Short paper: Using Formal Ontologies in the Development of Countermeasures for Military Aircraft

Nelia Lombard[1,2], Aurona Gerber[2,3], and Alta van der Merwe[3]

[1] DPSS, CSIR
nlombard@csir.co.za
http://www.csir.co.za
[2] CAIR - Centre for AI Research
Meraka CSIR and Univerity of Kwazulu-Natal
http://www.cair.za.net/
[3] Department of Informatics
University of Pretoria, Pretoria
http://www.up.ac.za/
South-Africa

Abstract. This paper describes the development of an ontology for use in a military simulation system. Within the military, aircraft represent a significant investment and these valuable assets need to be protected against various threats. An example of such a threat is shoulder-launched missiles. Such missiles are portable, easy to use and unfortunately, relatively easy to acquire. In order to counter missile attacks, countermeasures are deployed on the aircraft. Such countermeasures are developed, evaluated and deployed with the assistance of modelling and simulation systems. One such system is the Optronic Scene Simulator, an engineering tool that is able to model and evaluate countermeasures in such a way that the results could be used to make recommendations for successful deployment and use.

The use of formal ontologies is no longer a foreign concept in the support of information systems. To assist with the simulations performed in the Optronic Scene Simulator, an ontology, Simtology, was developed. Simtology supports the system in various ways such as providing a shared vocabulary, improving the understanding of the concepts in the environment and adding value by providing functionality that improves integration between system components.

Keywords: Ontology, Countermeasure, Simulation, Design Research

1 Introduction

Military forces consider aircraft as important and expensive assets often representing huge investments. The protection of these assets is considered to be a priority by most countries. Protection is needed from various threats and one of these threats are attacks through enemy missiles such as surface-to-air missiles, which are relatively cheap and easy to operate, and in addition, widely available in current and old war-zone areas [1]. These surface-to-air missiles are often complex and they are continually being updated to withstand aircraft countermeasures. In addition, missile systems differ from one another and the need to understand how each type of missile reacts in an aircraft engagement is crucial in the development of aircraft protection countermeasures[1]. The development of these countermeasures is often not possible in real-life situations, and modelling and simulation are therefore necessary for the development of aircraft protection countermeasures. Figure 1 illustrates a military aircraft ejecting a flare, which is a specific type of countermeasure used to protect against missile attacks.

Fig. 1. Countermeasures are implemented on aircraft to protect against missile attacks. Aircraft Ejecting Countermeasures Flares (www.aerospaceweb.org)

Simulation systems model real-world objects and simulate them in an artificial world [2]. One such a simulation system is the Optronic Scene Simulator (OSSIM), which has an application called the Countermeasure Simulation System (CmSim). CmSim uses models of real world objects such as the aircraft and the missile, and simulates different scenarios to evaluate the behaviour of these models under different circumstances [2]. Often these evaluations require substantial computing power and it is not uncommon to wait a few hours for simulation results.

At present, various problems are experienced when constructing the input models for CmSim simulations. Because models are used as input into CmSim simulations, it is necessary to ensure that these models are adequate and accurate for useful simulations. The input model is adequate when it captures sufficient input variables and context, and a model is accurate when it correctly captures the input variables and relations. It is for example possible to create input models that are syntactically correct, but the interaction between the models are not correctly set up in the simulation and therefore the results have no correlation with the real world. In addition, different users with various roles work with the system and it is necessary to acquire a common understanding and vocabulary for the constructs of the models and their characteristics. Furthermore, the creation of reference models for reuse across the user base would ensure better use of resources and time.

When investigating possible technologies that support modelling within information systems, ontologies and ontology technologies feature extensively. One of the original definitions for the term *ontology* is that by Gruber who defined an ontology as a formalisation of a shared conceptualisation [3]. A formal conceptualisation is a representation in a formal language of the concepts in a specific domain representing a part of the world. Formal ontologies are therefore ontologies constructed using a formal representation language such as Description Logics (DL) [4][4]. *Ontology* is also used as a technical term denoting an artefact that is designed for the specific purpose of modelling knowledge about some domain of interest. Typically a domain ontology provides a *shared and common* understanding of the knowledge in the chosen domain [5]. Given the characteristics and purpose of ontologies, we decided to investigate the use of an ontology to address the identified needs when constructing CmSim Models.

The remainder of this paper is structured as follow: Section 2 provides some background of the simulation environment and why it was necessary to build an ontology, as well as some background on ontologies. Section 3 discusses the development and nature of Simtology. Sections 4 and 5 discuss the contribution and conclude the paper in addition to discussing future work, as well as possible extensions to the ontology.

2 Background

One of the largest investments in the military of a country is aircraft. Aircraft is the target of unfriendly forces in order to weaken the military forces of a country. These attacks include attacks executed by shoulder-launched missiles, which are portable, easy to use and relatively easy to acquire. In order to counter these missile attacks, the military deploy various kinds of countermeasures on aircraft, and these countermeasures are developed, evaluated and deployed with the assistance of modelling and simulation systems such as the Optronic Scene Simulator (OSSIM).

2.1 The Simulation System Environment

CmSim is a software application that is part of the broader Optronic System Simulation (OSSIM) system [2]. OSSIM is an engineering tool used to model and evaluate the imaging and dynamic performance of electro-optical systems under diverse environmental conditions. OSSIM are typically used for the following applications:

- Development of optronic systems
- Mission preparation

[4] For the remainder of this paper we mean *formal ontology* when we use the term *ontology*

– Real-time rendering of infra-red and visible scenes

CmSim is specifically designed to do countermeasure evaluation for the protection of military aircraft. Models of the aircraft, the missile, the countermeasure and the environment are used to construct a scenario that simulates what will happen in the real world [2]. The models are used as input into CmSim, and it is necessary to carefully construct these models to get accurate simulations results. The generation of simulation results are complex and time consuming, and when inaccurate or faulty input models are used, valuable time is lost.

In order to construct better input models, it is necessary to improve the understanding of the simulation and the meaning of concepts in the simulation environment. Users of models often does not know what models exist already, to what level the models were constructed, and the scenarios that might be possible in the simulation, and knowledge is not shared between different role-players. The simulation practitioner setting up the simulation scenario might not have specialist knowledge of how the models interact, and can set up scenarios that are syntactically correct but do not correlate with the real world scenario. There is therefore a need to capture the specialised knowledge in reference models that could be used before the scenario is fed to the simulation. Figure 2 depicts the different role-players that could be involved in the simulation environment.

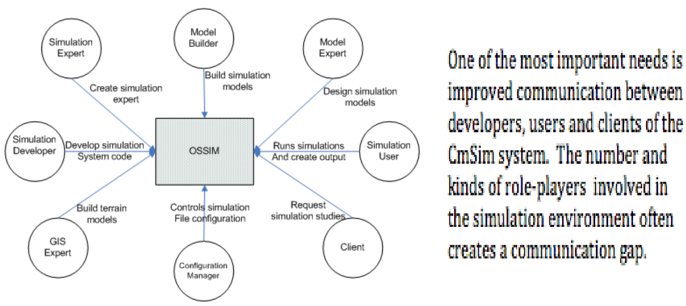

Fig. 2. Different Role-players Involved in a Simulation Environment

In order to address the above mentioned needs, we initiated a project based following the guidelines of design science research (DSR) [6]. DSR provides a research method for research that is concerned with the design of an artefact that solves an identified problem. The creation of an ontology based application was identified as a possible solution to the needs articulated when constructing OSSIM simulation models. DSR will be described further in Section 3.1. The next sections briefly introduce background on ontologies in computing.

2.2 Ontologies and Ontology Tools

The origins of the term *ontology* could be traced to the philosophers of the ancient world who analysed objects in the world and study their existence [3]. Modern ontologies use the principles of the ontology of early philosophers [7]. Ontologies formally describe concepts, so it is often used to capture knowledge of a specific domain. The role of ontologies in a specific domain are thus generally defined by [5] as to:

- Provide people and agents in a domain with a shared, common understanding of the information in the domain;
- Enable reuse of domain knowledge;
- Explicitly publish domain assumptions;
- Provide a way to separate domain knowledge from operational knowledge; and
- Setting a platform for analysis of the domain knowledge.

From the characteristics listed above it is possible to argue that an ontology may be a solution to the problems experienced in OSSIM simulations. Formal ontologies are represented in a specific formal knowledge representation language [4]. For building and maintaining Simtology, we adopted Protégé 4 constructing an OWL ontology. [8, 9]. Protégé is widely used and support for the use of the editor and the development of ontologies are readily available [10–12]. Protégé bundles reasoners such as Fact++ and Pellet with the environment [9, 13] and we used these reasoners to test for consistency or to compute consequences over the knowledge base during the development of Simtology [14].

2.3 Ontology Use in Modelling, Simulation and Military Systems

Within computing, modelling and simulation are used to build a representation of the real world and simulate the behaviour of objects presented in the models [2]. A simulation system is a specific application that uses a model as input and execute a computer program that determines consequences and resulting scenario information about the system being investigated [15].

Military systems and the knowledge captured therein are complex and often consist of layered information from different sources. To support this view, Clausewitz, in his book, *On War*, wrote about military information as follow [16]:

> '...three quarters of the information upon which all actions in War are based on are lying in a fog of uncertainty...'
> '...in war more than any other subject we must begin by looking at the nature of the whole; for here more than elsewhere the part and the whole must always be thought of together...'

Furthermore, Mandrick discussed the use of ontologies to model information in the military environment. According to Mandrick, ontologies in the military must adhere to the same requirements as ontologies in other domains, as described in Section 2.2. Important aspects to highlight is the ability of the ontology to provide a common vocabulary between planners, operators and commanders in the different military communities [16].

At present the adoption of ontologies in the military domain is primarily for support of command and control in the battlefield, as well as the management of assets and the sharing of knowledge[11, 17]. We also find ontologies where there is a need to integrate different data sources and the communication between these data sources [18, 19]. Although ontologies are used in the military modelling and simulation domain, examples are sparse and at present do not support the construction of input models for systems such as OSSIM. It could be argued that Simtology will therefore present a unique contribution to military information management.

3 Simtology

The development of Simtology was in response to the identified needs when using the Optronic Scene Simulator (OSSIM) [2] to develop countermeasures for missile attacks on aircraft as discussed in Section 2.1.

3.1 The Design and Development Process

The research design adopted for the development of Simtology, was Design Research (DSR). DSR is a research methodology for the design and construction of computing artefacts through the use of *rigour* (the use of fundamental knowledge) and *relevance* (basing the development of the artefact on real needs) [6, 20]. In this project, the artefact is Simtology, the fundamental knowledge is obtained from ontology knowledge, and the need is the construction of models for the OSSIM simulation environment. A DSR execution method that was proposed by Vaishnavi et al.[21] is depicted on the left in Figure 3. This method was adopted for this research project, and the development of Simtology is discussed further according to the steps in Figure 3.

3.2 Awareness and Suggestion

The first steps in the design research process is *awareness of the problem* and *proposing possible suggestions* for a solution. The following list summaries the issues and needs in the simulation system as discussed in Section 2.1 that created *awareness of the problem*:

- Different role-players: There are developers, model builders and users involved in the system. Inconsistencies in the terminology used between different users often led to frustration and wrong use of concepts. There is lack of a common vocabulary that is shared by everyone involved in building and using the system.

- Model complexity: One of the main characteristics is the ability of the system to execute models at different levels of detail. This poses a problem to users, when to know at which level of detail a model is implemented at.
- Reference models: Specific users that only interact with the system at a certain level, need more technical insight into model detail to know what is available in the system. This means that reference models are required that can define domain-specific concepts to these users.
- Model Interaction: A simulation consists of a scenario that is built up from interacting models. The models interact using a set of rules but there is currently no rules that verify model behaviour when a scenario are constructed.

Previous research efforts in the simulation environment attempted to address the the need for a standard notation for documentation of the simulation models. This proved to be problematic and one of the suggestions for further research was to investigate the use ontologies in the simulation environment. The *suggestion* according to the DSR process is therefore that a formal ontology is created to address the above mentioned needs for the simulation environment.

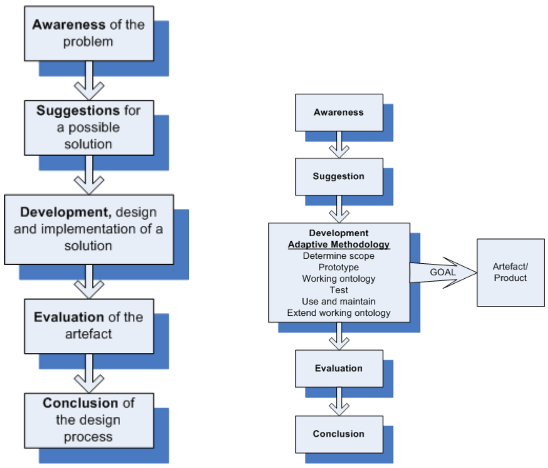

Fig. 3. The Design Research Process on the left, and the Adaptive Methodology Process on the right

3.3 Development

The ontology was build using the approach followed by the researchers who develop the Adaptive Methodology [22], and was chosen for its lightweight, incremental approach. Figure 3 depicts the development process steps as well as the alignment with the design process.

- **Scope and Purpose:** The first step is to scope the purpose and the extent of the ontology. In the case of a domain ontology, the concepts of the domain must be included. It is not necessary to include all the concepts of the domain. The level of detail will be determined by the purpose of the ontology.
- **The Use of Existing Structures:** There are several documents, structures and sources available in the OSSIM simulation environment available to use in order to gather information to develop the ontology and to, for example, make a list of the concepts in the simulation. Modelling reports, installation guides, white papers and technical documentation, as well as the source code of the system and the documented test point configurations were used as input into the ontology development process.
- **The Prototype:** The prototype structure is the first version of the ontology that is operational. The prototype for the simulation environment contains only a selected set of components from the domain. The concepts are on a high level and the nested structures of complex concepts were not included in the prototype. The prototype was developed in Protégé and is illustrated in Figure 4 on the left.
  The prototype is a proof-of-concept and in this project it played an important role to demonstrate the feasibility of the suggested solution. The prototype ontology supported the role of an ontology in the simulation system environment, and supported an ontology as a solution to a shared, common vocabulary. The tools also provided graphical views of the concepts defined in the ontology.
- **Development of the Working Ontology:** During this phase the prototype ontology was expanded by adding concepts from the domain not previously included in the ontology, as well as developing new functionality. The working ontology contains a full set of domain concepts that describe the simulation models and model properties and is called Simtology. The next section describes Simtology in more detail, as well as how Simtology is used in CmSim.

3.4 Description of Simtology

To do a simulation in CmSim, a scenario must be set up to act as input to the simulation. The scenario consists of various configuration files but the main file is the scenario file itself, which contains links to all other files necessary to describe a scenario and the components in it. Although the prototype already contained enough information to set up basic scenarios, Simtology contains all the concepts in the domain of CmSim.

The first task was thus to expand the prototype to present not only the basic objects, but all the possible objects in the CmSim domain. The classes and properties were expanded. The following list describes the concepts and properties defined in Simtology.

- **Concepts:** In Simtology, an example of a concept representing all the individual aircrafts modelled in the simulation environment, is `Aircraft`. Figure

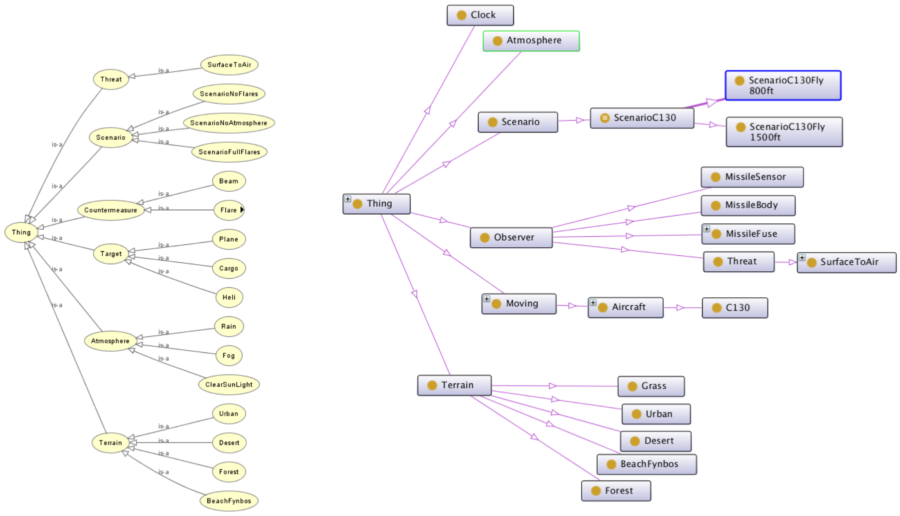

Fig. 4. Concepts from the prototype ontology on the left, and concepts in the final version of Simtology on the right

4 depicts an extract of the top-level concepts defined in Simtology, where the concepts were selected to present those in a simulation scenario. The main concepts in Simtology are the Moving, Observer and Scenario concepts. The choice of concepts relied heavily on the structure of the simulation configuration files. Therefore objects of type Moving have specific behaviour in the simulation and belong together in a concept.

- **Individuals:** Individuals are asserted to be instances of specific concepts. Specific scenarios can be build by choosing individuals from the ontology and thus creating an individual scenario.
  `ScenarioC130` is an individual of the `Scenario` concept that uses a specific type of aircraft.
- **Object Properties:** Object properties are used to link individuals to each other. In Simtology, a scenario must have a clock object, so having a clock object is an object property of the scenario concept. The name of the property is “hasClock“ and links an individual of class `Clock` to an individual from class `Scenario`.

  In Simtology the main object properties belong to a scenario. The following properties are sufficient to denote a valid scenario that can run in a simulation: `ScenarioC130` hasClock `Clock10ms`
  `ScenarioC130` hasTerrain `TerrainBeachFynbos`
  `ScenarioC130` hasMoving `C130Flying120knots`
  `ScenarioC130` hasObserver `SAMTypeA`
  `ScenarioC130` hasAtmosphere `MidLatitudeSummer`

– **Data Properties:** Data properties were added to Simtology to add data to individuals. Examples of data properties are geometric locations of moving objects, or data belonging to the class `Clock`, as depicted below:
`Clock10ms` hasInterval `10ms` and `Clock10ms` hasStopTime `10sec`

Functionality: A scenario can be complex and rules were built in to ensure that a valid scenario is constructed, for instance only certain types of flares can be used as a valid countermeasure on a specific aircraft. After building the scenario in the ontology, the scenario can be processed by a reasoner. The reasoners are used to compute the inferred ontology class hierarchy and to do consistency checking after a scenario is created.

Additional functionalities were developed for use with Simtology such as the integration of the ontology with the graphical user interface (GUI) used to set up the simulation. The ontology is used to populate the elements in the interface, resulting in several advantages such as that only one source of simulation information has to be maintained, as well as that the ontology can be used to change the language displayed in the GUI .

Functionalities were also developed to write out scenarios created in the ontology to files that can act as input to the simulation. This made it possible that a scenario can first be checked for logical correctness before it is run in the simulation. Modelling errors not handled by the simulation software are handled early in the simulation process by using the reasoning technology in the ontology. By having a scenario defined in the ontology, it is possible to export a high-level description of a scenario and its components to be used for reporting and documentation of simulation studies.

Testing of Simtology Testing the ontology was an important step towards creating a useful Simtology. When an ontology is small with a few concepts, it is easier to identify modelling problems but when there are large numbers of concepts with complex relationships, it is important to test the ontology regularly in order to avoid inconsistencies immediately and eliminate rework. During ontology verification the focus was mainly to ensure that the ontology was built correctly and that the ontology concepts match the domain it represents. The test phase of the ontology is part of the adaptive methodology process and this phase complements the evaluation phase of the design research process.

4 Evaluation

In Section 3.2, the simulation system environment was discussed. In order to evaluate the use of Simtology in the simulation system and the contribution it has for the improvement of the environment, the issues mentioned in Section 3.2 are used as evaluation criteria. The identified issues are 1) different users; 2) model complexity; 3) reference models; and 4) model interaction. When evaluated against the identified issues, Simtology provided the necessary solutions.

- **Different users:** Simtology provided a common, shared 'language' to assist with eliminating ambiguities and the inconsistent use of terminology by the different users of the system. The feedback by all concerned users was positive. An example of how Simtology assisted with regards to a common understanding is in the use of ambiguous terms. Some terms in the simulation had different meanings depending on the user using it and the application it was used for. An example of such a term is *countermeasure*, which was vague and previously many different types of countermeasures existed. In Simtology the concept `Countermeasure` was defined in such a way that it can be used as an explanatory tool to illustrate the different countermeasures available in the simulation as well as the use of each countermeasure.

  The Protégé editor allows for several ways to communicate the ontology, for example a graphical display of the concepts and the relations in the ontology A visual display of the different components in the simulation leads to better communication between all the people involved.
- **Model complexity:** Simtology formally defined the concepts, properties and individuals necessary for the construction of input models. When a user uses Simtology to construct her input model, the appropriate level of detail and complexity of the input models are specified.
- **Reference models:** Simtology provides a reference model for all the different users of the system to create their specific input models from. After introducing Simtology, very few problems were experienced by users when constructing simulation models because Simtology acted as a reference model informed their specific model design.
- **Model Interaction:** A simulation consists of a scenario that is build up from interacting models. Simtology provides a common shared language to be used in the simulation environment for both users and when interacting with other applications. The definitions of concepts in the system are kept in Simtology and made available to applications in the environment such as the Graphical User Interface.

  As a final evaluation, the guidelines proposed by Hevner et al. [6] for a design research artefact were used to evaluate and present the research process followed to develop Simtology. This discussion is outside the scope of this paper but it was demonstrated that the construction of Simtology followed the proposed guidelines.

5 Conclusion and Future Work

The outcome of the research project was Simtology, a domain ontology for the simulation environment that contains the model information for simulation scenarios. An added benefit was that the process to analyse the contents of the simulation environment to construct the ontology clarified the knowledge in the domain.

During the construction of Simtology, the following observations were made:

- With regards to modelling, it is important to distinguish part-of from subclass-of. An aircraft body is part of an aircraft, not part of a specific type of aircraft.
- It is important to correctly model roles. Modelling a missile as an *observer* in the simulation means that it can never be used in the simulation as an *object of type moving*. In Simtology, a missile can therefore never be used in a different role.
- Another important modelling decision has to do with the modelling of individuals vs. concepts. This decision has an impact on how the ontology could ultimately be used. The choice between concept and individual is often contextual and application-dependent but it needs to be evaluated in one of the development cycles.
- The development and use of the ontology should be an iterative process. As new functionality is added, it must be tested, used and evaluated. Existing functionality is maintained by making changes where necessary. Proper version control is therefore also necessary when constructing ontologies.

Several advantages of having an ontology in the simulation environment emerged after the ontology was created. The ontology can, for instance, be used in training exercises to show aircraft personnel the technical detail of the countermeasures deployed on the aircraft. Furthermore, the simulation environment is always expanding and improving through the addition of new models, the addition of new properties to existing entities in the system or through the addition of new functionality to entities. Future versions of the ontology need to incorporate these changes and there should therefore always be future expansions to the Simtology ontology. Furthermore, Simtology should ideally be expanded to not only include concepts in CmSim, but also in the Optronic Scene Simulator. One of the planned functions to be developed is to reverse engineer previously run simulations and add the scenario descriptions of those simulations to the ontology.

References

1. Birchenall, R.P., Richardson, M.A., Butters, B., Walmsley, R.: Modelling an infrared man portable air defence system. Infrared Physics & Technology (July 2010)
2. Willers, C., Willers, M.: Ossim: Optronics scene simulator white paper. Technical report, Council for Scientific and Industrial Research (2011)
3. Gruber, T.R.: A translation approach to portable ontology specifications. Knowledge Acquisition **5**(2) (1993) 199–220
4. Baader, F., Horrocks, I., Sattler, U.: Description logics as ontology languages for the semantic web. In Hutter, D., Stephan, W., eds.: Mechanizing Mathematical Reasoning: Essays in Honor of Jörg H. Siekmann on the Occasion of His 60th Birthday. Volume 2605 of Lecture Notes in Artificial Intelligence. Springer-Verlag (2005) 228–248
5. Noy, N.F., McGuiness, D.L.: Ontology development 101: A guide to creating your first ontology. Technical report, Stanford Knowledge Systems Laboratory (2001)

6. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design Science in Information Systems Research. MIS Quarterly **28**(1) (2004) 75–105
7. Guarino, N.: Formal ontology and information systems. In Guarino, N., ed.: Proceedings of the First International Conference on Formal Ontologies in Information Systems (FOIS-98), June 6-8, 1998, Trento, Italy, IOS Press, Amsterdam, The Netherlands (1998) 3–15
8. OWL: Owl2 web ontology language. Available at http://www.w3.org/TR/owl2-overview. [1 April 2011]
9. Protege: The protege ontology editor. Available at http://protege.stanford.edu. [13 April 2011]
10. Gáevic, D., Djuric, D., Devedíc, V.: Model Driven Engineering and Ontology Development. 2. edn. Springer, Berlin (2009)
11. Schlenoff, C., Washington, R., Barbera, T.: An intelligent ground vehicle ontology to enable multi-agent system integration. In: Integration of Knowledge Intensive Multi-Agent Systems. (2005) 169–174
12. Nagle, J.A., Richmond, P.W., Blais, C.L., Goerger, N.C., Kewley, R.H., Burk, R.K.: Using an ontology for entity situational awareness in a simple scenario. The Journal of Defense Modeling and Simulation: Applications, Methodology, Technology **5**(2) (2008) 139–158
13. Tsarkov, D., Horrocks, I.: FaCT++ description logic reasoner: System description. In: Proc. of the Int. Joint Conf. on Automated Reasoning (IJCAR 2006). Volume 4130 of Lecture Notes in Artificial Intelligence., Springer (2006) 292–297
14. Bock, J., Haase, P., Ji, Q., Volz, R.: Benchmarking OWL Reasoners. CEUR Workshop Proceedings (June 2008)
15. Benjamin, P., Patki, M., Mayer, R.: Using ontologies for simulation modeling. In: Proceedings of the 38th conference on Winter simulation. WSC '06, Winter Simulation Conference (2006) 1151–1159
16. Mandrick, L.B.: Military ontology. http://militaryontology.com/
17. Valente, A., Holmes, D., Alvidrez, F.C.: Using a military information ontology to build semantic architecture models for airspace systems. In: Aerospace Conference, 2005 IEEE. (March 2005) 1–7
18. Winklerova, Z.: Ontological approach to the representation of military knowledge. Technical report, Military Academy in Brno, Command and Staff Faculty, Czech Republic (2003)
19. Smart, P.R., Russell, A., Shadbolt, N.R., Shraefel, M.C., Carr, L.A.: Aktivesa. Comput. J. **50** (November 2007) 703–716
20. Hevner, A., Chatterjee, S. In: Evaluation. Volume 22 of Integrated Series in Information Systems. Springer US (2010) 109–120
21. Vaishnavi, V., Kuechler, W.: Design research in information systems. Available at http://desrist.org/design-research-in-information-systems/ Last Updated 16 August 2009 (January 2004)
22. Bergman, M.: A new methodology for building lightweight, domain ontologies. Available at http://www.mkbergman.com/908/a-new-methodology-for-building-lightweight-domain-ontologies/ (2010)