build the SPARQL query we map the items' names to the Entity Type class defined in the BOM ontology. Item's attributes are then mapped as Entity Attributes.
4. The final step is to send the SPARQL query to the simulation component repository, in our case SDR, and ask for the components matching our query requirements.

In order to narrow down the possible component matches by restricting the hits to the ones from the relevant domain we also use a domain ontology in our methodology.

Since SRML does not include a domain specification, we add a parameter to the SRML document, which is a link to the domain ontology for that simulation. The same ontology is also used in the SPARQL query and when describing simulation components (BOMs).

The steps of our Identification and Discovery methodology can be seen in Figure 1.

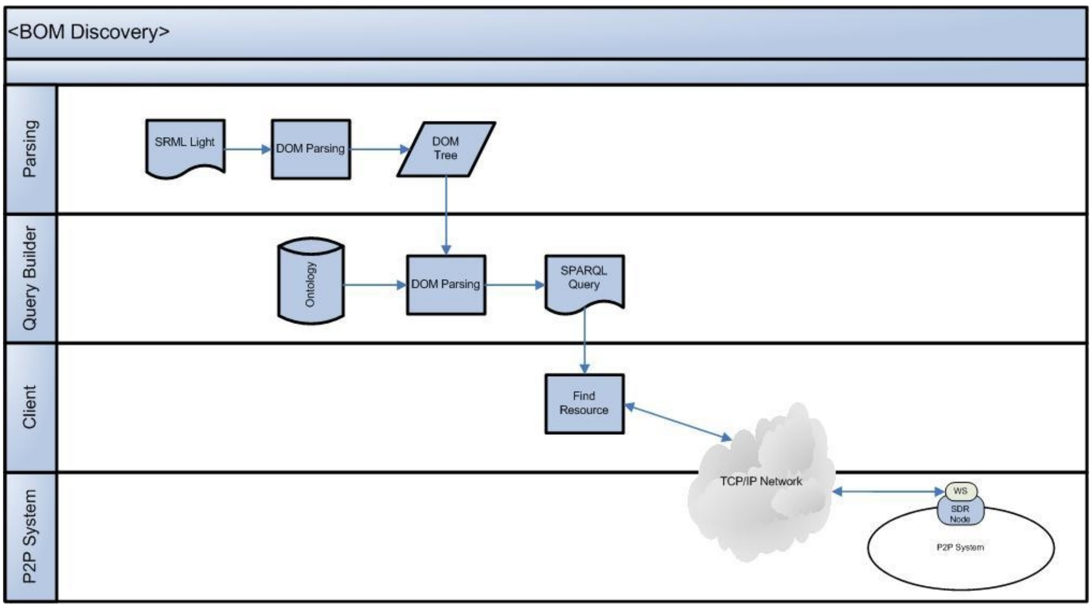

Figure 1: The process for identifying and discovering BOM models

3. USE CASES

To test our hypothesis, i.e. that it is possible to improve the simulation component identification and discovery with semantic techniques, we have built two different use cases. The first use case is based on a simulation requiring two models called “Tiger” and “Hawk”. Depending on the target domain they could represent (in the animal domain) a bird of prey and a cat, (in the military domain) a tank and helicopter, or (in the sports domain) the golf pro Tiger Woods and the skateboard pro Tony Hawk. A simple key word search without considering the domain of interest would result in a high recall with low precision. For example, searching for “Tiger” would result in many animal simulation models as well as tank and golf pro models. Taking the domain into consideration would yield lower recall and higher precision.

The second use case is based on the vehicle domain. In this use case we have different vehicle models like; bus, tractor, car and sports car. We also have models for the trips that can be made with the vehicles, and people interacting with them.

Listing 1: Small example of an SRML document

```
  ...
1.   <Simulation xmlns="urn:x-
 schema:SRML.xdr"
2.     xmlns:pt="http://sdr.foi.se/ontology/
      PublicTransport.owl">
3.     <ItemClass name="pt:Person" />
4.     <ItemClass name="pt:Customer"
        SuperClasses="pt:Person">
5.       <pt:Customer pt:name="Jon"
          pt:age="29"
          pt:travels="#BUS14">
6.         <Script Type="text/javascript">
7.           BroadcastEvent(all, "Greet",
         "Good Morning");
8.           SendEvent(Driver, "Pay");
9.         </Script>
10.      </pt:Customer>
11.    </ItemClass>
12.    <ItemClass name="pt:Driver"
        SuperClasses="pt:Person">
13.      <pt:Driver pt:name="Mike"
          pt:drives="#BUS14">
14.      </pt:Driver>
15.    </ItemClass>
16.    <ItemClass name="pt:Trip">
17.      <pt:Trip pt:name="route66"
          pt:hasSource="#Stockholm"
          pt:hasDestination="#Uppsala">
18.      </pt:Trip>
19.    </ItemClass>
20.    <ItemClass name="pt:BUS">
21.      <pt:BUS pt:name="BUS14"
          pt:hasDriver="#Mike"
          pt:hasCustomer="#Jon"
          pt:hasRoute="#route66">
22.        <EventSink Name="Greet" />
23.      </pt:BUS>
24.    </ItemClass>
25.  </Simulation>
```

To follow the methodology described in Section 2.2 we will begin by looking at a small example of an SRML document, see Listing 1. In this example we can see a simulation description containing different people models i.e. a driver and a customer. There is also the vehicle type bus and trip model describing the route and destination of the bus. Note that we have added a name space called **pt**, which stands for the PublicTransport domain ontology, see line 3. Note also that in this example, due to space constraints, we use the pt name space in the attribute values even though that is not allowed according to the XML standard. In step 1 and 2 of the methodology the SRML file is parsed to extract the Items and their properties.

The public transport ontology used in the SRML file can be seen in Figure 2. It gives an overview of the classes and their relationships.

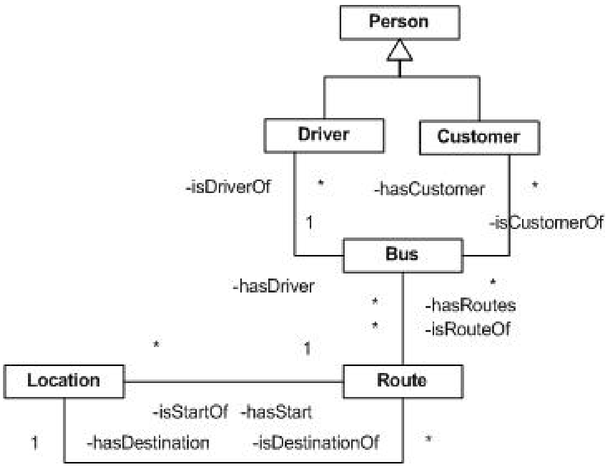

Figure 2: An ontology describing the public transport domain

Listing 2: The SPARQL Query Obtained from Parsing the SRML Document

```
PREFIX bom:
  <http://sdr.foi.se/ontology/bom.owl#>
PREFIX pt:
  <http://sdr.foi.se/ontology/
  publicTransport.owl#>
SELECT ?x
WHERE {
  {
    ?x bom:isBOMContextOf ?x0 .
    ?x0 bom:name "pt:Person" .
  } UNION {
    ?x bom:isBOMContextOf ?x1 .
    ?x1 bom:name "pt:Customer".
    ?x1 bom:isEntityTypeOf ?x10 .
    ?x10 pt:name "Jon" .
    ?x10 pt:age "29" .
    ?x10 pt:travels "#BUS14" .
  } UNION {
    ?x bom:isBOMContextOf ?x2 .
    ?x2 bom:name "Bus".
    ?x2 bom:isEntityTypeOf ?x20 .
    ?x20 pt:hasDriver "Mike" .
    ?x20 pt:hasRoute "route66" .
  }
}
```

After parsing the SRML file we move on to step 3 of the method where the resulting (slightly shortened) SPARQL query can be seen in Listing 2. The query will look for all BOMs that are of the right domain and contain the right kind of elements i.e. is either a Person, Customer and/or BUS component. The original domain ontology is also supplied here as a PREFIX called pt. This will ensure that when we send the query to the semantic based component repository only BOM components from the correct domain will be returned.

In step 4 when we send the SPARQL query to SDR we get a list of the matching BOM components. An example of how a retrieved semantic BOM description could look can be seen in Listing 3. It describes a Bus model that also is a school bus. It also has different EntityAttributes like chassis and weight.

Listing 3: Example of a Semantic BOM Description

```
...
<rdf:RDF ...
  xmlns:pt="http://sdr.foi.se/ontology/
    publicTransport.owl">
  xml:base="http://sdr.foi.se/ontology/
    bom.owl">
<BOMContext rdf:ID="busBOM1">
  <name rdf:datatype=
    "http://www.w3.org/2001/
    XMLSchema#string">
    Bus BOM1
  </name>
  <isBOMContextOf rdf:resource=
    "#SchoolBusA380"/>
</BOMContext>
<EntityType rdf:ID="SchoolBusA380">
  <name rdf:datatype=
    "http://www.w3.org/2001/
    XMLSchema#string">
    School Bus A380
  </name>
  <isEntityTypeOf rdf:resource=
    "#chassis"/>
  <isEntityTypeOf rdf:resource=
    "#weight"/>
  <hasBOMContext rdf:resource=
    "#busBOM1"/>
  <isSubClassOf>
    <BOMContextrdf:ID=
    "VehicleBOMContext"/>
  </isSubClassOf>
</EntityType>
<EntityAttribute rdf:ID="chassis">
  <name rdf:datatype=
    "http://www.w3.org/2001/XMLSchema#str
    ing">
    Chassis
  </name>
  <value rdf:datatype=
    "http://www.w3.org/2001/
    XMLSchema#string">
    RJ 45
  </value>
  <hasEntityType
    rdf:resource="#SchoolBusA380"/>
</EntityAttribute>
<EntityAttribute rdf:ID="weight">
  <name rdf:datatype=
    "http://www.w3.org/2001/
    XMLSchema#string">
    Weight
  </name>
  <value rdf:datatype=
    "http://www.w3.org/2001/
    XMLSchema#string">
    12 Tons
  </value>
</EntityAttribute>
...
```

4. TEST AND EVALUATION

To make the tests we created different semantic BOM descriptions following the described use cases. These descriptions were then stored in SDR. Note: the BOM descriptions do not need to be stored in SDR but could be stored in any OWL compatible RDF triple store. In parallel we also made some matching SRML descriptions. Following the described methodology they were parsed and the resulting SPARQL queries were evaluated.

The results from our tests, even though preliminary, show that the domain specific semantic queries give an exhaustive, exact and relevant result set. There is however need for further tests and experiments. The next step will be to compare semantic queries with keyword based queries on a large data set in order to be able to draw better conclusions regarding benefits of using domain specific semantic searches.

However, one can already assume that less specific keyword based queries would have a lower precision and with a significantly higher recall, thus yielding more noise.

4.1. Evaluation

Based on our preliminary results semantic queries have higher accuracy rate and yield more relevant result set. However, in our experience they are harder to implement and if the set is a large one it is harder to eliminate hits. This is in part due to inefficiencies in SPARQL where we cannot create cascaded queries.

Key word based queries on the other hand are cheaper to evaluate but result in massive result sets with a lot of noise. They are also easier for a user to formulate.

Building ontologies is a cumbersome task which requires domain and application knowledge, and also modelling expertise. Semantic BOM building is not a trivial task either since it requires knowledge of both the BOM and domain ontology. The solution is to construct a tool which transparently aids in interleaving the BOM ontology with any other domain ontology.

5. SUMMARY AND CONCLUSIONS

To summarize our work we have refined the composition process by improving the simulation component identification and discovery. We have achieved this by using a semantic approach and automating the process. The method described has improved the relevance of the discovered simulation components. We have also automated the process by describing and storing the simulation components (BOMs) in a semantic distributed repository combined with an SRML parser. To enable semantic query evaluation we have built a BOM Ontology and introduced a new parameter to the SRML document allowing the inclusion of a domain ontology.

Based on this work our conclusions are that *i*) there exists a mapping between SRML and BOM descriptions, *ii*) it is possible to make a BOM ontology, *iii*) it is possible to automate simulation component identification and discovery, and *iv*) it is possible to improve the relevance of simulation component discovery compared to not using semantic techniques.

REFERENCES

Bartholet, R. G., Brogan, D. C., Reynolds, P. F. Jr. and Carnahan, J. C., 2004. In search of the philosopher's stone: Simulation composability versus componentbased software design. *In Proceedings of the 2004 Fall SIW*, Orlando, FL,

BOM - Base Object Model. Web Site. http://www.boms.info/ [Accessed June 2009]

Chenine, M., Kabilan, M. V. and Garcia Lozano, M., 2006. A pattern for designing distributed heterogeneous ontologies for facilitating application interoperability. *EMOI-INTEROP.*

Ibarzabal, E., 2008. *Enabling Semantic Discovery of Simulation Components*. Master's thesis, KTH, TRITA KTH/ICT/ECS 2008-40.

Editors Prud'hommeaux, E. and Seaborne, A., 2008. SPARQL query language for RDF. Technical report W3C, January.

HLA - High Level Architecture. IEEE Standard 1516.

Hofmann, M., 2002. Introducing Pragmatics into VV&A. *In Proceeding of the European Simulation Interoperability Workshop*, London, UK,

Garcia Lozano, M., Moradi, F. and Ayani, R., 2007. SDR: A Semantic based Distributed Repository for simulation models and resources. *In AMS'07: Proceedings of the First Asia International Conference on Modelling & Simulation*, pages 171-176, Washington, DC, USA, IEEE Computer Society.

Miller, J. A., Baramidze, G. T., Sheth, A. P. and Fishwick, P.A., 2004. Investigating ontologies for simulation modeling. *In ANSS '04: Proceedings of the 37th annual symposium on Simulation*, page 55, Washington, DC, USA, IEEE Computer Society.

Moradi, F., Nordvaller, P. and Ayani, R., 2006. Simulation model composition using BOMs. *In DSRT '06: Proceedings of the 10th IEEE international symposium on Distributed Simulation and Real-Time Applications*, pages 242-252, Washington, DC, USA, IEEE Computer Society.

Moradi, F., Ayani, R., Mokarizadeh, S. and Tan, G. 2009. A rule-based semantic matching of Base Object Models, To be published in the *International Journal of Simulation and Process Modelling*.

Oses, N., Pidd, M. and Brooks, R. J., 2004. Critical issues in the development of component-based discrete simulation. *Simulation Modelling Practice and Theory*, vol. 12, pp. 495-514.

OWL - Web Ontology Language Overview. W3C Web site. http://www.w3.org/TR/owl-features/ [Accessed June 2009]

Petty, M. D., 2004. Simple composition suffices to assemble any composite model. *In Proceedings of the Spring Simulation Interoperability Workshop* Orlando FL, April 18-23.

RDF - Resource Description Framework. W3C Web Site. http://www.w3.org/RDF/. [Accessed June 2009]

SRML - Simulation Reference Markup Language. W3C Web Site. http://www.w3.org/TR/SRML/ [Accessed June 2009]

Szabo, C. and Teo, Y. M., 2007. On syntactic composability and model reuse. *In AMS '07: Proceedings of the First Asia International Conference on Modelling & Simulation*, pages 230-237, Washington, DC, USA, IEEE Computer Society.

AUTHORS BIOGRAPHY

Marianela García Lozano is a Scientist at the Swedish Defence Research Agency (FOI). She holds a Master of Science in Computer Science from KTH, Royal Institute of Technology, and is now pursuing her PhD in the field of Semantic Resource Discovery in Distributed Systems. She has worked at FOI since 2001, with information and knowledge modelling, software development in distributed systems, Grids, Semantic Web, modelling and simulation, and Service Oriented Architectures.

Farshad Moradi is a senior scientist and programme manager in the area of modelling and simulation, at the Swedish Defence Research Agency. He has been working on modelling and simulation for the past 14 years and has led numerous projects in this area. He holds a Master of Science in Computer Science and Engineering from Chalmers University of Technology, and a PhD in Distributed Simulations from KTH, Royal Institute of Technology. His research interests are in the areas of Distributed Systems, Distributed and Web-based Modelling and Simulation, Embedded Simulations, Service-Oriented Architectures, Group behavioural modelling, and Logistics.

Eneko Ibarzabal holds a Master of Science in Engineering, Information and Communication Technology (with the major in Software Engineering and Distributed Systems) from KTH, Royal Institute of Technology, He did his master thesis at FOI with the topic of "Enabling Semantic Discovery of Simulation Components".

Edward Tjörnhammar is a research engineer at the Swedish Defence Research Agency. His research areas, and interests, are in distributed and parallel systems, information modelling and functional languages. He has also worked as a simulation technician at the Division of Air Combat Simulation (FLSC) and has experience in grid technology and network security. His studies were conducted at KTH, Royal Institute of Technology, in distributed and parallel systems.