Purpose-Aware Interoperability: The ONISTT Ontologies and Analyzer

Reginald Ford
David Hanz
Daniel Elenius
SRI International
333 Ravenswood Avenue
Menlo Park, CA 94025
650-859-4375, 650-859-3996, 650-859-4836
reginald.ford@sri.com, david.hanz@sri.com, daniel.elenius@sri.com

Mark Johnson
SRI International
4119 Broad Street, Suite 210
San Luis Obispo, CA 93401
805-542-9330, x112
mark.johnson@sri.com

Keywords:
Substantive interoperability, technical interoperability, ontology, net-centricity

ABSTRACT: *Universal substantive interoperability among an arbitrary collection of heterogeneous live, virtual, constructive (LVC) systems for an arbitrary purpose is not attainable. The Open Netcentric Interoperability Standards for Training and Testing (ONISTT) project has applied and extended the Department of Defense Netcentric Data Strategy and semantic web concepts to the more limited but attainable objective of "purpose-aware" interoperability.*

The core of the ONISTT approach for purpose-aware interoperability comprises (1) a formal description of exercise needs and confederation resources, captured in domain-specific ontologies expressed via the Web Ontology Language; and (2) an Analyzer written in XSB Prolog that applies general logical reasoning and domain-specific rules to determine whether a candidate confederation can satisfy the requirements of a proposed exercise. In the ONISTT knowledge capture phase, knowledge bases (KBs) are constructed by populating the ontologies with instance data. In the Analyzer employment phase, an exercise planner identifies the specific exercise tasks and primary training audience, and may assign specific resources (both operational and LVC systems) to roles derived from the exercise tasks. For each required interaction between two roles, the Analyzer assesses whether the capabilities provided by the assigned resources are likely to provide a satisfactory level of substantive interoperability. The Analyzer also discovers and ranks potential resources for unassigned roles. The exercise planner may adjust assignments in response to Analyzer warnings about failed or degraded interoperability. The Analyzer can also generate configuration artifacts (to be used for exercise setup).

This paper explains the ONISTT approach and benefits. It also describes the January 2007 demonstration of a working prototype in which the Analyzer used information from approximately 40 KBs to assess interoperability in several demonstration cases, including two task scenarios: Movement to Contact and Joint Close Air Support.

ONISTT is sponsored by the Office of the Deputy Undersecretary of Defense for Readiness. The observations expressed in this paper are derived from the ONISTT effort, but are solely the views of the SRI International authors.

1. Introduction

1.1 Why achieving substantive improvisational interoperability is difficult

For more than two decades, activities within the Modeling and Simulation (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. We call these ad hoc aggregations *improvisational confederations*[1] and note that they may contain elements from the live, virtual, and constructive (LVC) simulation domains.

Although many successful improvisational LVC confederations have been built and employed, these successes have typically required considerable 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 [1] introduced the notion that the success of improvisational LVC confederations requires harmony between two distinct properties of interoperability, using the labels *technical* and *substantive* to describe the two "bins" into which various root causes of non-interoperability were distributed. Tolk and Muguira [2], and later Turnitsa [3], extended Dahman's decomposition to five- and six-bin models (respectively) within a hierarchical structure called the *Levels of Conceptual Interoperability Model* (LCIM). A mapping (provided by Tolk) shows correspondence between Dahman's substantive interoperability and the upper four levels of the LCIM (i.e., semantic, pragmatic, dynamic, and conceptual).[2] Davis and Anderson [4] provide a cogent explanation of the difficulties involved in achieving substantive interoperability.

Based on insight gained from these references, as well as considerable hands-on experience integrating and operating LVC confederations, we posit the notion that finding a "silver bullet" that will ensure substantive interoperability has proven elusive because the fundamental issues are intrinsically unsolvable in a universal, generic sense.[3] We hold the view that substantive interoperability is somewhat akin to a chronic incurable disease. However, the situation is not altogether hopeless: *palliative* measures may help a patient cope with an incurable malady and enjoy a reasonable quality of life (under certain restricted conditions). Our palliative approach for purpose-aware interoperability is described herein.

1.2 New players and new approaches

Historically, the operational systems community has eschewed the unpredictability of improvisational confederations of systems, preferring the dependability of a purpose-built system-of-systems (SoS) approach. However, more recently, the inflexibility of that approach has spawned initiatives to develop a framework for building systems that can be networked together to provide improvisational SoS capabilities (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) [5], Net-Centric Operations and Warfare (NCOW) [6], and the NATO Net Enabled Capability (NNEC) [7]. These initiatives are based largely on the creation of online accessible metadata, and employing the tenets of Service Oriented Architecture (SOA) [8].

We view these initiatives as providing fundamentally new approaches for dealing with substantive interoperability issues—approaches that fall within the realm of palliative measures. The main emphasis is on making the new systems more flexible in employment of their inherent capabilities—not on requiring them to have universally applicable capabilities. For example, consider a system X, whose stand-alone functionality is composed from workflows of services $S_1, S_2, \ldots, S_n$ (per definitions in [8]). In addition to being able to perform the system X specified stand-alone capabilities, such a system has the potential to "outsource" service S_i to another provider[4] if the

[1] Use of the term *confederation* is intended to convey the absence of a central governing body, as is typically found in a *federation*.

[2] We find it useful to have a collective term (i.e., substantive interoperability) for these upper levels.

[3] The types of problems found at the lower levels of LCIM can usually be addressed by a combination of mandated commonality and/or translation/mediation; however, many quality-related issues (accuracy, fidelity, etc.) found at the substantive interoperability levels simply cannot be fixed by translation/mediation. Nor is mandated commonality a panacea, since it could drive quality factors (and cost) to levels unnecessary for some systems stand-alone needs or result in uniformly inadequate quality.

[4] Or to perform its native service S_p for some other system in the confederation.

quality of the system's native service is inadequate to meet the needs of an improvisational confederation.[5] From the perspective of the confederation, this avoids (rather than fixes) the problematic service S_i quality issue, which (in our opinion) constitutes a palliative (vice curative) approach.

While the SOA construct provides a powerful flexibility mechanism, there are important issues that fall outside the SOA boundary. For example, in the above situation, what entity is responsible for concluding that system X is a good candidate for performing a role in the confederation (except for service S_i)? What criteria are used by that entity to reach such a conclusion?[6] The SOA reference model (SOA-RM) acknowledges the need to make such choices (see 3.3.1.5 in [8]), but states that these decisions are outside the scope of the SOA-RM.

1.3 What ONISTT provides

For the LVC composition problem, the questions posed above are answered by Open Netcentric Interoperability Standards for Training and Testing (ONISTT):

Q: What entity is responsible for concluding that system X is a good candidate for performing a specific role in the confederation (except for service S_i)?
A: The ONISTT Analyzer.

Q: What criteria are used by that entity to reach such a conclusion?
A: The needs associated with role(s) assigned to system X and the interaction(s) that must be supported by the entity playing that role (in association with the other candidate entities playing the other roles).

We call this approach *purpose-aware interoperability*, since it depends on an awareness of the purpose of forming the confederation to determine adequacy of the constituents. In a January 2007 demonstration, a prototype Analyzer successfully made correct interoperability assessments by linking concepts and facts describing exercise purposes and resources. We believe the ONISTT approach could provide a useful appliqué to augment the NCOW and NNEC initiatives.

1.4 Prior work in automated composition

SRI independently developed the ONISTT concept in mid-2005, but subsequently encountered prior publication of the same basic idea by Kasputis et al. [9] (late 2004) and Vick et al. [10] (early 2005). One distinction is that ONISTT is focused on "application level composition," the highest of nine distinct levels of composition identified in [11], while the efforts described in [9] and [10] focus on "model level composition."

The concept of semantic descriptors for models and simulations is described in [9]. 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." This is essentially the notion upon which the ONISTT project is based.

However, as far as we can determine, the concepts described in [9] and [10] have not been demonstrated for a real-world use case—at least for application level composition. As such, our recent demonstration could be viewed as a strong argument for the feasibility of these concepts.

1.5 What this paper covers

Section 2 in this paper describes the ONISTT approach to automating a significant portion of the planning and setup of training and testing events. Section 3 describes the "ontologies" that enable purpose-aware interoperability assessment. Section 4 describes the ONISTT interoperability analysis software, which uses the information captured in the knowledge bases (KBs). Section 5 describes a successful feasibility demonstration of the prototype Analyzer and KBs. Section 6 discusses some of the engineering and programmatic requirements for deployment of ONISTT. A more technical view of the ontologies and the Analyzer software is provided in [12].

2. The ONISTT Approach

Figure 2.1 summarizes the ONISTT approach, which is based on semantic web technologies. First we develop

[5] Subject to technical constraints, such as latency and communication bandwidth.
[6] System X cannot be expected to make that decision because the quality of its native services are, by definition, adequate to meet the stand-alone purpose for which it was built.

referents[7] for environments, tasks, infrastructures, and systems that are relevant to the problem (1).

Because source text documents may contain ambiguities and inconsistencies, we frequently express referents as UML2 models. Referents are then formalized into OWL ontologies (2) and populated with specific facts to form KBs (3). Using a GUI (which accesses these KBs, but hides the details) the human planner defines the objectives and constraints of an exercise and proposes a partial or full confederation of participants, systems and infrastructure (4). We have developed a plug-in to Protégé [14] as an engineering prototype of a GUI to facilitate the specification of the exercise. To determine whether a proposed confederation satisfies the interoperability needs of the specified exercise, 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 the level of interoperability is "good enough" for the purposes of the exercise.

ONISTT is not intended to guarantee the accuracy of Analyzer assessments.[8] However, we believe it is a realistic goal for the output of this process to be at least as good as the output of a traditional BOGSAT (Bunch of Guys Sitting Around a Table), while being cheaper, less time-consuming, and more reusable.

3. The ONISTT Ontologies

3.1 The what and why of ontologies

The term *ontology* has a long philosophical pedigree, defined generally as the study of the ultimate nature of reality or existence. Like many other terms from other disciplines, computer scientists have seized on this term and given it a special meaning. In its overview of the Web Ontology Language (OWL), the World Wide Web Consortium (W3C) defines an ontology as a "representation of terms and their interrelationships" [15].[9] In OWL, relationships are qualified using logical expressions. A collection of concepts and relationships comprises an ontology. An ontology that has been populated with instances (or "individuals") of those concepts (i.e., specific facts) is conventionally referred to as a KB.

Several fundamental characteristics of OWL make it particularly suitable for our purposes. First, the logical semantics underlying OWL provide a basis for drawing inferences by an automated reasoning program, that is, the information expressed in the ontology is "machine understandable." Second, every concept, property, and individual in an ontology is uniquely defined by a URI and thus is accessible for reference, elaboration, and qualification by other ontologies. This facilitates distributed development of related ontologies and KBs by different organizations. An ontology file has one or more base "namespaces" to which its constituent elements are referenced. One ontology can build on another by importing its namespace.[10]

3.2 Top-level ONISTT ontologies

Figure 3.1 shows the top-level concepts and relationships in the onistt ontology. In this diagram, OWL classes are represented as UML classes and OWL properties as UML associations, with property name displayed as the "target role" label of an association [16].[11]

The onistt ontology establishes a general pattern for purpose-resource matching that might be applied to any

[7] By *referent*, we mean the most accurate and complete information available about real-world objects, following the terminology described in [13].

[8] Technical reasons underlying this caveat will be the topic of a future paper.

[9] The expressions *term*, *concept*, and *class* are often used interchangeably, as are *relationship* and *property*.

[10] Most namespaces start with *http://*. The file containing a namespace may reside on the internet or on a local file system; the physical location is designated in the editor or application that processes the ontology. Namespaces are conventionally lowercase (e.g., onistt). We will refer to ontologies by the final term in the namespace, e.g., *onistt* refers to http://www.onistt.org/ontology/onistt/onistt.owl. The ONISTT ontology files have a World Wide Web namespace, but access is currently restricted.

[11] Diagrams in this paper are consistent with the Ontology Definition Metamodel defined in [16]. We have defined additional extensions in a "visualOWL" UML profile.

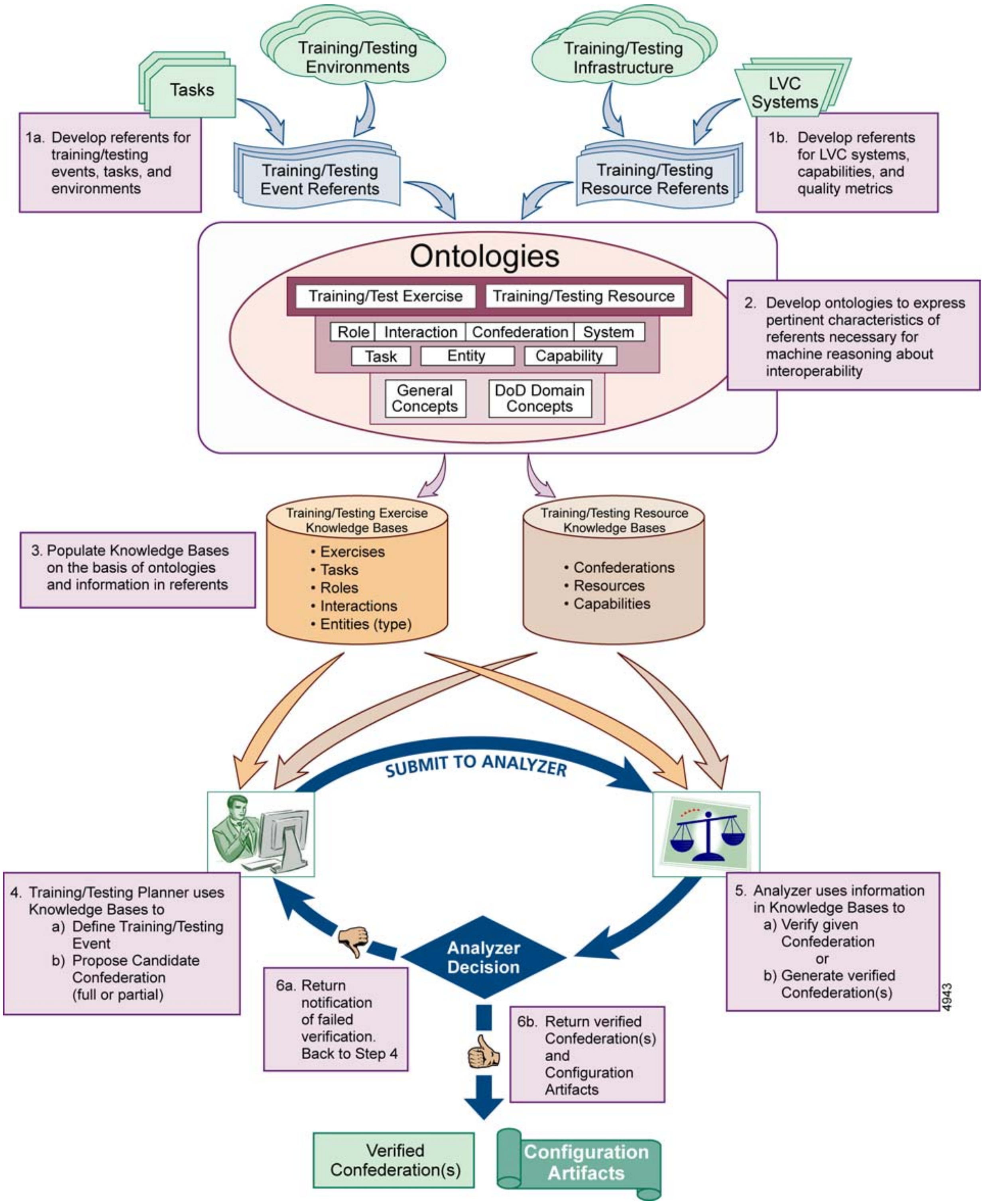

Figure 2.1 The ONISTT approach and methodology.

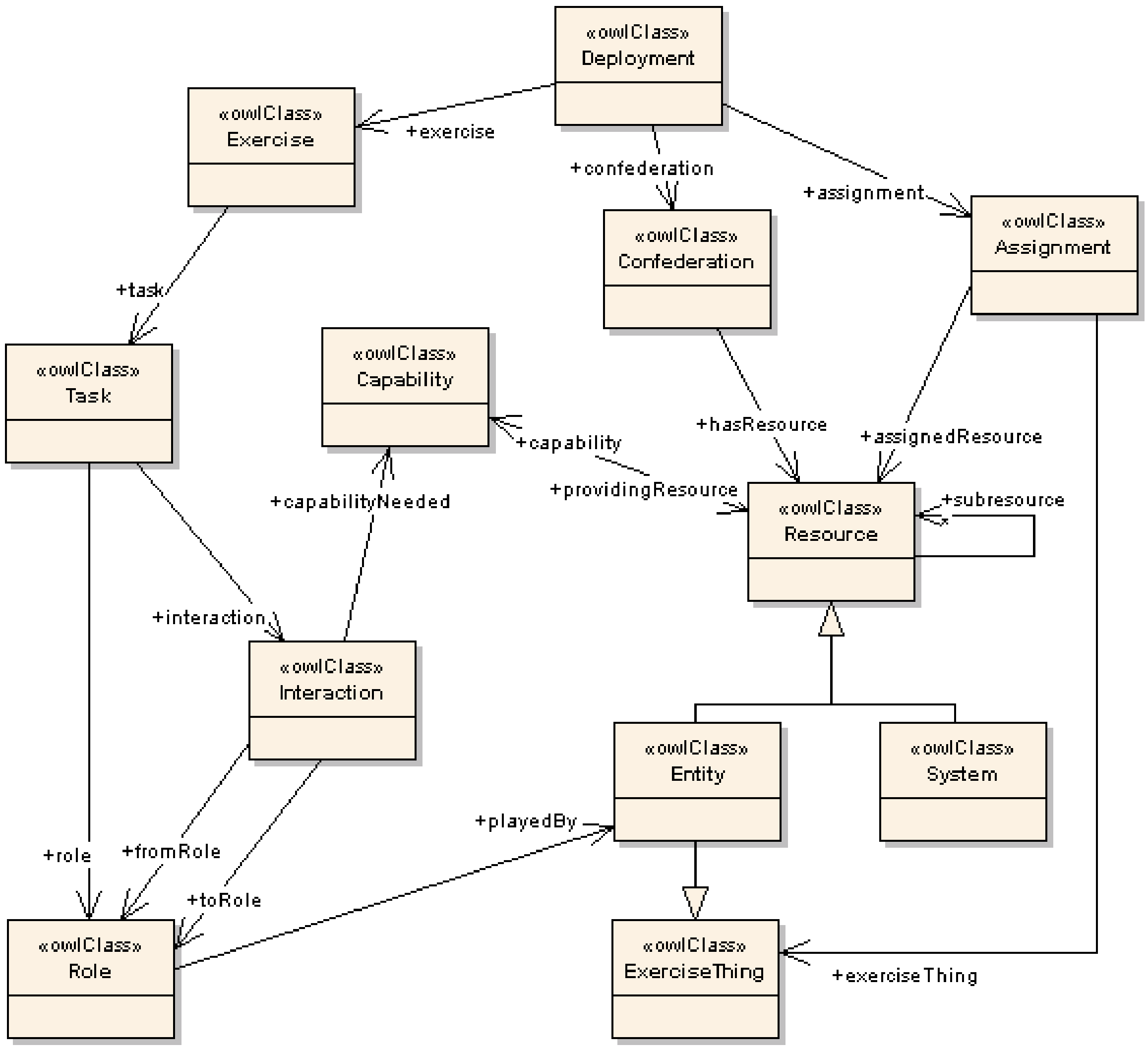

Figure 3.1 Top-level ONISTT concepts and properties.

domain. However, the top-level concepts are specific to the domains of LVC training and testing and their operational context.[12] The root concept is *Deployment*, which has three complementary parts*:*

1. An *Exercise* has *Task* objectives (i.e., purposes), from which an assemblage of needed *Capabilities* is derived.
2. A *Confederation* is a collection of *Resources* that provide Capabilities*.*
3. A set of *Assignments* match the Capabilitie*s* provided by individual Confederation *Resources* with specific Capabilities needed to accomplish each Exercise Task.

[12] The initial scope of the ONISTT program has been LVC training. However, we believe most ONISTT ontology can also be applied to testing and operational domains, either directly or through elaboration or specialization of concepts and properties.

The purpose of a training exercise is to improve the proficiency of military forces in the conduct of one or more tasks, which are defined for joint training in the Universal Joint Task List (UJTL). Responsibility for executing a task is distributed among several *Roles*. Each role in a task is responsible for performing certain actions. A complete description of tasks would include actions that can be assigned to a single role. However, the scope of ONISTT is interoperability, and therefore the only actions defined in the ontology are *Interactions* among roles. The initiator of an interaction is designated by the *fromRole* property, and the other end is the *toRole*. Although the focus of training is friendly force roles, many of the interactions are between friendly and opposing force (OPFOR) roles, and thus surrogate OPFOR roles are given equal emphasis in ONISTT KBs.

In defining an exercise, the planner may designate different *TrainingLevel* objectives for different roles. The particular interactions required between roles, and the required qualities and characteristics of the interactions, may depend on the training level. An exercise scenario defines other constraints on roles, such as *Location*.

Two concepts are shared between the ONISTT "purpose" and "resource" ontologies, providing the basis for purpose-resource matching. A Role is *playedBy* an ***Entity***, which is a kind of Resource. Each interaction requires that the roles involved in the interaction have certain **Capabilities**, which are provided by Resources.

The ONISTT "resource" ontologies are designed to support KBs that describe (1) the capabilities of individual resources (discussed in Section 3.3), and (2) the bundling of resources.

The main subclasses of Resource are Entity and System. An entity is an exercise participant, for example an F/A-18 aircraft. Training systems are often very complex, with multiple subsystems, which are recorded in KBs as subresources whose range is also the System class. An example is live training instrumentation that tracks the position of exercise entities, performs weapons effect simulation when entities engage each other, records data for After Action Review, and so on.

In LVC training, most resources are not accessible atomically but are bundled with other resources. The highest-level collection of resources in our ontologies is the Confederation. A Confederation is typically a loose and temporary aggregation of Resources—Systems, infrastructure, and other assets—that have a more stable, though not necessarily fixed, identity. Resources may have multiple *subresources*, which in turn have subresources, to an arbitrary number of levels. Dependencies among subresources constrain whether they can be used in a mix-and-match fashion or are coupled together.

Since most systems and other assets used in training exercises are multipurpose and adaptive, ONISTT ontologies are designed to describe resources apart from any intended application. However, the level of detail needed in KBs is limited to facts that are directly relevant to assessing interoperability among resources. One primary advantage of purpose aware ontologies is that an exhaustive description of each resource and capability is not necessary. In practice, we have found that examining Analyzer results is an excellent way to decide whether we have enough detail.[13]

The "assignment" ontology connects the "purpose" and "resource" ontologies. One or more Confederation Resources is assigned to each Entity defined in the Exercise. The assignments can be a mix of operational and training system resources. The Analyzer software examines each interaction and determines if the resources assigned to the interacting entities have all the requisite capabilities, and if the capabilities provided by interacting entities are compatible in quality and kind.

3.3 Capability ontologies

Capabilities are grouped in broad categories and then refined by a hierarchy of subclasses. The preliminary ONISTT taxonomy of capability groups for LVC systems in the exercise execution mode includes Archive, C^3, Communication, Countermeasure, Environmental, Exercise/Entity Control, Gameboard, Information Assurance, Initialization, Mediation, Mobility, Observability Signature, Sensor, and Weapon Interaction. Additional capability groups are defined for pre-exercise and post-exercise modes.

Properties associate capabilities with classes that define the substantive characteristics and qualities of the capability. For example, the mobility capability group for live instrumented entities includes a time-space position information (TSPI) measurement capability. For many reasons, there are difficulties involved with expressing geographic position information precisely and unambiguously, while still preserving necessary geometric, physics-of-motion, and bounded computational complexity properties. As a result, many mathematical schemes have been developed—each of which tends to favor some subset of these properties (at the expense of others). The ISO/IEC 18026 Spatial Reference Model [18] standardizes the description of these schemes, in terms of a spatial reference frame

[13] This mirrors the experience of others developing OWL ontologies, such as described in [17].

(SRF).[14] Using the logical structure of ISO 18026, we translate the attributes of SRFs that are most commonly used in training systems into an ontology. If an interaction requires two systems to exchange TSPI data, the analyzer can examine metadata in the appropriate KBs to determine if there is adequate degree of semantic congruence between each system's TSPI to support the needs of that interaction. If there is adequate semantic congruence, the Analyzer can then determine if each system knows how to interpret the data sent by the other; for example, if they do not use precisely the same SRF, do they have a capability to translate (or "mediate") between the SRFs? If translation is required, the Analyzer can determine if the translation will be perfect, or if certain attributes, (such as line-of-sight calculation or relative angles) will be distorted.

To determine whether an interaction will be sufficiently "good" often depends on qualitative characteristics of capabilities. For example, the fidelity of an engagement interaction between live instrumented entities may depend in part on the accuracy of their position measurements. Our TSPI ontology follows the ISO guide for expressing uncertainty.

One major challenge for ONISTT is expressing quality metrics for capabilities in a standard way, so that the "goodness" of capabilities provided by candidate resources can be compared with the goodness of the capabilities needed. We recognize that the static figure-of-merit approach we have used to date only supports the Analyzer in making high confidence decisions when the goodness metric is either overwhelmingly adequate or overwhelmingly inadequate—leading to the need for the Analyzer to render decisions in a trinary logic: definitely good, definitely bad, and "maybe" (i.e., inadequate granularity of quality information to make the call).

The TSPI position measurement examples above leverage the logical structure from two ISO standards. Because neither standard was documented as an OWL ontology,[15] we had no choice but to develop our own prototypes. We expect that reusable ontological representations of established standards will become increasingly available. Unfortunately, in the domain of LVC interoperability, many of the foundational standards have not yet been written [4][9][13][19].

3.4 Behavior ontologies

Standards for communication objects and their exchange have greatly facilitated LVC technical interoperability. Because some issues remain to be resolved (e.g., difficulties arising from the multiplicity of standards), work continues in this area [20]. Important as these efforts may be, the perfection of communications would still leave many issues in the substantive levels of interoperability untouched. Many capabilities of LVC resources involve behaviors that determine whether a successful communication will result in a successful or failed substantive interaction.

In the ONISTT ontologies, communication is only one kind of behavior in a sequence of behaviors that constitute an interaction protocol. Determining interoperability requires not merely assessing the ability to exchange information, or even to share a semantic understanding of information, but assessing whether all the behaviors involved in the interaction are compatible in type and characteristics (e.g., commensurate in quality).

For example, to determine whether one instrumented or simulated entity can transfer control of a missile simulation to another entity requires knowing details about the transfer of control capability and associated protocol that the resources assigned to each support [21]. Do they support the same communication protocol (e.g., as defined by DIS or HLA)? If not, is there a mediation capability that can map the protocols compatibly? Do they support the same subclasses of the protocol (e.g., "push" or "pull")? If communications are dropped or delayed, do they have compatible recovery behaviors (e.g., adjudicating duplicate missile endgame determination when the resource attempting to divest control of a simulation thinks the transfer failed when it has actually succeeded)?

3.5 Capability access and delegation

NCOW and SOA provide many foundational concepts and tools for facilitating adaptive and improvisational interoperability. The ONISTT approach to granular and precise description of capabilities could accelerate this evolution.

---

[14] An SRF is composed from an abstract coordinate system (ACS), an object reference model (ORM), and a reference datum (RD).

[15] We have seen indications that the normative portion of future standards will be expressed as ontologies using a language such as OWL.

Consider a resource that lacks a certain capability, or has a capability whose qualities may not be acceptable for all uses. A KB may indicate that the resource is able to acquire that capability from another named or unspecified resource. If the capability requirements are adequately specified, the Analyzer may be able match them with a providing resource whose capabilities are equally well described.

For example, the Analyzer might discover that in a designated exercise scenario, two interacting entities cannot share information in the "normal" way due to security classification levels, but they are each capable of employing a compatible simulation handoff protocol. The Analyzer might also be able to generate an exercise setup configuration artifact to specify, for example, that participants whose instrumentation is cleared for security level green will delegate missile flyout or engagement effects to participants whose instrumentation is cleared to security level orange. We have created an ontological description of such a handoff protocol, based on the concepts described by Hill [21].

4. The ONISTT Analyzer

This section provides a brief overview of the Analyzer. For additional technical details, see [12].

While the ontologies provide the declarative knowledge of the problem domain, the Analyzer provides the computational side of the automation. Its job is to look at the information provided and draw conclusions according to a set of rules.

We wanted to frame the operation of the Analyzer as a problem of logical deduction, as this would provide a clear semantics of what the Analyzer does. We considered (and tried) several different possibilities.

The most natural approach would be to formulate the operation of the Analyzer as an OWL subsumption reasoning problem. This would allow us to use any OWL reasoner right out of the box. However, we found it impossible to formulate the problem in this way, mainly because OWL is very restrictive with the use of quantifiers and variables. Another approach was to use OWL augmented with SWRL [22] rules. However, the OWL+SWRL combination also proved insufficient to express the operation of the Analyzer. A third approach was to translate the OWL KB to first-order logic (FOL) [23] and axiomatize the operation of the Analyzer in FOL. We tried this approach (with the SNARK[16] theorem prover), but found it too slow and sensitive to small changes in the problem formulation (as is often the case with applications of FOL theorem proving).

Ultimately, we decided to write the Analyzer as procedural code. We needed a tight integration of procedural code (the Analyzer) and declarative content (the KBs). Prolog [24] is a natural choice in this kind of situation. A large fragment of OWL, called Description Logic Programs (DLP) [25], can be readily translated to Logic Programs [26] (the logical underpinnings of Prolog). Prolog can also be used as a programming language for writing procedural code. In particular, we chose to use XSB Prolog,[17] to avoid problems that ordinary Prolog has with recursive structures such as equivalent classes or properties.

The Analyzer implementation consists of a pair of software components: A *Translator* that translates from the OWL+SWRL KBs into XSB Prolog, and the *Analyzer Core* that runs domain-specific tests on the information in the knowledge base.

Once the OWL+SWRL knowledge base has been translated, it can be loaded into our Prolog engine, and used by the Analyzer Core. The Analyzer Core currently consists of a few hundred lines of code, written in XSB Prolog. While the translated knowledge base uses XSB in a declarative way by only stating the facts, the Analyzer Core is a procedural program that queries the knowledge base in various places. This is a very flexible approach to the integration of programming and reasoning.

The Analyzer examines each pair of Roles for which an Interaction is defined. It then compares the Capabilities needed for that type of Interaction with the Capabilities provided by the Resources used to represent the Role. The Analyzer applies rules to decide whether the Resources are sufficient, insufficient, or somewhere in between. For each problematic condition, it generates a warning and assigns a severity level.

5. Analyzer Application to Example Use Cases

This section presents several of the use cases developed to demonstrate and test the ONISTT Analyzer.

---

[16] http://www.ai.sri.com/snark/

[17] http://xsb.sourceforge.net/

Case 1: Movement to Contact (fragment)

This test case involved a notional M1A1 virtual platoon assigned to the Blue Force (BLUFOR) maneuver role. The OPFOR armored role and additional BLUFOR roles (e.g., scouts) were to be assigned to a common constructive simulation. The task for the Analyzer was to evaluate candidate constructive simulations.

For the first Analyzer run we specified a legacy constructive simulation. This simulation program did not have a standard external communication mechanism (neither DIS nor HLA), so the Analyzer quickly reported this particular candidate confederation as unsuitable for the given training objectives.

A second run was made specifying a modern constructive simulation with a DIS interface. The Analyzer concluded this was an acceptable confederation, but it also issued a warning related to the portrayal of constructive entity motion on 3-D virtual displays. Specifically, this constructive simulation was not designed to drive 3-D displays, and its simulated entities were prone to exhibit distracting behaviors (e.g., instantaneous sharp turns) when viewed by trainees in the virtual system. However, if the training scenario called for largely static OPFOR entities, this unrealistic motion would not be problematic. Note that the Analyzer did not reject this confederation out of hand, but generated a warning and left the final decision to the human exercise designer.

A final run of this case specified a true SAF as the constructive simulation system. Since this system was designed to drive 3-D virtual displays, the Analyzer did not issue a warning related to unrealistic motion modeling. However, this SAF was a not *known* to have synthetic terrain correlated to the terrain used by the virtual tank trainer, so the Analyzer reported that this *might* be a potential problem that deserved further investigation. (In the previous run, the KB included the fact that the constructive simulation had terrain correlated to the virtual tank system.)

Case 2: System health status monitoring

For the purposes of this discussion, system health status can be thought of as a message indicating that a system is alive and well (i.e., still running and connected to the confederation network). In actual practice, this message could contain a richer set of information about a confederate system.

This example case, based on a real problem discovered in the field, is made up of only two systems, both of which use the exact same object model for health status reporting. However, each system, developed independently, has a different protocol governing *when* to send health status information.

The ONISTT ontologies provide a method for describing protocols. In this case, system A sends status information on a periodic interval, while system B sends status information only on demand. Since system A never asks, system B will never send, thus system A will erroneously conclude that system B is down. The Analyzer warns the exercise designer of this incompatibility.

Case 3: Joint Close Air Support

This richer, more complex scenario utilizes the following six roles: blue armored maneuver force, red opposing maneuver force, blue UAV, blue forward observer, blue terminal air controller, and blue ordnance-delivering aircraft. In this particular Analyzer run, all roles were assigned to live resources except for the aircraft, which were designated as virtual.

A full summary of all Analyzer results for this case is not possible due to space constraints; what follows are some interesting highlights:

- A live M1A1 was assigned the role of an opposing force T72. This was permitted because the KB for the training instrumentation selected to play the live entities indicates that the system supports *guising*.

- The Analyzer *discovered* the following successful five-step communication path from the live instrumented entities and the virtual aircraft system:

  Instrumentation RF messages
  → instrumentation base station (wire LAN) message
  → instrumentation LAN gateway to DIS
  → DIS to HLA gateway
  → HLA gateway to native HLA simulator

Note that a breakdown in any of these five steps would cause the proposed confederation to be rejected, and would force an exercise planner to adjust one or more role assignments. The ONISTT KB contained facts about the various training systems *and* the available training data *gateways* necessary to ensure a successful deployed confederation.

6. Future Work

The ONISTT effort to date has been focused on developing a set of ontologies and KBs sufficient to support a feasibility demonstration of the concept for several non-trivial use cases. Planned future work includes the following:

- Ontology, KB, and Analyzer development to support feasibility demonstrations in the test and evaluation (T&E) domain
- Collaborative efforts to bring more discipline to the process of creating machine-understandable semantic-rich metadata, including application in the NCOW and NNEC realms
- Maturation of the tools and techniques to support transition of the technology to an Operations & Maintenance organization for deployment and routine use

7. Conclusions

Achieving interoperability among systems that were not designed specifically to work together is a notoriously difficult problem, at least in its general form. In military settings, such as complex training exercises, it is often a top priority to minimize engineering effort and maximize the flexibility associated with deployment of “improvised” systems of systems. We believe the work described herein, based on a novel application of semantic web technologies, demonstrates a promising way forward.

A key enabler of this approach is the explicit representation of purpose (i.e., tasks, roles, interactions, and the capabilities required for their fulfillment). We have found that bounding the scope of the problem (by assessing interoperability for a given purpose) results in a considerably more manageable problem than the unconstrained achievement of substantive improvisational interoperability.

8. References

[1] J.S. Dahman, “High Level Architecture Interoperability Challenges,” presentation at NATO Modeling & Simulation Conference, Norfolk, VA, October 1999, NATO RTA Publications.

[2] A. Tolk and J.A. Muguira, “The Levels of Conceptual Interoperability Model (LCIM),” Proceedings Fall Simulation Interoperability Workshop, 03F-SIW-007, 2003.

[3] C.D. Turnitsa, “Extending the Levels of Conceptual Interoperability Model,” Proceedings IEEE Summer Computer Simulation Conference, IEEE CS Press, 2005

[4] P.K. Davis and R.H. Anderson, “Improving the Composability of Department of Defense Models and Simulations,” Rand Corporation, 2003.

[5] “Data Sharing in a Net-Centric Department of Defense,” DoD Directive 8320.2, December 2, 2004.

[6] Net-Centric Operations and Warfare Reference Model (NCOW-RM), Version 1.1.

[7] NATO Network Enabled Capability (NNEC) Feasibility Study, Version 2.0, NATO C3 Agency.

[8] Reference Model for Service Oriented Architecture 1.0, OASIS Committee Specification 1, August 2, 2006.

[9] S. Kasputis, I. Oswalt, R. McKay, and S. Barber, “Semantic Descriptors of Models and Simulations,” 04F-SIW-070, September 2004.

[10] S. Vick, S.P. Murphy, R.S. Cost, and W. Bethea, “An Agent Based Approach for Model Composition,” 05S-SIW-089, March 2006.

[11] M.D. Petty and E.W. Weisel, “A Composability Lexicon,” Proceedings Spring Simulation Interoperability Workshop, 03S-SIW-023, 2003.

[12] D. Elenius, R. Ford, G. Denker, D. Martin, and M. Johnson, “Purpose-Aware Reasoning about Interoperability of Heterogeneous Training Systems,” International Semantic Web Conference (ISWC) 2007, November 2007.

[13] “Report from the Fidelity Implementation Study Group,” Redacted by D.C. Gross, Simulation Interoperability Standards Organization, 1999.

[14] H. Knublauch, R.W. Fergerson, N. Noy and M.A. Musen, “The Protégé OWL Plugin: An Open Development Environment for Semantic Web Applications,” International Semantic Web Conference 2004 (ISWC04), pp 229-243.

[15] "OWL Web Ontology Language Overview," W3C Recommendation, 10 February 2004.

[16] "Ontology Definition Metamodel," Object Management Group (OMG) draft adopted specification, ptc/06-10-01, October 2006.

[17] A. Kershenbaum, et al., "A View of OWL from the Field: Use cases and Experiences," November 2006, http://owl-workshop.man.ac.uk/acceptedPosition/submission_19.pdf.

[18] "Spatial Reference Model," ISO/IEC 18026, 2007.

[19] S.Y. Harmon and S.M. Youngblood, "Leveraging Fidelity to Achieve Substantive Interoperability," 01S-SIW-114, April 2001.

[20] R. Saunders, S.D. Vick, G. Conrad, "Functional Interoperability Assessment—HLA and TENA," 06F-SIW-108, September 2006.

[21] F. Hill, "Transfer Control DIS Standards Document," Draft SISO Reference Document, December 14, 2004.

[22] I. Horrocks, P.F. Patel-Schneider, H. Boley, S. Tabet, B. Grosof, M. Dean, "SWRL: A Semantic Web Rule Language Combining OWL and RuleML," W3C Member Submission, 2004.

[23] C.-L. Chang, R. C.-T. Lee: Symbolic Logic and Mechanical Theorem Proving, Academic Press, New York, 1973.

[24] L. Sterling and E. Shapiro: The Art of Prolog, MIT Press, Cambridge, 1986.

[25] B.N. Grosof, I. Horrocks, R. Volz, and S. Decker, "Description logic programs: Combining logic programs with description logic, " Proceedings of the 2nd International Semantic Web Conference (ISWC2003).

[26] J. W. Loyd: Foundations of Logic Programming (2nd extended ed.), Springer Verlag, New York, 1987.

Author Biographies

REGINALD FORD is Software Development Manager for the Instrumentation & Simulation program at SRI International. He has a B.A. from Yale University and a Ph.D. from University of California, Berkeley. He has 27 years experience in training range instrumentation systems for the National Guard, Army, Navy, Air Force, and Marine Corps. He is a technical lead for the development of ONISTT. He also manages integration of LVC software systems and components in the Joint Training Experimentation Program. His previous assignments include software development and management for the Deployable Force-on-Force Instrumented Range System.

DAVID HANZ is a Principal Engineer in the Instrumentation & Simulation program at SRI International. He has 27 years experience providing technical and program support for Department of Defense and Service test and training range acquisition, development, and test. He is project manager for SRI's support to the ONISTT project. In addition to leading a team that developed a framework and process, open interoperability patterns and standards, policy recommendations, and an Analyzer for Training Confederations, he has been active in the Network Centric Operations Industry Consortium to bring insights gained from ONISTT to the attention of that forum.

MARK JOHNSON is a Senior Software Engineer in the Software Engineering Program at SRI International. He holds B.S., M.S., and Ph.D. degrees in mechanical engineering from Stanford University. Dr. Johnson currently serves as a lead engineer for the Joint Training Experimentation Program supporting integration of constructive simulations into LVC military and homeland security training exercises. He has supported the framework/process development phase of the ONISTT project, and has served as a subject matter expert for various LVC systems of interest. He has also developed or codeveloped several of the core ONISTT ontologies.

DANIEL ELENIUS is a Computer Scientist at the Computer Science Lab at SRI International. He holds an M.S. in Computer Science and Engineering from Linkoping University, Sweden. He implemented the ONISTT Analyzer and codeveloped the ONISTT ontologies. He has also implemented reasoning components for the DARPA neXt Generation (XG) program. His research interests include logic, machine reasoning, and the Semantic Web.