ONTOLOGIES AND TOOLS FOR ANALYZING AND SYNTHESIZING LVC CONFEDERATIONS

Reginald Ford
David Martin
Daniel Elenius

SRI International
333 Ravenswood Ave
Menlo Park, CA 94025, USA

Mark Johnson

SRI International
4119 Broad Street
San Luis Obispo, CA 93401, USA

ABSTRACT

Establishing and maintaining interoperability among heterogeneous systems is a major challenge and expense for large business and military projects. Data interoperability and service-oriented architecture (SOA) approaches, while essential, do not provide a complete solution. We describe a complementary approach that uses Web Ontology Language (OWL) and Semantic Web Rule Language (SWRL) to capture information about the roles and capabilities required to complete a task, and the attributes of candidate resources. Our toolset applies automated reasoning to determine whether each candidate resource has the requisite capabilities and is compatible with other resources. If there are multiple candidates for a role, the reasoner ranks the relative goodness of each with respect to constraints and metrics that are appropriate for the specific task needs of the exercise or deployment. We also describe a further application of the ontologies and toolset to assist in the creation of composable data exchange models.

1 INTRODUCTION

1.1 The Problem

Much hard work and ingenuity has gone into making heterogeneous Modeling and Simulation (M&S) and live instrumentation systems work together in complex live-virtual-constructive (LVC) federations. But the level of interaction achieved may be relatively superficial (e.g., a common operating picture), and the gains fragile. The points of interaction among systems are myriad and often involve subtle nuances. A successful federation of systems may be forged using one or more of the extant interoperability architectures, but if one changes a little here, a little there, it all comes unglued.

One of the fundamental impediments is the lack of standard, unambiguous, and accessible descriptions of those nuances and the possible consequences of a mismatch. There is a growing recognition that nascent Semantic Web technologies hold considerable promise in overcoming this kind of problem, although practical results to date are relatively sparse. Another impediment is the lack of automated tools that are capable of transcending the naturally limited human ability to process voluminous and intricate information.

This paper is about a conceptual approach for expressing detailed and precise information about systems and their interoperability context, and an automated toolset that applies a reasoning engine to draw conclusions about interaction issues and opportunities. Although the Open Netcentric Interoperability Standards for Training and Testing (ONISTT) approach and the toolset were developed to assist planning for improvisational LVC training and testing events, they are very general in design and could be applied to a wide variety of domains.

1.2 Related Work

ONISTT builds on an extensive foundation of related work. For more than two decades, activities within the M&S community have pioneered the art of connecting disparate systems in temporary lash-ups to provide a desired set of capabilities that no single system could provide. Although many successful improvisational LVC federations have been built and employed, these successes have typically required considerable event-specific effort. Despite a number of activities to define standard data

models, communications mechanisms, and integration processes, routine success has been elusive. Considerable research has been devoted to finding the root causes for this shortcoming.

Dahman (1999) introduced the notion that the success of improvisational LVC confederations requires two distinct kinds of interoperability, termed "technical" and "substantive". Tolk and Muguira (2003), and later Turnitsa (2005), extended Dahman's decomposition to five- and six-level models (respectively) within a hierarchical structure called the Levels of Conceptual Interoperability Model (LCIM). While many interoperability communication architectures have addressed the first two LCIM levels (i.e., technical and syntactic), the ONISTT ontology-based framework also addresses the upper four levels (i.e., semantic, pragmatic, dynamic, and conceptual). See (Tolk et al. 2008) for a discussion of the relationship between LCIM and Semantic Web technologies.

ONISTT has also been influenced by recent DoD initiatives to develop a framework for building systems that can be networked together to provide improvisational capabilities, that is, capabilities that were not initially defined for the constituent systems at the time of their construction. These initiatives include the Net-Centric Data Strategy (NCDS), Net-Centric Operations and Warfare (NCOW) (DoD CIO 2009a), the DoD Metadata Registry (MDR) (DoD CIO 2009b), and the NATO Net Enabled Capability (NNEC) (Director, IS & NNEC ICT 2009). These initiatives are based largely on the creation of online accessible metadata, and employing the tenets of Service Oriented Architecture (SOA) (OASIS 2006).

Kasputis et al. (2004) discuss the use of semantic descriptors for models and simulations. The thesis is that a well-structured system of semantic descriptors can be used to assess the validity of a simulation federation — and can also define the context under which that validity would hold: "If the same language was used for both simulation requirements and a description of model capability, the groundwork would be in place for development of automated user-defined composable simulations." Vick et al. 2006 introduce the same basic idea. This is essentially the notion upon which the ONISTT project is based, except that ONISTT is more broadly concerned with composable resources of all types (including live and virtual systems in addition to constructive systems).

A related set of problems has been the focus of inquiry in the context of research on Semantic Web Services. This field, which aims to enable the automation of the development and use of Web services, takes as its first challenge the enrichment of Web service descriptions. For example, Web Ontology Language for Services (OWL-S) (Martin et al. 2007), the pioneering effort in this field, introduces the expression of preconditions and effects in a Semantic Web-compatible manner, and also relies on the ability to use OWL to construct class hierarchies of services (OWL and other Semantic Web technologies are also used in ONISTT, as discussed in Section 3). Based on such descriptions of services, a variety of approaches have been devised for the selection and composition of services. However, service-based selection and composition rely on descriptions of interfaces and behavior, whereas ONISTT is principally concerned with descriptions of capabilities, which we view as a higher level of abstraction in portraying systems.

Several papers describe tool suites that apply OWL's combination of machine-understandability and rich semantic expressiveness in ways that resemble aspects of the ONISTT framework described herein. Preece et al. (2007) apply ontologies and prototype software to determine which ISR assets (e.g., type of UAV and sensors) are best suited to a particular mission, which may include constraints such as current weather conditions. Alternative solutions are ranked. Silver, Hassan, and Miller (2007) describe an Ontology Driven Simulation (ODS) design tool that maps concepts from domain ontologies to the Discrete Event Modeling Ontology (DeMO), then translates the resulting ontology individuals to an intermediate XML representation, and finally generates an executable simulation model (see also Miller and Baramidze 2005). The tool suites described in Preece et al. (2007) and Silver, Hassan, and Miller (2007) share with ONISTT the ability to draw on heterogeneous domain ontologies that may have significant differences in level of abstraction, inconsistent taxonmic classification, and the like.

2 ONISTT OVERVIEW

2.1 The ONISTT Methodology

The ONISTT methodology is shown in Figure 1. The top half of the figure shows the ONISTT knowledge capture phase. The essential products of this phase are knowledge bases (KB) shown in the middle of the figure (3). These KBs are expressed in terms of concepts formally described in ontologies (2). The "referents" used for the ontologies are the most accurate and authoritative information available about the real-world environments, tasks, infrastructures, and systems that are relevant to the problem (1).

The bottom half of the figure shows the ONISTT employment phase. Using a graphical user interface (GUI), the human planner defines the objectives and constraints of an exercise and proposes a partial or full confederation of participants, systems, and infrastructure (4). To determine whether a proposed confederation satisfies the interoperability needs of the
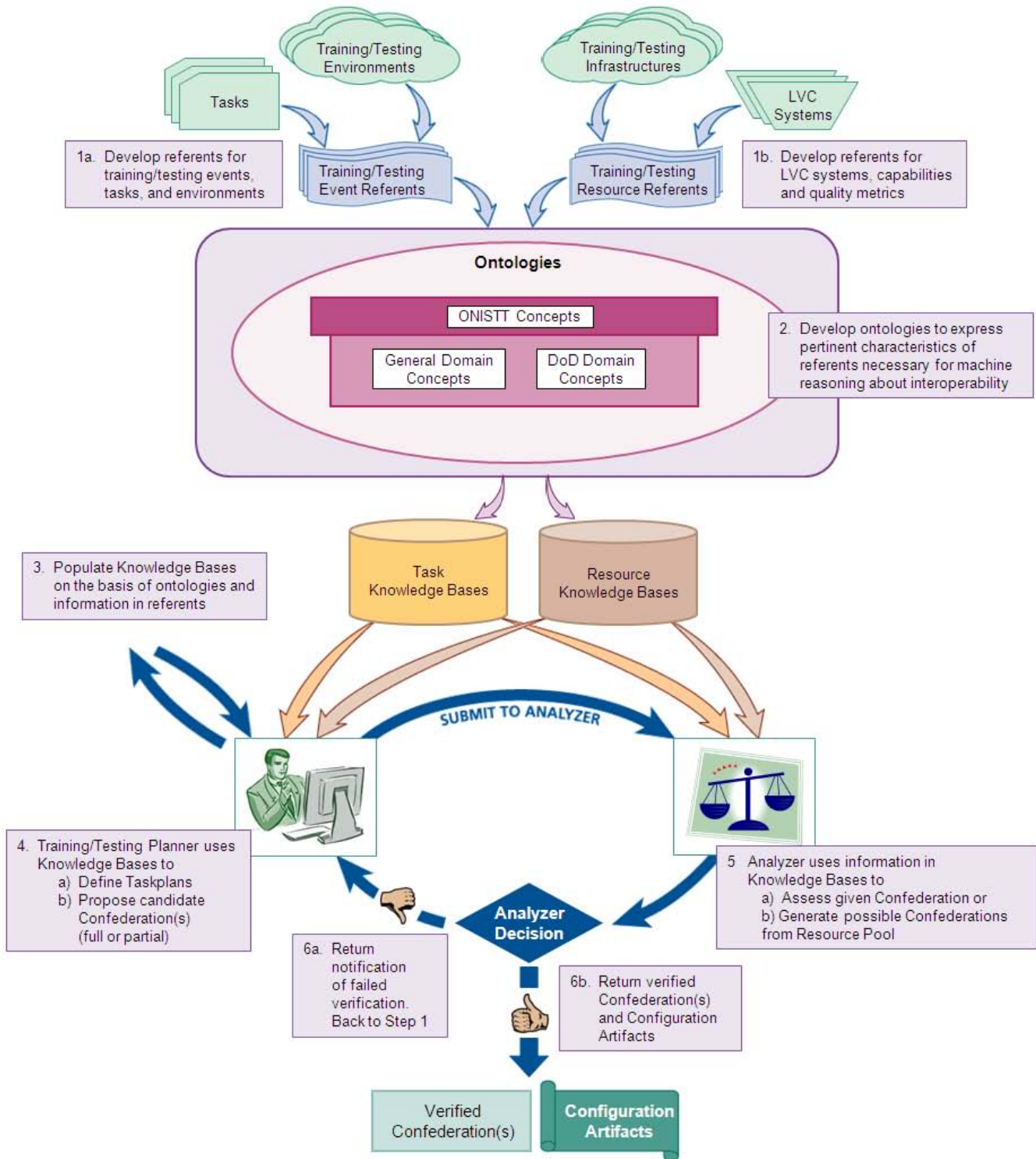
Figure 1: ONISTT Knowledge capture and Analyzer employment phases

specified event, the Analyzer applies domain-specific interoperability rules, general reasoning technology, and facts captured

in the KBs (5a). If the planner leaves some of the confederation assignments blank, the Analyzer selects and ranks candidate systems (5b).

The Analyzer either warns the planner about potential interoperability problems (6a), or returns a verified confederation and configuration artifacts (6b). The Analyzer assigns a severity level to warnings. At this point the planner can submit a modified proposal, or decide that the level of interoperability is good enough for the purposes of the exercise. Since designing and verifying a complex confederation is normally an iterative process, the Analyzer allows the planner to focus on particular parts of the exercise and save results for use in building up the larger confederation. Analysis results can also include artifacts such as specification of settings for configurable resources that are appropriate for the proposed deployment context. The long-term goal for ONISTT envisions hosting the KBs on the DoD MDR and implementing the analysis/synthesis function as a Net-centric Enterprise Service.

At the heart of the ONISTT approach is a collection of descriptions of the capabilities of *resources*, which can include all kinds of LVC systems, and the activities, or *tasks*, in which the resources will be used in test and training events. These descriptions are based on ontologies and are stored in knowledge bases. Given these descriptions, the Analyzer can evaluate the suitability of the resources for those tasks, analyze the ability of the resources to interoperate successfully to carry out those tasks, and perform automated planning of some aspects of test and training events.

2.2 An Illustrative Use Case

Interoperability analysis is conducted in the context of matching specifications of *capabilities needed* (by tasks) against specifications of *capabilities provided* (by systems or other kinds of resources). Capabilities of constructive and virtual systems are represented in much the same way as those of live systems, except that these systems usually have some additional capabilities with properties that are specific to the world of simulation.

An example is provided by an ONISTT use case in which the Analyzer evaluates the compatibility of various candidate simulation systems to interact in a movement to contact training task. The task includes a Blue Force (BLUFOR) armored maneuver role, some additional BLUFOR roles (e.g., scouts), and an opposing force (OPFOR) armored maneuver role. In this use case, the assignment of an M1A1 virtual training system to the BLUFOR maneuver role is fixed, and the OPFOR armored role and additional BLUFOR roles may be assigned to any of several candidate constructive simulations. A direct fire engagement task is associated with capabilities needed for tracking ground vehicles, including line-of-sight (LOS) visibility (for the shooter) and realistic ground vehicle motion (of the targets). In considering the ability of various combinations of systems to provide these capabilities, the Analyzer evaluates a rule stating that if there is an interaction (such as direct fire) requiring that a virtual system have LOS of moving entities, and the simulation fidelity of these entities might result in irregular motion, the motion may appear to be unrealistic to the users of the virtual system.

The KB for each candidate virtual or constructive system includes information about the fidelity of its entity motion simulation capabilities. One of the candidate constructive simulations was designed to handle large numbers of entities and was not intended to drive 3-D displays. The Analyzer issues a warning that the entities simulated by this system are prone to exhibit distracting behaviors (e.g., instantaneous sharp turns) when viewed by trainees in the virtual M1A1. However, if the training scenario called for largely static OPFOR entities, this unrealistic motion would not be problematic and no warning would be issued. Note that the Analyzer does not reject this candidate simulation out of hand, but generates a warning and leaves the final decision to the human exercise designer. Another of the candidates is a true Semi-Automated Forces (SAF) constructive simulation system. Since this system was designed to drive 3-D virtual displays, the Analyzer does not issue a warning related to unrealistic motion modeling.

Ford et al. (2007) provides additional detail about a demonstration of this use case, including evaluation of terrain modeling and communications compatibility.

2.3 Sponsorship

ONISTT is sponsored by the Office of the Deputy Under Secretary of Defense/Readiness/Readiness & Training Policy and Programs (ODUSD/R/RTPP). The first-generation version of the ONISTT ontologies and Analyzer was demonstrated in January 2007 (see (Ford et al. 2007)). The Analyzer for Netcentric System Test Confederations (ANSC) project was subsequently established by Under Secretary of Defense for Acquisition, Technology, and Logistics (USD/AT&L) Test Resource Management Center (TRMC) Test and Evaluation (T&E) Science and Technology (S&T) Program (USD/AT&L-TRMC S&T Portfolio) to develop a second generation ONISTT whose capabilities include support for the special needs of net-centric T&E event planning. The initial demonstration of ANSC capabilities was conducted in January 2009. US Joint Forces Command (USJFCOM) Joint Warfare Fighting Center (JWFC) and ODUSD/R/RTPP are jointly sponsoring the

Joint Training Enterprise Ontology Workshop (JTEOW) to bring subject matter experts (SMEs) into the process of defining referents, ontologies, and KBs. Current and planned ONISTT, ANSC, and JTEOW work is discussed in Section 6.

3   LANGUAGES AND TOOLS

To perform these kinds of analysis and synthesis, a firm foundation is needed for representing and reasoning about knowledge. In particular, one needs a representation language with support for defining concepts in ontologies, and with a well-defined semantics for reasoning about descriptions based on those ontologies. The semantics of such a language provides a mathematically precise foundation in which the conclusions that are reached are guaranteed to be valid (known as *soundness* of reasoning), and all valid conclusions are guaranteed to be reached (known as *completeness* of reasoning).

To meet these requirements, ONISTT employs OWL (McGuinness and van Harmelen 2004), which in turn is layered atop the Resource Description Framework (RDF) (Klyne and Carroll 2004). Both of these knowledge representation technologies have been standardized at the World Wide Web Consortium (W3C) in recent years, as part of the Semantic Web initiative. The Semantic Web initiative arose in 2000 as the result of a technology development partnership between DARPA and the W3C. OWL, a description logic language, is well suited to ONISTT's objectives of describing, classifying, and performing comparisons among categories of resources, capabilities, and tasks, based on their properties.

In addition to the functionality provided by OWL, the ability to express *constraints* and *rules* is needed. For this purpose, we make use of the Semantic Web Rule Language (SWRL). SWRL was designed to be used with OWL and allows for the expression of if–then rules similar in character to those used in the Prolog programming language. Although SWRL has not been standardized, it has a thorough, stable specification that has been submitted to the W3C (Horrocks et al. 2004).

A number of commercial and open-source tools and software libraries are available for use in working with OWL and SWRL. For editing and maintaining ontologies and knowledge bases expressed in these languages, we have primarily relied on the open source tool Protégé (Knublauch et al. 2004). Our use of Protégé, and our Prolog-based reasoning engine, are discussed further in Section 5.

In terms of Figure 1, Protégé is used in developing and maintaining ontologies (2) and knowledge bases (3), both of which are represented using OWL and SWRL. Resource and event referents (1) are expressed in a less formal manner, typically involving text documents and/or design documents expressed in the Unified Modeling Language (UML) (Booch 1993), or UML-based frameworks such as DoD Architecture Framework (DoDAF) (DoD CIO 2007). The most relevant UML diagram types include class and activity diagrams. In developing ontologies based on these referents, ONISTT leverages the Ontology Definition Metamodel (ODM) (Object Management Group 2009). ODM, recently standardized by the Object Management Group (OMG), supports ontology development and conceptual modeling in several standard representation languages, including OWL, and provides a coherent framework for visual ontology creation based on OMG's Meta Object Facility (MOF) and UML.

4   ONTOLOGIES

The ability of entities, systems, and infrastructure resources to interoperate often rests in very subtle interactions among their individual capabilities and the operational context. ONISTT interoperability analysis must be grounded in detailed, authoritative, and up-to-date KBs about simulators, training instrumentation, vehicles, communication architectures, terrain, training and testing events, and so forth. It is neither feasible nor desirable for most of these to be designed by or under the control of ONISTT developers. Section 4.1 describes some domain ontologies that are used by ONISTT but may also be employed for other purposes. Section 4.2 describes the very small set of core ontologies that make it possible for the domain ontologies to be marshalled for use in ONISTT interoperability analysis.

4.1 Domain Ontologies

We use the term *domain ontology* to indicate an ontology that is (or should be) developed and maintained by a recognized standards body or an organization that is responsible for a resource or is authoritative about a subject area. Domain ontologies range from fundamental scientific or engineering concepts to specific types of systems or resources. In most cases the ontologies we need do not yet exist. Since several domain ontologies are critical to ONISTT project objectives, the ONISTT team is forced to create our own implementations; we will use the official versions if and when they are made available. Whenever possible, our domain ontologies are patterned on available standards documents.

Our EngineeringValue ontology for physical quantities is an example of a fundamental domain ontology. EngineeringValue subclasses are *quantity types* like *Mass* or *Length*. EngineeringValue individuals contain a *magnitude* and a mandatory *unit*

of measure. Among the objectives of the ONISTT project is to reduce ambiguity to the greatest practical extent; most readers have probably encountered problems with unstated units of measure, or cases where units were not collocated with the relevant data. Every interesting simulation problem requires the use of physical quantities.

Also encoded in the EngineeringValue ontology are conversion factors to the unit of measure in the International System (SI) standard for each particular quantity type. (For example, *meter* is the SI standard unit of length; the conversion factor from *foot* to *meter* is 0.3048.) One or at most two applications of these conversion factors allow conversion between any supported pair of units. This practice is consistent with the approach taken by the National Institute of Standards and Technology (NIST) in Thompson and Taylor (2008). This ontology has been enhanced with SWRL rules that support the definition and usage of quantity *intervals* (e.g., "0 to 100 meters") and the comparison of quantities and quantity intervals.

Related to EngineeringValue is our EngineeringMeasurement ontology. This ontology is based on the practices documented in International Organization for Standardization (1995). EngineeringMeasurement individuals are built from an EngineeringValue representing the *measurand* and another EngineeringValue of the same quantity type representing the *uncertainty* in the measurement (typically one standard deviation). Acknowledging, documenting, and sharing measurement uncertainties can be critical in LVC confederations where Live simulations (such as instrumented players and platforms) are included.

It should be noted that these ontologies are used to document, analyze, and synthesize simulations systems and systems-of-systems and are not transmitted as part of the information that is exchanged repeatedly. Contents of simulation messages *on the wire* do not need to be as complete or as verbose. A simplified example is the case where OWL-based, machine-processable metadata declares that all Lengths will be exchanged as meters; the meter indicator can then be omitted from the wire message to reduce bandwidth consumption.

The domain ontologies and KBs developed by the ONISTT team are combined to provide standard, unambiguous, and ontologically "deep" descriptions of their concepts. For example, the quantitative concepts described in EngineeringValue and EngineeringMeasurement are referenced by a group of related ontologies based on the ISO 18026 Spatial Reference Model (SRM) standard (International Organization for Standardization 2006), including SpatialReferenceFrame, AbstractCoordinateSystem, ObjectReferenceModel, and ReferenceDatum. The SRM ontologies are used, in turn, by domain KBs describing various LVC interoperability communication architectures to identify the spatial reference frames that are associated with their time-space-position information (TSPI) object models or messages. Domain KBs descriptive of specific resource types (e.g., an EA-6B simulator) use the SRM ontologies to describe their native spatial information representation and also the communication architecture ontologies to describe how this information is published.

4.2 Core Ontologies

The ONISTT core ontologies (see Figure 2) are designed to support automated reasoning about the ability of a collection of *resources* to act in concert for a specific *purpose*, that is, to perform a specific set of tasks in a specific context. The root concept for that context is **Deployment**.

The key concepts describing purpose are on the left side of Figure 2. A **Task** is an intended action that requires some resource(s), and potentially has some additional constraints associated with it. A **Role** is a "slot" of a task that needs to be filled with some resource. A **TaskPlan** is a plan for how to perform a task, including assignment of resources to roles. The task and task plan ontologies and the use of SWRL to specify rules and constraints are described in more detail in Elenius et al. (2009).

The key concepts to capture details about resources are on the right side of Figure 2. A **Resource** is a thing that can *do* something, through the use of capabilities. Examples of resources are a tank, a human, a simulator, or training instrumentation. We also allow for intangible resources such as software systems. A resource can have subresources. A **Capability** is a discrete piece of functionality that belongs to a resource. Examples include the capability to send a "weapon fired" message using some message format, the capability to physically move (at some maximum speed), or the capability to detect things in the infrared spectrum. A **Confederation** is a set of resources.

Examples of extending these core ontologies are the ONISTT ontologies of capability types for military training and testing resources containing subclasses of the **Capability** class, such as **MovementCapability**, **DetectabilityCapability**, and **DirectFireCapability**. It is important to have a common capability ontology, because task descriptions will refer to these capability types. Task ontologies include variants of military **Task** types such as **TST** (time-sensitive targeting) and **JCAS** (joint close air support).
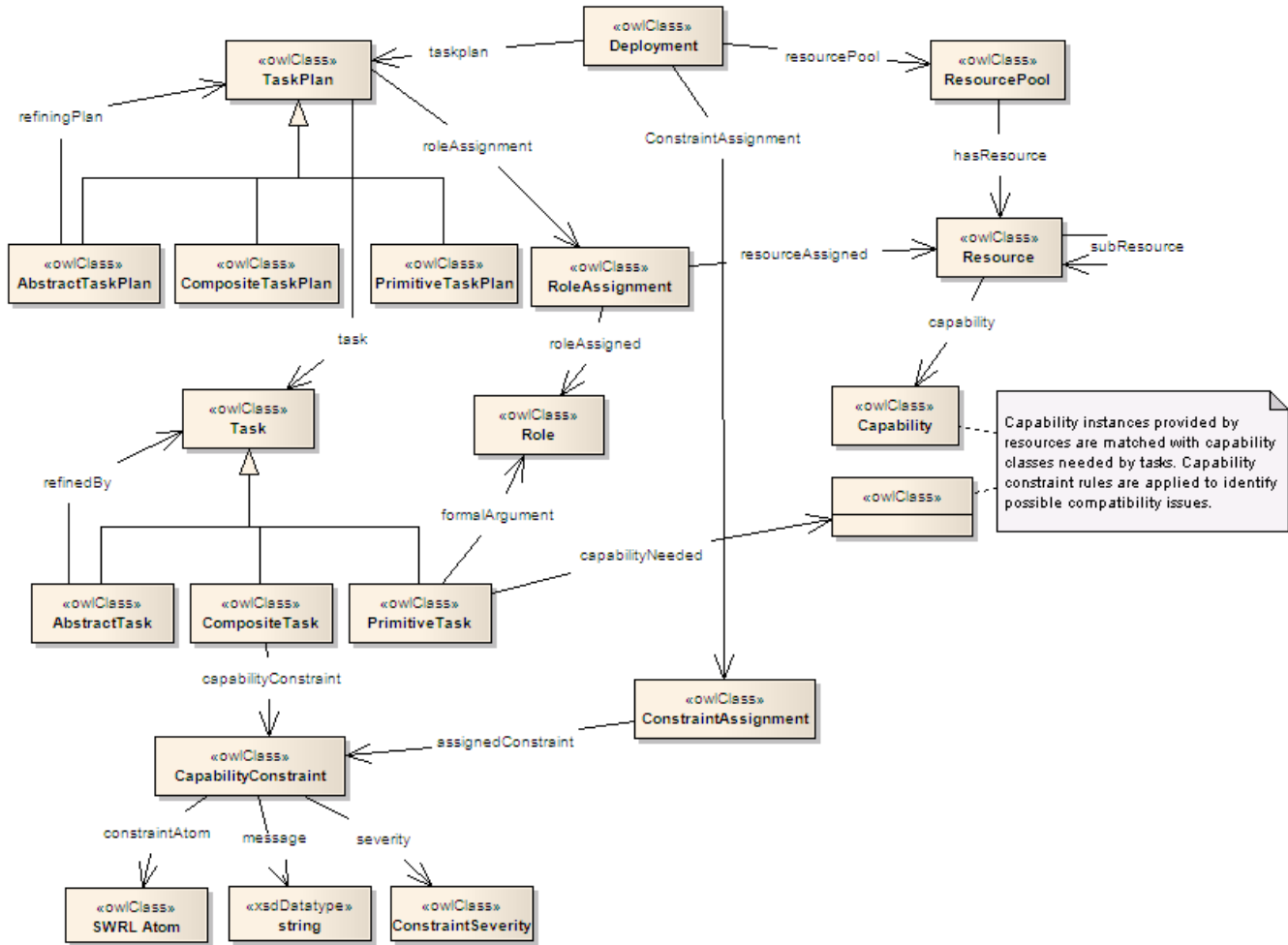
Figure 2: ONSITT core ontology

5   ANALYZER

This section provides a brief overview of our Analyzer tool. Although a Synthesis capability has been added to the tool since its original naming, we still refer to it as the "Analyzer". For additional technical details, see (Elenius et al. 2009).

While the ontologies provide the declarative knowledge of the problem domain, the Analyzer provides the computational side of the automation. The job of the Analyzer is to look at the information provided and draw conclusions according to a set of rules. More specifically, the Analyzer takes a (usually partial) task plan and returns a set of completed task plans that are compatible with the input task plan. In doing so, the Analyzer performs several functions:

- Analysis or synthesis of assignments of resources to roles in the task plan. If several resources fulfill the requirements of a role, different solutions are returned.
- Evaluation of constraints. Success or failure of constraints, along with the severity associated with the constraints, determine the goodness score of a solution.
- Evaluation of configuration artifacts. These are used to return additional information to the task planner, regarding the configuration of the assigned resources, or explanations of why certain constraints failed, as applicable.
- Selection of abstract task refinement. Our task ontology includes a notion of abstract tasks, which can be performed in several different ways. The Analyzer chooses among these options. This can give rise to multiple solutions, if several task refinements are possible given the assigned resources.

The components of the Analyzer tool are shown in Figure 3. The ontologies are stored in OWL files and loaded into the ontology development framework Protégé (Knublauch et al. 2004). A plug-in to Protégé connects the ontological knowledge with our reasoning tools. The plug-in contains a GUI that helps users create and navigate tasks and task plans, invoke the task engine (described below), and navigate and utilize results from the task engine.

The OWL KB is translated to a native Prolog representation. The translation uses the well-known correspondence of a large subset of OWL, called DLP (Description Logic Programs) (Grosof et al. 2003), to Horn clauses (the translation is described in detail in our previous work Elenius et al. (2007)). This means that not all of OWL's semantics are covered (i.e., the query answering is not complete), but in practice we have not found this to be a limitation for the ontologies with which we work, as we do not tend to rely on the more complex OWL axioms and the inferences that they would enable. The Interprolog library connects the Java and Prolog environments.

The Task Engine is the heart of the reasoning process. It is a custom Prolog program that analyzes and synthesizes task plans, evaluates constraints, and produces configuration artifacts. The engine is implemented in XSB Prolog `<xsb.sourceforge.net>`. Prolog was a natural choice because it provides built-in backtracking, which we use to generate all solutions during task plan synthesis.
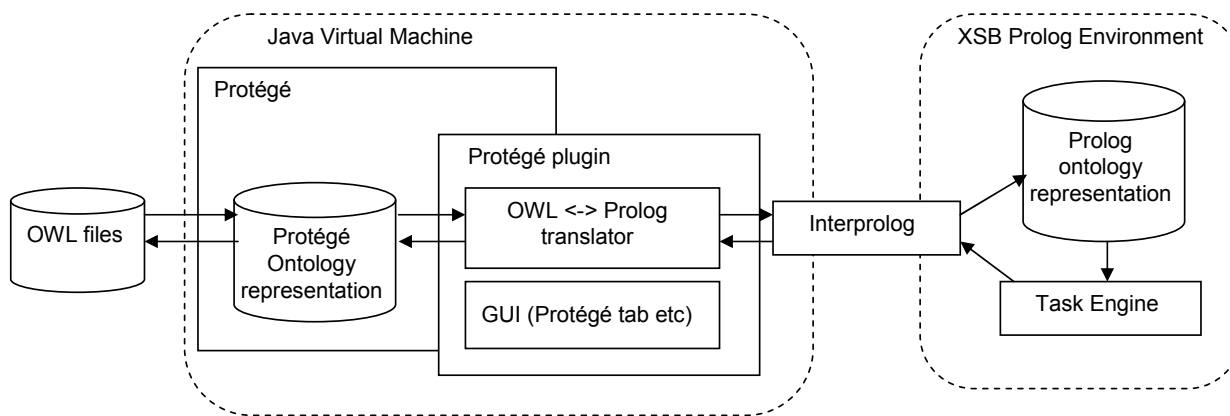

Figure 3: Implementation architecture. The new components are the Protégé plug-in and task engine. As shown by arrows, results from the task engine can be translated all the way back to OWL files.

Figure 4 shows one of several views of the results of running task synthesis for a given deployment. Results can be fairly complex. Our plug-in helps the user explore the result in terms of which resources were used, what warnings were generated, and the structure of the generated task plan.

The Analyzer tool was used for ANSC Demonstration #1, where it solved problems primarily related to net-centric discovery and transmission of video between different systems. See Section 6 for additional information about the demonstration.

While our main focus is on military training and testing, it should be noted that the Analyzer tool is fully general. No domain assumptions are built into the tool — all knowledge is provided through ontologies. The tool can thus be used for widely differing problems, as long as they can be expressed in terms of assignment of resources to roles and generation of task plans according to our task and task plan ontologies. One non-training related example that we have encoded is the problem of blood transfusion. The problem is to assign a donor that matches a recipient according to the rules of blood type compatibility.

6 CURRENT AND FUTURE WORK

Current work in the ONISTT project includes applying the knowledge representation framework to describe capabilities required to conduct Joint training events within specific Land, Air, and Maritime environments, and the capabilities available from LVC resources. Work is also being done to prepare ONISTT for transition from a demonstration capability to operational employment.

ANSC is adding capabilities to the ONISTT ontologies and Analyzer needed for planning of net-centric LVC testing events. ANSC Demonstration #1 introduced the second generation ONISTT toolset described in Section 5. One test vignette
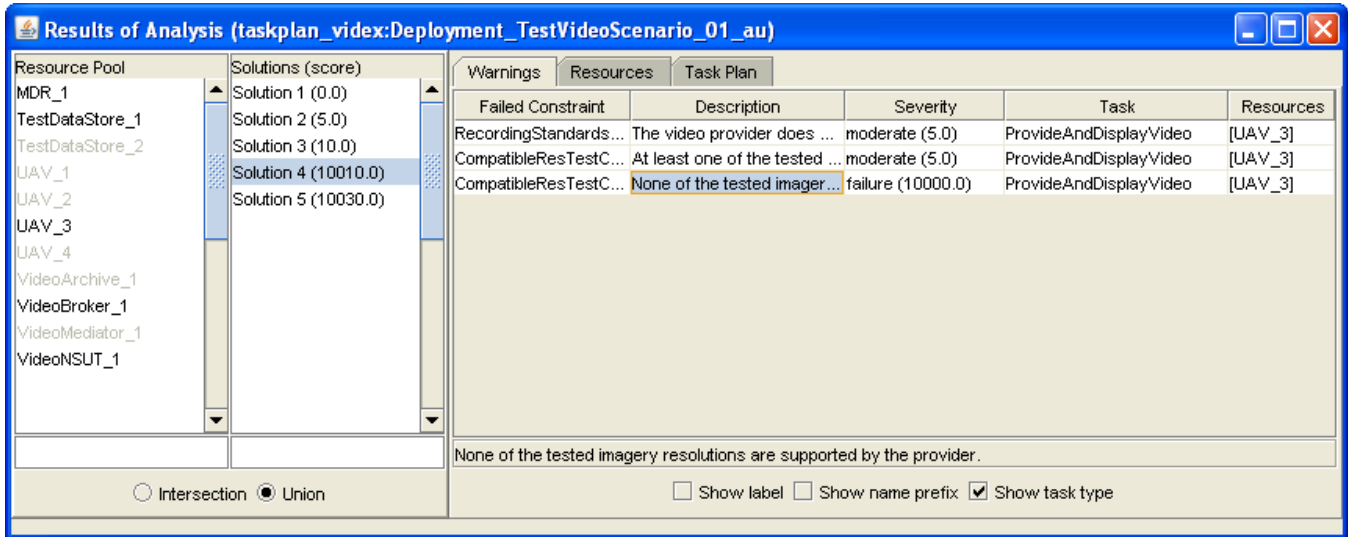
Figure 4: Results of running the task engine from the Protégé plug-in. This view shows which resources were used to generate the selected solution. In the view shown, the user can explore warnings due to failed constraints.

concerned the ability of an airborne Fire Support Coordinator (FSC) to use net-centric capabilities to find an improvisational source for motion imagery. The Analyzer needed to verify that

- The resource pool includes an MDR;
- The MDR has info about the candidate improvisational video sources.

The Analyzer also needed to verify that

1. The FSC resource knows how to use the MDR and can access it, or
2. The resource pool has a broker capable of evaluating video sources in the MDR, and the FSC knows how to use broker services.

The Demonstration #1 Task KB defined an abstract task that had (1) and (2) as alternate concrete refinements. The Analyzer found a valid solution in which the FSC uses an MDR broker.

ANSC Demonstration #1 also included specification of test parameters in the scenario KB and the ability of the Analyzer to use the specifications to perform test coverage analysis. For example, the Analyzer assessed the ability of various candidate video sources (e.g., live UAVs, or a data archive) to present the spectrum of motion imagery compression standards that were needed to fully test the FSC's video capabilities.

ANSC Demonstration #2 in August 2009 will apply the ontologies and Analyzer retrospectively to a recent LVC testing event. The objective is to verify that automated reasoning applied to ANSC KBs can arrive at conclusions consistent with the findings of the SMEs responsible for planning the actual event.

Section 2.3 introduces JTEOW as a new ONISTT-related project. The first JTEOW in August 2009 will convene SMEs in several technical domains important to LVC interoperability. The objective of the JTEOWs is to identify the key concepts for each domain and their relationships, and to define rules constraining interoperability in particular scenario contexts. After each workshop, semantic technology experts will formalize technical working group findings as OWL ontologies and SWRL rules. Standardization is an exacting business, and therefore the JTEOW will need to be an ongoing activity to achieve its goals.

The Joint Composable Object Model (JCOM) project (Lutz et al. 2009) is developing an Architecture Neutral Data Exchange Model (ANDEM) as a superset of model elements for the four major LVC interoperability communication architectures, that is, DIS, HLA, Test and Training Enabling Architecture (TENA), and Common Training Instrumentation Architecture (CTIA). One of the goals of JTEOW is to develop ontologies that will provide a semantic foundation for ANDEM. As with natural languages, translation will always be imperfect because the various communication architectures

do not have identical expressivity. The ontologies underlying ANDEM will also provide a basis for understanding what is lost or subtly transmuted in translation.

One area for future work is improving GUI presentation methods so that SMEs who are not ontology experts can readily understand and navigate KBs and explore the results of interoperability and test coverage analysis. It is particularly challenging to explain why something failed, because there are usually an extremely large number of ways to fail, but only a handful of ways to succeed. Another challenging area is ensuring that the time and space complexity of the ONISTT toolset automated reasoning scales to handle the full complexity of large training and testing events.

7   CONCLUSIONS

Military training and testing events are highly complex affairs, involving numerous live instrumentation and M&S systems as well as infrastructure resources. Our approach to facilitating such events is ambitious: describe the systems and requirements in great detail using ontologies, and use automated reasoning to automatically find problems and identify potential solutions or mitigating options.

Developing standardized, authoritative ontologies can reduce the high degree of ambiguity and diversity we find in the description of systems, the tasks they need to perform, the environments in which they operate, and the architectures designed to facilitate their working together. However, it is neither possible nor desirable to retire everything that is old, or to constrain everything that is new to co-evolve in lockstep. Heterogeneity is here to stay. In this paper, we describe an approach that makes it easier to cope.

The long-term goal is to provide a complete system that is usable by military training and testing experts who are not necessarily knowledgeable in Semantic Web technologies. For such a transition to be successful, several different Semantic Web technologies and research areas need to progress further. The scale and distributed nature of the necessary ontology development will require significant improvement in ontology engineering approaches and tools.

ACKNOWLEDGMENTS

REFERENCES

Booch, G. 1993. *Object-oriented analysis with design and applications (2nd ed)*. Addison-Wesley, Boston, MA.

Dahman, J. 1999, October. High level architecture interoperability challenges. In *Proc. NATO Modeling & Simulation Conference*: NATO RTA Publications.

Director, IS & NNEC ICT 2009. NNEC Information Portal. <transnet.act.nato.int/WISE/Informatio> [accessed September 30, 2009].

DoD CIO 2007, April. DoD Architecture Framework, Version 1.5, Volume I. <www.defenselink.mil/cio-nii/docs/DoDAF> [accessed September 30, 2009].

DoD CIO 2009a. DoD CIO/OASD(NII) Homepage. <www.defenselink.mil/cio-nii> [accessed September 30, 2009].

DoD CIO 2009b. DoD Metadata Registry and Clearinghouse. <metadata.dod.mil> [accessed September 30, 2009].

Elenius, D., R. Ford, G. Denker, D. Martin, and M. Johnson. 2007. Purpose-aware reasoning about interoperability of heterogeneous training systems. In *The Semantic Web, 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007 + ASWC 2007, Busan, Korea, November 11-15, 2007*, Volume 4825 of *Lecture Notes in Computer Science*, 750–763: Springer.

Elenius, D., D. Martin, R. Ford, and G. Denker. 2009. Reasoning about Resources and Hierarchical Tasks Using OWL and SWRL. forthcoming.

Ford, R., D. Hanz, D. Elenius, and M. Johnson. 2007, September. Purpose-Aware Interoperability: The ONISTT Ontologies and Analyzer. In *Simulation Interoperability Workshop, 07F-SIW-088*. Simulation Interoperability Standards Organization.

Grosof, B. N., I. Horrocks, R. Volz, and S. Decker. 2003. Description logic programs: Combining logic programs with description logic. In *Proc. of the 2nd International Semantic Web Conference (ISWC2003)*.

Horrocks, I., P. F. Patel-Schneider, H. Boley, S. Tabet, B. Grosof, and M. Dean. 2004. SWRL: A semantic web rule language combining OWL and RuleML. W3C Member Submission, at <www.w3.org/Submission/2004/SUBM-SWRL-20040521> [accessed September 30, 2009].

International Organization for Standardization 1995. *Guide to the expression of uncertainty in measurement*. Geneva: International Organization for Standardization.

International Organization for Standardization 2006. *Information technology Spatial Reference Model (SRM)*. Geneva: International Organization for Standardization.

Kasputis, S., I. Oswalt, R. McKay, and S. Barber. 2004. Semantic descriptors of models and simulations. In *Simulation Interoperability Workshop*. 04F-SIW-070.

Klyne, G., and J. J. Carroll. 2004, February. Resource description framework (RDF): Concepts and abstract syntax. W3C recommendation, W3C. <www.w3.org/TR/2004/REC-rdf-concepts-20040210> [accessed September 30, 2009].

Knublauch, H., R. Fergerson, N. Noy, and M. Musen. 2004. The Protégé OWL plugin: An open development environment for Semantic Web applications. In *Proc. 3rd Intern. Semantic Web Conference (ISWC 2004), Hiroshima, Japan, November 2004*, ed. S. McIlraith, D. Plexousakis, and F. van Harmelen, 229–243: Springer. LNCS 3298.

Lutz, R., J. Wallace, A. Bowers, D. Cutts, P. Gustavson, and W. Bizub. 2009, April. Common Object Model Components: A First Step Toward LVC Interoperability. In *Simulation Interoperability Workshop, 09S-SIW-031*. Simulation Interoperability Standards Organization.

Martin, D., M. Burstein, D. McDermott, D. McGuinness, S. McIlraith, M. Paolucci, E. Sirin, N. Srinivasan, and K. Sycara. 2007, September. Bringing semantics to web services with OWL-S. *World Wide Web Journal* 10 (3): 243–277.

McGuinness, D. L., and F. van Harmelen. 2004. OWL Web Ontology Language Overview. World Wide Web Consortium (W3C) Recommendation, at <www.w3.org/TR/owl-features> [accessed September 30, 2009].

Miller, J. A., and G. Baramidze. 2005. Simulation and the semantic web. In *Proceedings of the 2005 Winter Simulation Conference on Winter*, ed. M. E. Kuhl, N. M. Steiger, F. B. Armstrong, , and J. A. Joines, 2371–2377. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

OASIS 2006, October. *Reference Model for Service Oriented Architecture*. OASIS.

Object Management Group 2009, May. Ontology Definition Metamodel, Version 1.0. <www.omg.org/spec/ODM/1.0> [accessed September 30, 2009].

Preece, A., M. Gomez, G. de Mel, W. Vasconcelos, D. Sleeman, S. Colley, and T. L. Porta. 2007. An Ontology-Based Approach to Sensor-Mission Assignment. In *1st Annual Conference of the International Technology Alliance (ACITA), Maryland, USA*.

Silver, G. A., O. A.-H. Hassan, and J. A. Miller. 2007. From domain ontologies to modeling ontologies to executable simulation models. In *Proceedings of the 2007 Winter Simulation Conference*, ed. S. G. Henderson, B. Biller, M.-H. Hsieh, J. Shortle, J. D. Tew, and R. R. Barton, 1108–1117. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Thompson, A., and B. N. Taylor. 2008. Guide for the use of the international system of units (SI). Technical Report Special publication 811, National Institute of Standards and Technology, Gaithersburg, MD.

Tolk, A., and J. Muguira. 2003. The Levels of Conceptual Interoperability Model (LCIM). In *Proceedings IEEE Fall Simulation Interoperability Workshop*: IEEE CS Press.

Tolk, A., C. D. Turnitsa, and S. Y. Diallo. 2008. Implied ontological representation within the levels of conceptual interoperability model. *Intelligent Decision Technologies* 2 (1): 3–19.

Turnitsa, C. D. 2005. Extending the Levels of Conceptual Interoperability Model. In *Proc. IEEE Summer Computer Simulation Conference*: IEEE CS Press.

Vick, S., S. Murphy, R. Cost, and W. Bethea. 2006, March. An agent based approach for model composition. In *Simulation Interoperability Workshop, 07S-SIW-089*. Simulation Interoperability Standards Organization.

AUTHOR BIOGRAPHIES

REGINALD FORD is a Program Manager in the Software Engineering Program at SRI International. He has 28 years experience in training range instrumentation systems for the National Guard, Army, Navy, Air Force, and Marine Corps. He has a B.A. from Yale University and a Ph.D. from University of California, Berkeley. Dr Ford is the project leader of ONISTT, ANSC, and the JTEOW. His primary responsibilities include developing ontologies for training and testing domains and resources using OWL. Previously, Dr. Ford managed integration of live-virtual-constructive software systems and components in the Joint Training Experimentation Program (JTEP). He was software development manager for the Deployable Force-on-Force Instrumented Range System (DFIRST). As technical adviser supporting procurement of the Joint Tactical Combat Training System (JTCTS) and other government systems, his roles included system requirements specification

development, design review, and acceptance testing. His email address is `<reginald.ford@sri.com>`.

DAVID MARTIN is a Senior Computer Scientist in the Artificial Intelligence Center at SRI International, where he has been on the research staff since 1994. He has worked extensively in the field of agent-based systems, and was one of the principal designers of the Open Agent Architecture (OAA). He was a Principal Investigator in DARPA's DAML (DARPA Agent Markup Language) program, and helped to develop the Semantic Web technology stack that grew out of that program, with a particular focus on semantics for Web services and SOAs. He serves as chair of the research coalition that developed OWL-S, a service description ontology based on Semantic Web technologies. He was also co-chair of the language subcommittee of the Semantic Web Services Initiative (SWSI). He has organized workshops and presented overviews and tutorials on Semantic Web Services in a number of venues, and was a co-chair of the World Wide Web Consortium's 2005 Workshop on Frameworks for Semantics in Web Services. He is currently engaged in several projects investigating semantics-based approaches to systems interoperability. His email address is `david.martin@sri.com`.

DANIEL ELENIUS has been a computer scientist in the Computer Science Laboratory of SRI International since 2005. Before joining SRI, he was working as a research assistant at Linkoping University, Sweden. He received a Master's degree in Computer Science and Engineering from Linkoping University in 2003. Mr. Elenius has both theoretical knowledge of and practical expertise in Semantic Web, Semantic Web Services, policies, and machine reasoning technologies. Mr. Elenius successfully developed and implemented two different approaches to reasoning technology for DARPA's neXt Generation (XG) project, as well as the existing Analyzer technology in the ONISTT and ANSC projects. He is the primary developer of the tool suite described herein and also helped guide and contribute to the development of ontologies. Mr. Elenius has published papers in several peer-reviewed conferences and workshops. His email address is `daniel.elenius@sri.com`.

MARK JOHNSON is a Program Manager in the Software Engineering Program at SRI International. He holds B.S., M.S., and Ph.D. degrees in Mechanical Engineering from Stanford University. He has served as a lead engineer for the JTEP supporting the integration of constructive simulations into military and homeland security LVC training exercises. He has also played similar roles for the XCTC (eXportable Combat Training Capability) program. Dr. Johnson has also been a member of the ONISTT project team since its inception. This project leverages leading-edge ontological and semantic web technologies to facilitate the analysis and validation of LVC system confederations. He has supported the framework and process development phases of the ONISTT project, and has served as a subject matter expert for various LVC systems of interest. He has also developed or co-developed several of the core ONISTT ontologies. His email address is `mark.johnson@sri.com`.