

14th ICCRTS: C2 and Agility

Title:

“Improving Plan Adaptation Process Through Semantic Technologies”

Topic 5: Experimentation and Analysis

Authors

Gennady R Staskevich

Joseph A. Carozzoni

Point of Contact:

Gennady R Staskevich

Air Force Research Laboratory Information Directorate

Information Systems Research Branch (RISB)

525 Brooks Road, Rome, NY 13441-4505

315-330-4889

Gennady.Staskevich@rl.af.mil

Abstract: Advancements in technology have significantly impacted the methods and tactics of modern military operations, thereby significantly shortening the window of opportunity to act. The plan development process takes up a significant portion of the available time and has a direct impact on effectiveness of the response. We postulate that the plan development process can be improved and shortened by leveraging the semantic technologies to improve the adaptation of human experiences within a distributed, mixed-initiative, case-based planning system. In particular, we present research that builds upon the semantic extensions of the Distributed Episodic Exploratory Planning project, where agents utilize the capabilities of actors and resources through the available taxonomies to improve the instantiation and adaptation of plans.

Keywords: Distributed Planning, Semantic Interoperability, Core Plan Representation, Network-Centric Operations (NCO), Mixed-Initiative Planning

Introduction: This paper describes an in-house program at the U.S. Air Force Research Laboratory Information Directorate to support the concepts of distributed planning within Network Centric Operations (NCO) [Alberts & Garstka & Stein, 1999]. More specifically, presented is the research into extending the work of DEEP-CPR Semantic Extensions [Staskevich & Hudack & Lawton & Carozzoni, 2008] to develop a semantic based plan adaptation agent. The Distributed Episodic Exploratory Planning (DEEP) [Ford & Carozzoni 2007] is a framework that facilitates distributed planning. We begin with a discussion of the way Command and Control (C2) is changing from traditionally singular to joint operations, which motivates the need for a system like DEEP. Within this paper, we use the term *semantic* to describe an agreed upon, controlled vocabulary that ensures a shared understanding of concepts and allows for unambiguous communication between agents, both human and machine within a distributed environment.

1.1 Future C2 Requirements

The Command and Control (C2) centers of the future will require frameworks that are distributed and truly interoperable where they must clearly convey commander's intent to all of the participants to ensure team consensus. Furthermore, these systems will rely on aid of multi-agents systems for plan development, logistics, and action coordination of coalition forces. We believe that semantic technologies will play a critical role in the integration and, more importantly, interoperation of these distributed systems. Mission success depends on the notion that every participating individual within a distributed team be in accord on the meanings of terms. Within a Network Centric environment we can expect to see a diverse set of planning systems and therefore we must provide this capability to capture and express major aspects of the planning process and their semantic meanings. Using the standards based semantic technologies, we are anticipating on capturing the semantic agreements, enabling portability, and plan extensibility based on the specialization of the Core Plan Representation (CPR) framework [Pease 1998]. Semantic technologies should enable partial automation of plan development (specifically translation of data), and has the capability to provide wizard based user guided planning systems.

1.2 *Brief overview of the DEEP Project*

The long-term goal of the Distributed Episodic Exploratory Planning (DEEP) project is to develop in-house a prototype system for distributed, mixed-initiative planning that improves decision-making by applying analogical reasoning over an experience base. The two key objectives of DEEP are:

- Provide a mixed-initiative planning environment where human planning expertise, intuition and creativity is augmented by the machine's number crunching capabilities.
- Support the distributed planners in multiple cooperating command centers to conduct distributed and collaborative planning.

The architecture of DEEP was explicitly designed to support the tenets of truly distributed planning. The fact that DEEP is not based on any current C2 system, we are able to explore concepts such as combining planning and execution to support dynamic re-planning, machine-mediated self synchronization of distributed planners, and experiment with the impact of trust in an NCO environment (i.e., "Good ideas are more important than their source").

This paper is organized as follows. This section identifies current limitations of the C2 planning process and focuses on addressing the future needs of the coalition based distributed C2. Section 2 introduces the proposed solution (DEEP) and its goal to more effectively integrate the planning process across globally-linked air and space operation centers to achieve true mixed-initiative planning capabilities that are effective and robust. Section 3 will give a brief overview of semantic technologies and their relevance to the C2 challenge. Section 4 discusses an approach of semantics based plan adaptation agent within DEEP-CPR system, and finally Section 5 concludes.

2 Framework for Supporting Distributed, Mixed-Initiative Planning

2.1 DEEP Framework

The Distributed Exploratory Episodic Planning (DEEP) framework is our attempt to the challenges of the future distributed C2 planning systems. DEEP is a decision-support planning system designed for providing computer-assisted planning capabilities. The goal of DEEP is to improve the commander's decision-making process by providing contextually relevant alternative actions.

The DEEP framework, shown in Figure 1, is built on the premise that planning experiences can be captured, retrieved, instantiated, and adapted in both a machine- and a human-readable form. To enable a true mixed-initiative planning process, the system utilizes a distributed multi-agent architecture in conjunction with a Case-Base Reasoning (CBR) [Kolodner 1993] system to provide a fully recursive environment that learns by building its case-base and fine-tune its assistive planning capabilities. This architecture is domain-independent and applicable within any level of command hierarchy [Ford & Carozzoni 2007]. This independence is directly related to the contents of a case base, as it will be shaped and adapted through retention of new planning experiences.

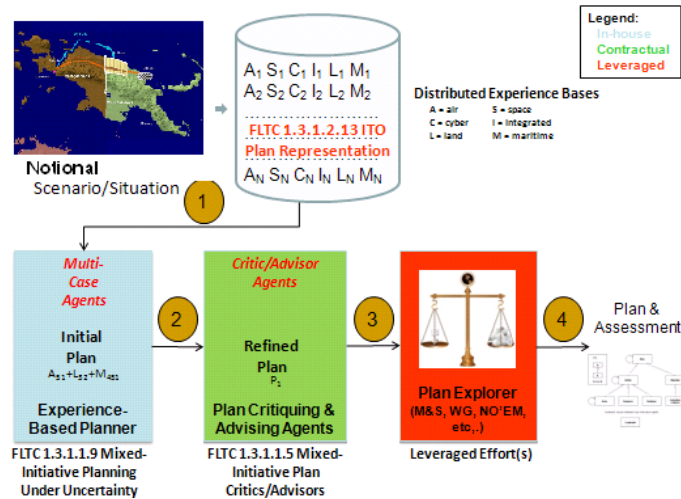


Figure 1 - The DEEP Process

2.2 ARPI CPR

To support the DEEP architecture, a flexible plan representation was needed that was capable of maintaining planning experiences in both a human- and machine-readable form. Several candidate plan representations were evaluated and contrasted such as: Information Warfare and Planning Capability (IWPC) [General Dynamics Advanced Information Systems 2007], Joint Air/Ground operations: Unified, Adaptive Re-planning (JAGUAR) [Wagner, DARPA] and ARPI-CPR [Pease & Carrico 1997] (see section 2.2). ARPI-CPR was selected as most appropriate plan representation to be used within the DEEP framework as it provides an extensible and a simple generalized upper planning abstraction model.

The number of planning systems and heterogeneity between them has grown significantly in recent years necessitating a unified plan representation to support the core needs of many different planning systems [Pease & Carrico 1997]. During the years of 1991-1998 ARPA and Rome Laboratory, as part of ARPA Rome Planning Initiative (ARPI), co-sponsored the Object Modeling Group (OMG) to derive a neutral base plan representation in order to facilitate information exchange among different planning systems. Their study resulted in what is known as the ARPI Core Plan Representation (ARPI-CPR) model.

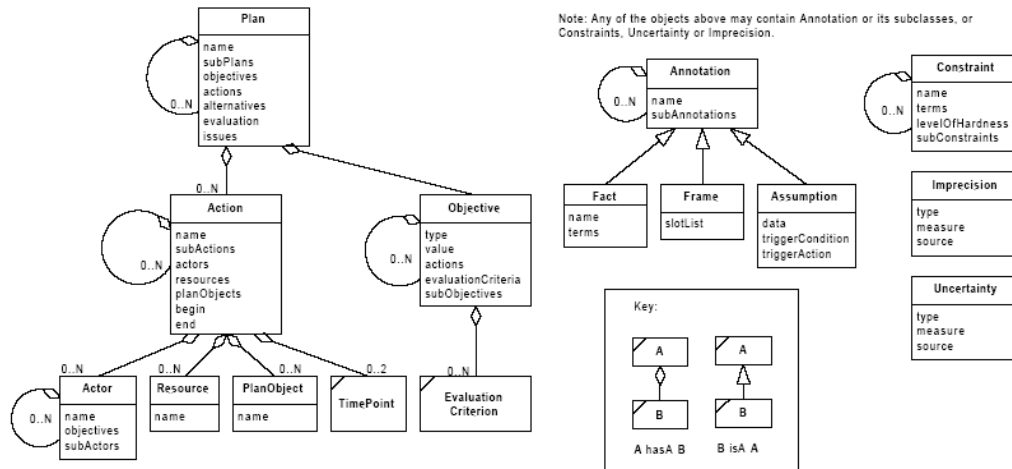


Figure 2 - ARPI-CPR Model

The ARPI-CPR model, shown in Figure 2, is an abstract specification that provides a highly flexible and recursive architecture for the plan representations. This specification is based on a commonly shared set of objects: Action, Actor, Objective and Resource [Pease & Carrico 1997], where objective and an action are the core elements of a plan. This framework syntactically captures the foundational planning concepts using object oriented design.

2.3 DEEP adaptation of CPR

While the DEEP framework represents a “conceptual” plan, we needed to extend it to include other entities required for a mixed-initiative, distributed planning system. Our extensions to the APRI-CPR model were driven by the requirement to encode plans as part of cases in a CBR system. A significant difference between ARPI- and DEEP-CPR is that planning information within DEEP is structured (currently using taxonomies), making the free text used in ARPI-CPR inadequate. Further, since DEEP uses a CBR system for plan selection and storage, it was necessary to extend DEEP-CPR to make it a component of a case.



Figure 3 - DEEP-CR Model

Figure 3 presents a representation of the CPR model as extended for DEEP. At the most abstract level, the planning experiences is encoded and stored inside a DEEP Cases (component of CBR system), which we refer to as the DEEP Case Representation (DEEP-CR).

3 Semantic technologies and their relevance to the C2 challenge

The primary goal of this section is to briefly introduce the reader to the foundational concepts of the semantic technologies, their purpose and the benefits of using them. Our motivation here is to help the prospective adapters of Semantic Technology to avoid majority of the confusion that is associated while learning it. Secondly, is to show the simplicity of RDF and to show applicable examples to the C2 domain.

3.1 Semantic technologies

Unlike people, computers lack the capability to understand; words have no explicit meaning to the machine; in fact words are merely collections of symbols. There is a specialized field of mathematics called the Set Theory [Jech 2003]. This theory provides a convenient and a formal way of describing the meaning of concepts through a set of special relations, in particular the class membership. Basically stated, the semantic web is the specialized implementation of the set theory, enabling computers to appear to comprehend, while in fact they merely follow a set of simple and well-defined set of rules.

3.2 Resource Description Framework

Semantic technologies provide a way to capture a set of mathematical relations in a computer readable way. At the very core of Semantic Technologies is the Resource Description Framework (RDF) [W3C RDF Core Working Group, 2004]. Conceptually, there are two primary concepts in RDF: *objects* and *relations*. To a mathematician a RDF is akin to a directed graph concept (Ellis, et al. 1995).

Objects in RDF are used to represent *People*, *Places*, and *Things*, while the relations are used to describe the links or a “relations” between objects.

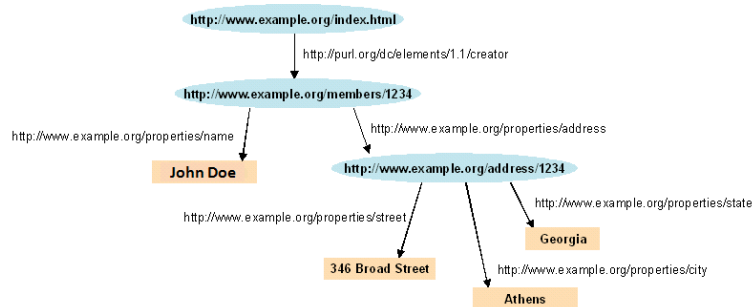


Figure 4 – Objects and Relations

3.2.1 RDF Resource, Literal

In RDF everything is a *Resource* or a *Literal*. All Objects and Relations are *Resources*. Resources is used to describe People, Places, Things, and Relations. A literal is a string of characters with no special meaning associated to it, much like a comment in a programming language. Literals are always used as a leaf node in a graph.

3.2.2 RDF Triple, and Statement

In RDF all of the information is expressed in a form of a Triple, Figure 5. An RDF Triple is composed of three parts: two objects (person, place or thing) and a relationship that links them together. Another name for a triple is a Statement (and often it is referred as such). Properly speaking a triple is made up of Subject, Predicate and Object. Where subject and object are essentially two things that describe some person, place, or thing. A predicate is essentially a relationship that specifies an action, property (e.g., walking, owns...). There is a direction in a triple, typically in the graph representation the direction is indicated by the predicate's arrow. For example, without a direction the meaning of a statement would change. To illustrate this look at Figure 5, we can see that **John hasFather Richard** triple would have a completely different meaning then **Richard hasFather John**.

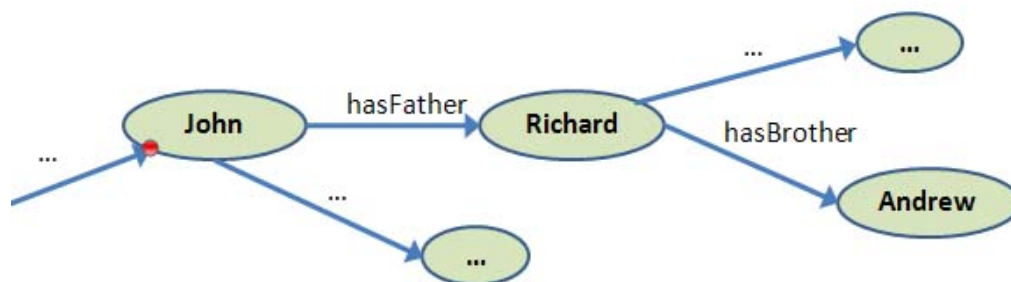


Figure 5 – RDF Triple and an RDF Statement

3.3 RDF Schema (RDFS) and OWL

RDF provides a convenient way to describe general purpose or domain specific information in a form of a graph as a set of objects and relations between them, Figure 5. Essentially, the RDF Schema is a controlled vocabulary that provides a set of core relations such as: *Class*, *Subclass*, *Property*, *Subproperty*, and *Individual* where each has a very specific meaning within RDF. Likewise Web Ontology Language (OWL) [Smith, Welty and McGuinness, 2004] is yet another set of specialized relations that have very specific meaning in RDF. Furthermore, OWL is divided into three categories: *OWL-Light*, *OWL-DL*, and *OWL-Full* [Smith, Welty and McGuinness, 2004]. This categorization is based on the complexity and the computability of OWL relations. The relations defined in OWL-Light and DL ensure that there are no infinite loops in logic and that reasoner is guaranteed to eventually compute the answer.

3.4 *Inference engines*

When properly used, the semantic technologies have the potential to provide an invaluable source of information. Perhaps the greatest benefit that semantic technologies have to offer is the ability to dynamically at runtime to infer new information using well defined formal logic. A reasoner utilizes an inference engine that implements one or more rule sets (specified by RDF, RDFS, and OWL) in conjunction with user provided ontology to deduce new information.

3.5 *Semantic Technologies and their relevance in a collaborative environments*

Semantic Web provides a convenient way to describe domain specific knowledge in a machine-readable form. Furthermore, they enable us to externalize this knowledge in a way that has consistent meaning to all members, independent of any operating system (and programming language). Semantic technologies are formally defined and based on mathematics rather than particular spoken language. The very nature of the semantic technologies makes it an ideal framework for sharing information within collaborative environments.

3.6 *Suggestions*

Based on a personal experience, the most challenging part of working with semantic technologies is the process of ontology development. However, when done correctly the effort will pay for itself. The first recommendation is to scope out the semantic landscape to see what is available and strongly consider reusing other popular ontologies (FOAF, DC, etc...). As a last resort build your own, but remember to keep it simple or you may very quickly end-up trying to describe the entire Universe.

Our suggestion for ontology development is to start by defining the properties of individual's first rather than building a class hierarchy. Doing so will help to keep the system requirements to be operationally driven. Furthermore, this will help to avoid the specifying constraints at the top of the class hierarchy that may not hold true to every sub-member of the hierarchy. The implication here is that the children in the hierarchy unintentionally will be forced to inherit invalid constraints and leading to erroneous results.

It is important to be aware of implications of using Object Orient Programming (OOP) paradigm for the ontology development process. For an example the OOP design, instances are defined at project development; while in a Set Theory, class membership is rather determined by the properties that an individual (instance) possesses. Thus an individual is a member of all classes that it satisfies the constraints there of. Remember, that the goal of having a good ontology is to let the inference engine deduce the individual's (instances) class membership at runtime and / or validation of constraints. Furthermore, the ontology development process will aid a user to expose hidden assumptions of their environment.

4 SAAGE – Semantic Adaptation AGent

The goal of this section is to introduce the reader to our research of developing a Semantic Adaptation AGent (SAAGE). Our SAAGE adaptation framework builds on top of the DEEP project as described in [DeStefano and Lachevet, 2008] [Ford and Carozzoni, 2007] and the semantic extensions of DEEP-CPR

[Staskevich, et al. 2008]. Conceptually, the DEEP project can be described as an experience based planning framework within a distributed environment using agent based technologies. The semantic extensions leverage RDF to provide semantic groundings for the plan representation objects (plan, objective, action, actor, resource, etc) as defined in DEEP architecture.

4.1 SAAGE overview

Our research builds upon the semantic extensions of the Distributed Episodic Exploratory Planning (DEEP) project [Staskevich et al, 2008], where agents utilize semantic technologies to assign actors and resources of an action in order to improve the instantiation and adaptation phases of the plan development process. The SAAGE framework has three primary parts (Figure 6): action relevance critic, action instantiation agent, and semantic adaptation agent. The inputs to SAAGE are: a Situation object and relevant Case object. The Situation object describes current objectives available actors and resources. The Case object contains actions of past planning experience.

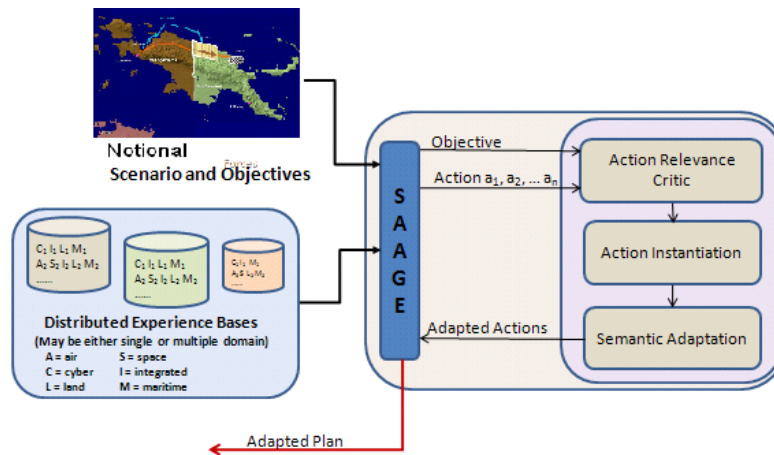


Figure 6 – SAAGE Components

The first step in the SAAGE framework is to compute relevance scores of past events against current objectives [Ford and Carozzoni, 2007]. Within this paper, the term “event” is identical to an already executed action. The second step is the instantiation. After the relevance scores are computed, actions that meet the threshold criteria are instantiated. Finally, the instantiated actions are passed to the semantic adaptation agent. If no relevant actions are found for the objective, then the highest scored action is adopted.

4.2 Action Relevance Critic

The relevance scores are computed (described below) by the Action Relevance Critic (Figure 7). Recursion is used to compare sub-components of the Objective object against the sub-components of an Action object (Staskevich, et al. 2008). The *Objective* object has three parts:

- Purpose – A concept that captures the reason or the intent of an objective.
- Method – A concept that states how a given objective is going to be achieved.
- End-state – A concept that defines a measurable metric to determine a success or a failure of a given objective.

Likewise, the Action object is also composed of other objects such as Actors, Resources, Roles, Effects, Location, etc. The final relevance score is computed using weighted average by recursively comparing:

- Objective’s End-state objects against Action’s Effect objects
- Objective’s Purpose objects against Action’s Role objects
- Objective’s Method objects against Action’s Role objects

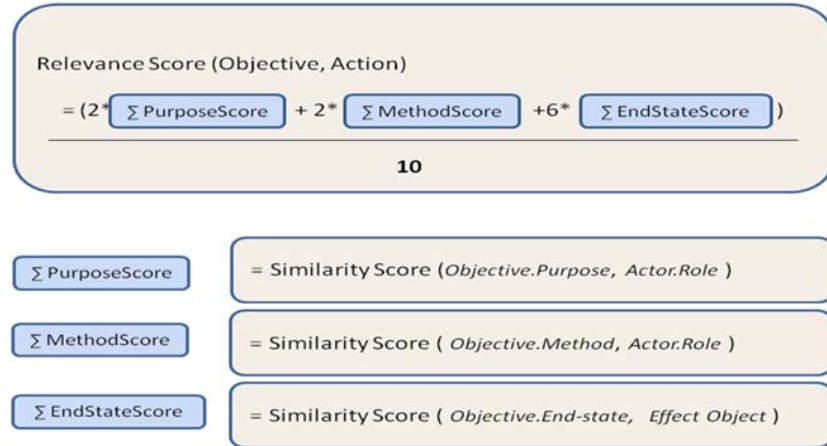


Figure 7 – Action Relevance Critic

Most of the weight of the relevance score is contributed by comparison of the objective’s End-state object to an Effect object of an action. The reason being is that they contain more actionable information in the form of simple statements, such as (**Object-1, Relation, Object-2**), where Objects can be any actors, resource, location, etc.... Likewise, each of the **SimilarityScore(...)** methods are composed of other simpler score methods.

The similarity is computed by comparing two terms defined within the DEEP ontology. Most often these terms refer to the relation part of the statement. The comparison is based on a variation of the (Wu 1994) link distance algorithm. The link distance is computed as follows: if there is no subclass between terms then the score is zero, if two terms are identical then the score is 1. If there is a subclass connection somewhere in the tree that means that both terms share a common parent. The depth of the “common parent” is divided by the depth of the deepest branch shared by the common parent, thus always returning a relevant score.

4.3 Action Instantiation

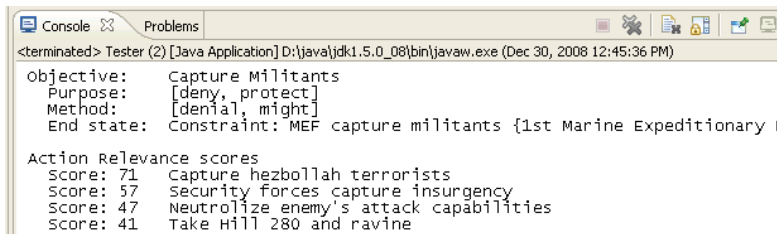
The second part of the SAAGE framework focuses on action instantiation. Instantiation is one of the most primitive types of adaptation. Metaphorically speaking an instantiation is a process of reframing a past event in retrospect of current situation. Thus, in our case this means that the action found to be relevant is updated to contain current resources, location, etc... The instantiation process works by recursively breaking down the action object into its sub components and instantiate each object at a time.

4.4 Semantic Adaptation

The third part of the SAAGE framework focuses on the adaptation of past actions. The Situation object is used to describe the current situation and provides a current set of resources, locations, events, and objectives. Furthermore, the Situation object is used as a template for the adapted plan. Next, each

Objective object is paired up with a set of actions from past experience that were scored to be relevant by the Action Relevance Critic. The adaptation process begins by sending the actions (along with their relevant objectives) to the semantic-based adapter, where each action is adapted with respect to its relevant objective. The adaptation works by recursively breaking down the Action object into its sub components and adapts them one object at a time. After the adaptation, each adapted action is added to the adapted Plan object. Finally, the adapted Plan object is returned.

Our simple prototype has been successfully implemented, and provides acceptable results (Figure 8). In the current implementation of SAAGE, there are several components that have elementary capabilities, such as: location similarity and adaptation, type (and sub-type) based adaptation of actors and resources. Next prototype will focus on reengineering parts of our taxonomies to provide better support for capabilities based adaptation of actors and resources.



```
<terminated> Tester (2) [Java Application] D:\java\jdk1.5.0_08\bin\javaw.exe (Dec 30, 2008 12:45:36 PM)
Objective: Capture Militants
Purpose: [deny, protect]
Method: [denial, might]
End state: Constraint: MEF capture militants {1st Marine Expeditionary
Action Relevance scores
Score: 71 Capture hezbollah terrorists
Score: 57 Security forces capture insurgency
Score: 47 Neutrolize enemy's attack capabilities
Score: 41 Take Hill 280 and ravine
```

Figure 8 – Action Relevance Critic scores

5 Conclusion

Presented in this paper is the Semantic Adaptation AGent (SAAGE), a semantic-based plan adaptation agent that scores past actions against current objectives, and adapting those that are found to be relevant. SAAGE is one of many components that make up the Distributed Exploratory Episodic Planning Project (DEEP) framework. The motivation behind this project was to leverage semantic extensions of the DEEP-CPR model to develop a plan adaptation agent that takes the advantage of the richer content description for reasoning and adapting plans. Semantic technologies enable us to externalize domain specific data independent of an operating systems and programming languages while contributing to the expansion and management of the coalition planning community through a *network effect* (Liebowitz and Margolis 1995).

Planning within coalition environments first and foremost requires an agreement of terminology. Our contribution here is that we provide our vocabulary based on the ARPI-Core Plan Representation, as we express our domain specific terminology through controlled vocabularies and ontologies using RDF. Doing so enables us to contribute our building blocks (much like LEGO pieces) for the planning community within a coalition environment.

Semantic technologies, while not a silver bullet, are a key ingredient for an enablement of truly distributed planning systems. Thus, when developing networked/collaborative applications, one should consider taking advantage of using semantic technologies (e.g., RDFS/OWL) to externalize as much of the domain specific data, such as definitions of concepts their relations. Doing so will enable more flexible reasoning and expose the structure of stored knowledge in a form that is accessible to both the

machine and the users. These capabilities support the requirements for dynamic, distributed heterogeneous planning agents in a more transparent representation that is accessible by all users in the command chain.

References

- Alberts, D., Garstka, J., and Stein, F., 1999. Network-Centric Warfare: Developing and Leveraging Information Superiority. CCRP Publication Series.
http://www.dodccrp.org/files/Alberts_NCW.pdf
- Ford, A., Carozzoni, J., 2007. Creating and Capturing Expertise in Mixed-Initiative Planning. Proceedings of the 12th International Command and Control Research and Technology Symposium. Department of Defense Command and Control Research Program, Washington DC.
- Jech, T., 2003, Set Theory: Third Millennium Edition, Springer Monographs in Mathematics, Berlin, New York: Springer-Verlag, ISBN 978-3-540-44085-7
- Kolodner, J. 1993. Case-Based reasoning. San Mateo CA: Morgan Kaufmann.
- Pease, A., and Carrico, T., 1997. The JTF ATD Core Plan Representation: A Progress Report. *In proceedings of the AAAI Spring Symposium on Ontological Engineering.*
- Pease, A., 1998. Core Plan Representation, Object Model Working Group. *Request for comment.*
- Staskevich, G., Hudack, J., Lawton, J., Carozzoni, J., 2008. Semantic Interoperability in Distributed Planning. Proceedings of the 13th International Command and Control Research and Technology Symposium. Department of Defense Command and Control Research Program, Seattle WA
- Wu, Z., Palmer, M., 1994, Verb Semantics and Lexical Selection. ACL