2006 CCRTS
THE STATE OF THE ART AND THE STATE OF THE PRACTICE

Support for Dynamic Collaborative Action Teams

C2 Architecture, C2 Concepts and Organizations
Architecture
Experimentation

R. Scott Cost
Markus E. Dale
David P. Glock
James Mayfield
Christine O. Salamacha
David P. Silberberg

The Johns Hopkins University Applied Physics Laboratory
11100 Johns Hopkins Road, Laurel, MD 20723-6099

POC: R. Scott Cost
The Johns Hopkins University Applied Physics Laboratory
11100 Johns Hopkins Road
Laurel, MD 20723-6099
FAX: 443-778-6904
Phone: 443-778-1137
Email: Scott.Cost@jhuapl.edu

Abstract

An essential step in fielding a timely and effective response to events of global importance is the ability to rapidly identify and integrate a crisis action team. This group should consist of exactly those individuals best qualified to manage the situation. Often, the organization of such a team follows identifiable patterns. Thus, it is important to rapidly identify the type of team, or pattern, required, and to identify the individuals that meet the requirements specified by this pattern. This is a challenging task, as information about people is often distributed across multiple locations, inconsistent or out-of-date, and phrased in the language of different domains. We present a framework that facilitates the rapid integration of teams by identifying scenario-based patterns, and using agent-based search across enterprise boundaries to identify people and assist in their assignment.

George Abitante, Michael Cramer, Steve Forsythe, Timothy Frey, Anil John, and Kim Richeson contributed to the concepts and technical discussions presented in this paper. The authors appreciate their inputs and valuable insights.

Keywords: Collaboration, patterns, dynamic activation, virtual resource broker, dynamic collaborative action team

1. Introduction

Disasters such as the 2004 tsunami striking Southeast Asia happen with little or no warning. An essential component of a timely response is a team whose members have a wide range of expertise. Such teams are often identified during contingency planning. But the breadth of possible events makes it difficult to plan for every conceivable possibility. The actual circumstances of an event usually require the modification of existing plans and often require additional expertise. Even when a static list of team members has been generated, it is difficult to keep the information up-to-date in a centralized location. Personnel from different domains are reassigned, organizational units change and operational foci shift. Resource contention, where one person is assigned to multiple tasks is also an issue. Nonetheless, the roles and expertise needed to respond to specific events often follow patterns. The identification of such patterns and the creation of tools to help map people to pattern roles together form the core of an approach for assembling crisis response teams.

The Johns Hopkins University Applied Physics Laboratory has been building systems to support the activation and operation of *dynamic collaborative action teams* (DCATs) based on this model [1][2]. Our work addresses the technical challenges associated with bringing together participants in a crisis situation and equipping them with tools that support effective collaboration in a dynamic environment.

This paper describes our approach. First, we describe two key component technologies - software agents and the Semantic Web. Next, we detail our approach to representing team patterns and to mapping the people in an organization onto the roles of those patterns. Finally, we describe our implementation plans and provide some discussion of the approach.

2. Key Technologies

Two technologies serve as key underpinnings to our approach: software agents and the Semantic Web.

2.1. Software Agents

The term "software agent" has been used in a variety of contexts. We follow the definition of Wooldridge [3] by defining a software agent as a software program exhibiting:

- *Autonomy*: agents execute in their own thread of control.
- *Social ability*: software agents are capable of interacting with other software agents by exchanging high-level messages using common ontologies.
- *Reactivity*: agents perceive changes in their software environment and can act on those changes.
- *Proactiveness*: agents have goals and pursue actions that will help them achieve those goals.

These attributes of agents are well suited to dynamic environments where structure and available resources are in a constant state of flux. Agents can adapt their plans in reaction to changes in the environment. Autonomy combined with proactiveness allows agents to independently pursue and adapt goals as necessary. Their social abilities allow agents to collaborate even across organizational boundaries.

The adoption of a software agent approach allows us to exploit a wealth of software that has made its way out of research laboratories and into operational systems over the past decade.

2.2. The Semantic Web

The Semantic Web, an effort to facilitate the automated manipulation of information on the Web, provides a common framework that permits data to be shared and reused across application, enterprise, and community boundaries. With roots in the DARPA DAML effort, the Semantic Web is a collaborative effort led by W3C with participation from a large number of researchers and industrial partners [4]. It is built on the Resource Description Framework (RDF), which is in turn built on XML for syntax and Uniform Resource Identifiers (URIs) for naming. Sir Tim Berners-Lee stated that "the goal of the Semantic Web initiative is to create a universal medium for the exchange of data where data can be shared and processed by automated tools as well as by people." [5]

Information encoded in the Semantic Web relies on the existence of ontologies to describe the concepts within various domains. An ontology is a formal specification of concepts and their relationships [6]. The Web Ontology Language (OWL) has been developed and is widely used for this purpose [7]. By directly referencing ontologies, domain-specific meaning of concepts can be readily accessed, facilitating matching and translation.

The Semantic Web facilitates a broad range of applications, including scheduling [8], service discovery [9], information retrieval [10], and trust [11].

These concepts are important to DCAT formation because they support matching of people to requirements. Our overall goal is to facilitate the process of assembling teams, both across and within enterprise boundaries. Often, resources owned or catalogued by different organizations are described in the language of different domains. A system attempting to exploit such information must look behind the differences in the direct representation to access the underlying characteristics of the resources. The use of Semantic Web ontologies allows us to reason about the meaning of resource descriptions, and therefore better compare resources across domain boundaries, and make more intelligent decisions about team formation.

3. Approach

We propose a multi-tiered rule-based approach to automated DCAT formation. A resource broker agent implements a behavior which is defined by a high level set of rules. More specific behavior, such as access to particular data sources or preferences for resolving constraints in certain environments are given in more specific rule sets, which are integrated as appropriate. In this way, we can generate system behavior that is optimally tuned to the particular environment, context and need.

The process begins with the identification of a pattern; this pattern is the input to the resource broker process.

3.1. Patterns

The DCAT approach begins with the creation of patterns. Patterns define the set of roles that are required to field an effective response to a given event, and to specify the inter-relationships among these roles. Developed through advanced planning, training and exercises, patterns will serve as a foundation for understanding the response needs for various events. Ideally, a repository of patterns exists for a large number of event types, and these patterns can be adapted or revised to provide solutions to newly encountered problems. While the information describing the characteristics of the individual roles is important in resolving roles to specific people or other resources, the information about how they interrelate provides further constraints or clues. In addition, the information in templates may be static (e.g., expertise, access privileges) or operational (e.g., availability, current deployment).

3.2. Rules and Rule Sets

Each pattern has a set of roles that must be filled by people or other resources in order to establish the DCAT. Roles are filled by a *resource broker* agent. For each type of role, there is a standard procedure or set of procedures for locating resources to fill that role. These procedures may include database searches, human contacts, etc.

We adopt a rule-based approach to filling roles. That is, the resource broker, and related components are driven by sets of rules, which define their behaviors. These rule sets define *Tactics, Techniques and Procedures*, or *TTPs.* This approach provides good alignment with the nature of the task at hand, in that the resolution of various types of resources is generally governed by a specific set of rules or procedures particular to that resource or class of resources. The role of the resource broker is to implement a variety of search and integration tasks, which are defined as procedures or rules to be followed. These procedures can be referenced and integrated as part of the behavior of the agent as task needs arise.

There are various levels at which a task may be defined. At the highest or most abstract level, we define certain basic operations, such as a general search for a resource, given a set of constraints. This provides the driving pattern of activity for the resource broker. Rule sets at this level define the basic behaviors of the resource broker. For the primary search task described, there would likely be only one set.

The general pattern is then augmented to reflect more specific factors, such as the specific nature of the search, time and access constraints, or user preferences. Thus, the selection of additional rule sets would likely be driven by the context of the request.

Specific search operations are then mostly tightly defined by the rule sets, or TTPs. Given a need to access a specific resource, these rules sets specify the rules or procedures for engaging this resource. For example, if a search for a specific target resource required access to database A, the TTP for engaging database A would be identified in a repository of TTPs, based on the class of resource needed (or possibly the specific resource or role), and invoked.

3.3. Resource Brokers

Given a specific event, a resource broker must identify a DCAT pattern that describes the resources required to respond to the event. Specifically, the pattern contains references to the set of roles that are required, and the location of TTPs that are to be used in the resolution of these roles.

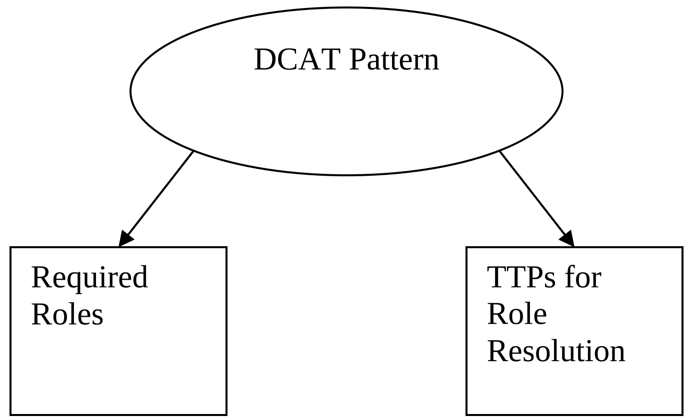

Figure 1: DCAT Pattern

A resource broker is tasked to identify resources that match the roles specified by the template. The behavior of the broker is defined by the general set of rules governing this task, and by the additional rules set by the context or user. That is, one master set of rules may describe the overall behavior for the task (a specific type of pattern resolution, such as crisis management in military space), which may then be augmented by rules that define practices specific to the organization or individual authorizing the search, and reflecting enterprise policies or individual preferences.

For a given role, the broker will identify a set of data resources appropriate for the information required. It will then access a repository to recover the specific TTP required for interaction and negotiation with that resource. The nature of such TTPs may vary widely, as the nature of interaction with various resources will correspondingly vary.

Once a set of potential resources has been identified from one or more data sources, the broker must determine if the resources satisfy the constraints given by the DCAT pattern. These constraints determine relationships among the roles in a team, such as common experience or skills, previous working relationships, chain of command, and so forth. These constraints will guide the broker to a selection of an optimal set of available resources. It may also require that searches for some roles be revisited if not all constraints can be satisfied.

After the broker has identified a set (or sets) of viable resources, it returns this information to the individual or system that initiated the request, to begin to process of negotiation and resource acquisition.

4. Implementation

The resource broker is implemented as a rule-based agent. The agent use the Jade agent framework [12]. Reasoning is performed by the Jess inference engine [13]; rules defining agent behavior are therefore implemented as Jess rules.

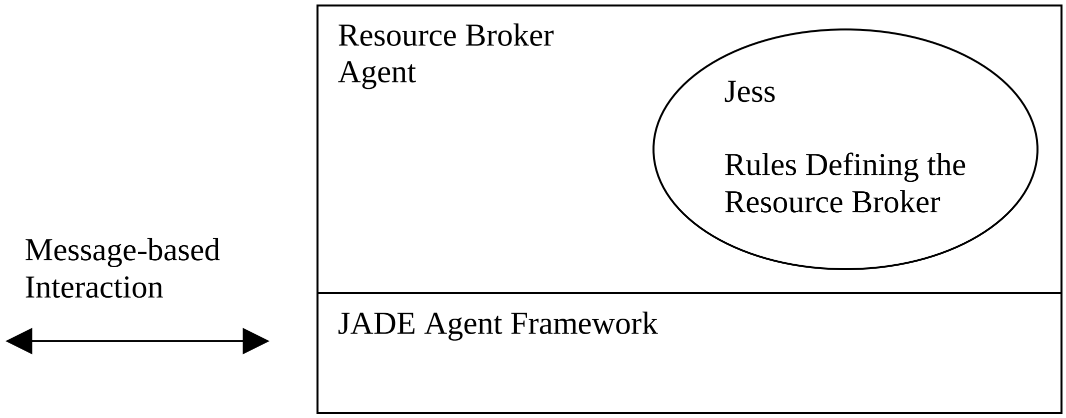

Figure 2: Resource Broker Architecture

Successful resource discovery requires the availability of accurate and sufficient information about the available resources. Yet such information is often incomplete or non-existent, is distributed across a variety of locations, is encoded in different ways, and/or may have varying levels of access restrictions.

To ameliorate these problems, we are developing technologies to facilitate the distribution and maintenance of resource information to create a Dynamic Resource Integration (DRI) framework.

This technology is based on the distribution of proxies, which carry information about resources, and act as proxies to parent resources. Rather than relying on centralized registration and search capabilities alone, resources create DRI agents that autonomously propagate through the network. These agents are locally instantiated, generic, and rule-based; each agent contains the following specification:

- Behavioral rules such as propagation policies and proxy;
- State information such as nodes visited, clients successfully served and queries about required resources; and
- Metadata about the parent resource.

The rules defining the behaviors of each agent are tightly constrained. DRI agents can propagate through the network by sending messages containing these rules, accumulated state and metadata. Upon receipt of a message containing a DRI specification the DRI Framework creates a new DRI agent based on a combination of the original specification augmented by local node policies. When clients query for resources, the DRI Framework returns references to the DRI agents that can best satisfy the immediate resource needs. The clients can then use those DRI agents as proxies to the parent resources.

DRI specifications can also be used to propagate queries for required resources through the network. The DRI Framework updates DRI specifications upon changes in the parent resource, such as the allocation of the resource to a specific task, or metadata changes, reflecting changes in information owned by the resource. The use of the DRI Framework allows metadata about resources to be distributed and updated efficiently throughout the network. Thus, it supports the formation of Dynamic Collaborative Action Teams by providing the most recent information available.

5. Discussion

The resource broker is an important element in the overall effort to facilitate the integration of cross-domain teams, or collections of resources. We have discussed the issue more broadly in other work [2]. Here, we have discussed the process whereby resources are discovered, as a first step in the integration process.

One value of the rule-based approach to TTPs that we have taken is that it facilitates the integration of new data sources, and modification of interaction patterns for individual resources.

An area that we would like to develop further is constraint resolution for selecting the optimal team, or providing sets of viable options. These advances would maximize the value of the effort, and increase the chances of success both in assembling the team, and fielding a response. We would also like to explore the integration of more advanced search technologies, and agent-based resource integration technologies (as described above) in broadening the ability of the system to work across enterprise boundaries. In addition, the integration of semantic web technologies will be essential in enhancing the ability of the system to match roles with resources described in the languages of different domains, via the use of ontologies.

References

[1] Christine N. Salamacha, N. Ray Briscoe and Steve Forsythe, "Managing Dynamic Collaborative Action Teams in a Net-Centric Environment", Presented at the 10th International Command and Control Research and Technology Symposium, San Diego, USA, 2005.

[2] Christine N. Salamacha, Michael Cramer, Timothy Frey and David P. Glock, "Dynamic Collaborative Action Teams: Implementing a Transformational Concept", Submitted to the 11th International Command and Control Research and Technology Symposium, 2006.

[3] Michael Wooldridge and Nicholas R. Jennings, "Intelligent Agents: Theory and Practice", The Knowledge Engineering Review, Vol 10(2), pp. 115-152, 1995.

[4] T. Berners-Lee, J. Hendler, and O. Lassila, "The Semantic Web," Scientific American, May, 2001.

[5] A. Updegrove, "The Semantic Web: An Interview with Tim Berners Lee," Consortium Standards Bulletin, June 2005. http://www.consortiuminfo.org/

[6] T. Gruber, "A Translation Approach to Portable Ontology Specifications," Knowledge Acquisition, vol. 5, pp. 199-220, 1993.

[7] Web Ontology Language (OWL), http://www.w3.org/2004/OWL/.

[8] R. S. Cost, T. Finin, A. Joshi, Y. Peng, C. Nicholas, F. Perich, H. Chen, L. Kagal, Y. Zou, and S. Tolia, "ITtalks: A Case Study in the Semantic Web and DAML+OIL," IEEE Intelligent Systems, January/February, 2002.

[9] L. Li and I. Horrocks, “A software framework for matchmaking based on semantic web technology,” In Proc. of the Twelfth International World Wide Web Conference (WWW 2003), 2003.
[10] T. Finin, J. Mayfield, C. Fink, A. Joshi, and R. S. Cost, “Information retrieval and the semantic web,” Proc. Of the 38th International Conference on System Sciences, 2005.
[11] L. Kagal, T. Finin, A. Joshi, “A Policy Based Approach to Security for the Semantic Web,” In Fensel, D., Sycara, K., and Mylopoulos, J., eds., Proceedings of the 2nd International Semantic Web Conference, volume 2870 of LNCS, 402-418, Springer, 2003.
[12] The Java Agent DEvelopment (JADE) Framework, http://jade.tilab.com/.
[13] E. Friedman-Hill. Jess, the Rule Engine for the Java Platform. http://herzberg.ca.sandia.gov/jess/.