

Every Agent a Web Server, Every Agent Community an Intranet...

Marshall Brinn
BBN Technologies
10 Moulton Street
Cambridge, MA 02138
(617) 873-2717
mbrinn@bbn.com

Todd M. Carrico, Ph.D.
DARPA
3701 N. Fairfax Drive
Arlington, VA 22203
(703) 526-6616
tcarrico@darpa.mil

Nathan Combs
BBN Technologies
10 Moulton Street
Cambridge, MA 02138
(617) 873-4495
ncombs@bbn.com

ABSTRACT

The research community is struggling with the right approach to integrate emerging agent technology with the exploding web page-based Internet. This paper presents the approach used in Cognitive Agent Architecture (Cougaar) under DARPA's Advanced Logistics Project (ALP). In this effort, each agent was constructed with a build-in web server, supported by specialized API for easy creation of modular PlugIns which transforms agent internal object structures into XML and formatted HTML. In addition, the project has developed an efficient set of techniques for transparent proxies to allow any agent entry-point to serve as an Intranet portal to the whole community of agents. Further, as each agent holds only its part of the plan, the underlying infrastructure transparently supports traversal across the boundaries of one agent's plan into that of a different agent. Thus the interfaces support a visualization of the global plan though physically it is a partitioned blackboard distributed among many agents on different machines. The benefit of this approach is that developers can easily build specialized user interfaces into the agent internal plans and state which is accessible through a standard browser by connecting to any agent URL in the society.

1. INTRODUCTION

Cougaar (for Cognitive Agent Architecture) is an innovative software architecture that enables building distributed agent-based applications in a manner that is powerful, expressive, scalable and maintainable. In fact, Cougaar is a code baseline that has successfully demonstrated its utility at constructing dynamic, complex, distributed applications [1][2]. Perhaps of even more significance than the software that implements its concepts, Cougaar represents a methodology, a tried and powerful approach towards designing and building distributed applications.

Cougaar was developed for DARPA (Defense Advanced Research Projects Agency) under the Advanced Logistics Project or ALP and is available Open Source from www.cougaar.org.

Copyright 2001 Association for Computing Machinery. ACM acknowledges that this contribution was authored or co-authored by a contractor or affiliate of the U.S. Government. As such, the Government retains a nonexclusive, royalty-free right to publish or reproduce this article, or to allow others to do so, for Government purposes only.

Agents '01, May 28-June 1, 2001, Montréal, Québec, Canada.

Copyright 2001 ACM 1-58113-326-X/01/0005...\$5.00.

The focus of the Advanced Logistics Project has been developing techniques to better capture and solve the difficult problems of military logistics planning and execution. The scope and complexity of the problem of military logistics is tremendous:

- Millions of different object types to be managed
- Tens of thousands of different interleaved discrete business processes
- Thousands of different organizations with their own physical plants, constraints and user requirements
- Complex, continual interplay between planning and execution
- Over a thousand legacy databases and systems with different data models and protocols

Standard software modeling techniques have proven inadequate to tackle a problem of this size and complexity. Standard software tools are unable to manage an ontology of so many distinct object types. The complexity of developing a standard top-down decomposition of the aggregate business processes involved defies monolithic modeling. Further, the interdependencies among different aspects of the model would tend to make such a model unmaintainable. Logistics systems have tended to treat planning and execution as entirely different realms, with different systems and architectures, due to the potentially chaotic feedback that execution can play on any plan.

Cougaar was developed to tackle these challenges posed by the problems of military logistics head-on. But Cougaar is an approach to building software that transcends the domain of military logistics. Many complex domains can benefit from the approaches pioneered and embodied in Cougaar.

What kinds of problems are well suited to a Cougaar solution? While Cougaar was developed to handle a problem with all the complexities listed above, it is of potential value in any domain bearing any of the above complexities. For example, any of the following problem categories would benefit from being modeled in Cougaar:

- Problem domains that entail hierarchical decomposition and tracking of complex tasks
- Complex application domains involving integration of distributed separate applications and data sources
- Domains involving the generation and maintenance of dynamic plans in the face of execution

- Highly parallel applications with relatively loose-coupling and low-bandwidth communications between parallel streams
- Domains too complex to model monolithically, best modeled by emergent behavior of components

We should note that while Cougaar was designed to address the requirements of military logistics planning, we have seen applications of Cougaar to many different domains—some only tangentially related to military logistics, others completely separate. Nonetheless, this document will contain a flavor of military logistics in many of its examples and illustrations, as this is the domain to which Cougaar has been applied most broadly and successfully. It is important to keep in mind that the Cougaar technology is a domain independent architecture for large scale distributed agent systems.

2. Challenges in Distributed Agent UIs

This poster presented the approach taken in the Cougaar architecture to provide an easy mechanism for the development of content servers in agents. These content servers provide information from the agent’s local PLAN formatted in accordance with its PSP, a small formatting module provided by the developer. In addition to providing the information access mechanisms, the architecture also provides the support infrastructure to allow the crossing of agent information boundaries and the proxy routing of requests to any agent in the society.

The Cougaar UI architecture uses distributed query/response system approach. Each component resides in a cluster and provides information from what is known within that cluster. UI components within a cluster interact with the external UI clients via HTTP connections, allowing for a convenient web or network-based interface. UI developers create customized UI displays or

information streams by creating PlanServerProviders (“PSPs”) which run in the cluster and format and filter information to feed the displays. Each PSP is provided full access to the cluster’s LogPlan, just like a PlugIn, but only executes when a user invokes the PSP via an HTTP request.

If the external UI client is a browser, html can served directly and the PSP is responsible for formatting that data into html formatted text, tables, and lists. For non-browser external UI clients, the PSP is responsible for creating the XML or other structured information streams that can be processed within the client for appropriate formatting and display. Figure 1 illustrates how both approaches can be supported in a single cluster with multiple PSPs.

The combination of these features, when coupled with the capabilities and features of the architecture for planning and execution provide a powerful web-based user interface capability. The society of agents becomes their own intranet serving and fusing information from across the society.

The Cougaar architecture is a general-purpose agent architecture for the construction of large-scale distributed agent systems. The development of logistics unique PlugIns has afforded the construction of a large military logistics command and control prototype for experimentation and demonstration. The architecture itself, without the sensitive military specific PlugIns, is available under OpenSource at: www.cougaar.org. In the OpenSource realm, ALP uses the name Cougaar (Cognitive Agent Architecture) to emphasize the modeling of the human cognitive process inherent in the decomposition of organization processes into component functions as PlugIns.

The Advanced Logistics Program continues to enhance the depth and breadth of its model of various aspects of military logistics. In parallel with and in support of this effort, ALP continues to refine and extend the infrastructure to allow for more successful interoperations between agents.

3. ACKNOWLEDGEMENTS

This work is sponsored by the Defense Advanced Research Projects Agency (DARPA), Tactical Technology Office and is managed under DARPA’s Joint Logistics Technology Office (JLTO).

4. REFERENCES

- [1] Cougaar Architecture Document, ALPINE, www.cougaar.org, Aug 2000.
- [2] Cougaar PlugIn Developer’s Guide, ALPINE, www.cougaar.org, Oct 2000.

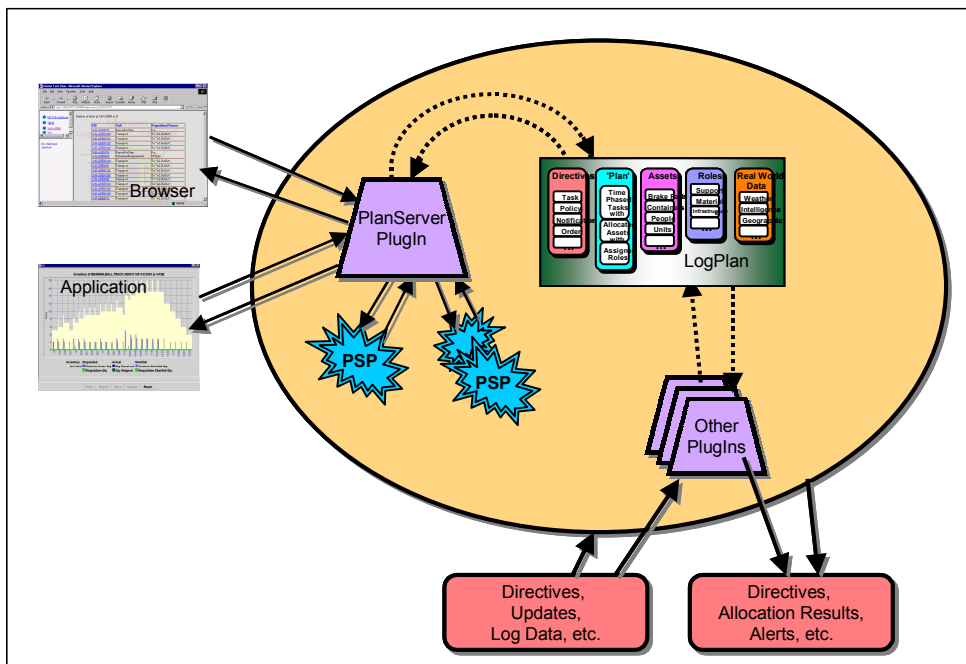


Figure 1. Cougaar UI Architecture