AI Workflow Management in a Collaborative Environment

Michelle Cheatham
Information Directorate, AFRL
michelle.cheatham@wpafb.af.mil

Michael T. Cox
BBN Technologies
mcox@bbn.com

ABSTRACT

Collaborative environments allow geographically distributed groups to work together to generate new knowledge. These systems contain many tools to facilitate collaboration, including workflow management systems (WfMS). WfMS allow multiple agents to work towards achieving a common goal by enabling communication between them. This paper presents a brief overview of collaborative environments in general, and then discusses the distinctive characteristics of current WfMS. We consider the utility of using techniques employed in next-generation grid-based WfMS in collaborative systems that are available today. Specifically, the idea of constructing workflows by applying artificial intelligence planning techniques to a user-specified goal is explored.

KEYWORDS: Workflow Management, AI Planning, Collaborative Environments, Service Oriented Architecture

1. INTRODUCTION

People have been trying to find better and more effective ways to communicate for a long time. Lately, two trends have begun to affect this search. The first is that people need to communicate across vast distances and multiple time zones due to the rise of multinational organizations. The second is that the information that needs to be communicated is more abundant and complex than ever before. These changes have led to the creation and evolution of collaborative systems.

A collaborative environment is a collection of tools that allows a geographically separated group to exchange information and work together effectively. Collaborative environments started out simply -- email is one of the oldest and most frequently used collaborative systems. As people needed to communicate greater amounts of information faster, collaborative systems evolved to include other types of tools. Today's collaborative environments are scalable frameworks that contain facilities for document management, audio-video conferencing, web-based portals, and workflows, among others.

This paper will focus on workflow management systems (WfMS), which are a component of many contemporary collaborative environments. A workflow is a series of operators chained together to accomplish a goal. An example is the process a company goes through when ordering new inventory. Steps in the process might include collecting cost estimates, choosing a vendor, ordering the product, and testing it on arrival, and adding the item to the company's internal inventory tracking system. As the number and diversity of operators available for use in workflows increases, it becomes more difficult to know what services are available and how they can be combined to solve a given problem. Researchers involved in next-generation grid-based collaborative systems have suggested using AI planning techniques to help automate workflow creation [2, 5, 10]. This paper will consider whether this approach can also be applied to the workflow management systems available in current collaborative environments.

The remainder of the paper is organized as follows: Section 2 provides a brief overview of collaborative environments, focusing on workflow management systems in integrated collaborative systems; Section 3 motivates the need for AI planning in WfMS and discusses how the characteristics of current WfMS influence the use of AI planning strategies in this domain; a preliminary implementation of a planning-based workflow generator in a commercial collaborative environment is shown in Section 4; conclusions and future work are covered in Section 5.

2. BACKGROUND

McQuay proposes a useful framework for discussing collaborative systems in [9]. He divides them into four types: *standalone*, *federated*, *integrated*, and *grid*. Standalone systems are those that facilitate asynchronous human-to-human communication. Examples include email, shared calendars, and newsgroups. These are the oldest types of collaborative systems, and they have the significant benefit of being utilized by many people as part of their everyday work. As Smith points out in [12], the usefulness of many state-of-the-art collaborative environments is limited by a lack of user adoption.

As standalone collaborative systems became commonplace, the need for tools that would allow groups of people to communicate synchronously with one another became apparent. Federated systems, which typically combine chat, shared whiteboard, and video teleconferencing, were devised to allow real-time communication. Possibly the most well-known federated system is Microsoft NetMeeting, which provides all of the previously mentioned services. While these systems provide a greater variety of communication mechanisms than standalone systems, the tools making up federated systems are generally tightly coupled and have limited extensibility and customization options.

It is often pointed out that much of the data available to us today is useless without the proper tools to view and analyze it [12]. This realization has led to the development of integrated collaborative systems, which are based on open, scalable architectures that allow new tools to be added to support simulation, prediction, and decision making. Integrated environments allow both synchronous and asynchronous communication between humans, machines, and combinations of the two.

Most collaborative environments available today fall into the "integrated" category in McQuay's framework and include the same basic set of components. This set includes tools from earlier types of collaborative systems, such as threaded discussions, audio-video conferencing, chat, and shared whiteboard. Another tool present in many current collaborative environments is a web-based portal that provides users with a personalized view of information that is relevant to them, including project deadlines and milestones, news, links to important information, and a shared calendar. A document management facility is also usually available that provides version control tools and allows users to create, search, view, and edit files. Documents managed by a collaborative environment are typically accessed as if they resided on a networked storage device; they are guaranteed to be accessible by all members of a group, regardless of their location. Some systems also provide access to domain-specific tools that can be used to view data or run simulations.

The aspect of current collaborative environments that will be the focus of the remainder of this paper is the workflow management system (WfMS). Most integrated collaborative environments provide a means to create and run workflows. The types of operators that are available for use in these workflows vary from simple, pre-defined activities such as sending an email, to free-form activities written in programming languages such as Java or TCL, to complete domain-specific applications, such as modeling and simulation tools. Workflows created with these systems typically represent business processes that are relatively static and consist primarily of human operators. An example is the series of steps a customer service department goes through when an item is returned. More recently, workflow technology has been used to facilitate simulation, prediction, and decision support by chaining various modeling and simulation tools together to analyze the vast amounts of data that are available to an organization. The WfMS in current collaborative environments have some capability to support these more advanced workflows by providing wrapper code to integrate the required tools into the collaborative framework. In fact, so many tools may be available that finding and ordering the services needed to achieve a specific goal may become difficult. As mentioned previously, the idea of using techniques from the field of AI planning to address this issue has been proposed. We will examine the utility of applying these AI planning concepts to the WfMS available in current collaborative environments.

3. AI PLANNING IN WORKFLOW MANAGEMENT SYSTEMS

When a user of a collaborative system needs to create a workflow, she is faced with a problem. The user understands the problem that she needs solved, but she may not know what steps to take in order to actually solve it. There may be dozens of operators available in the collaborative environment, and the user would have to know the specifics of each one – its inputs, outputs, pre-conditions, and post-conditions – to know which operators to choose for this particular workflow. In addition, the user would need to be proficient with the workflow development environment and the underlying middleware in order to create the workflow. Because of this, most workflows are actually created by software developers instead of end users. This is not an ideal situation because the end user must wait for a developer to become available and then explain the problem she is trying to solve to a software developer who may have little knowledge of the problem domain. Workflow management systems would be more valuable if end users could create basic workflows without the assistance of a software developer.

One way to move towards automatic creation of workflows is to use AI planning techniques. The goal is to allow end users to specify *what* they want the workflow to achieve, instead of *how* to achieve it. This is accomplished by representing each operator available in the collaborative environment in terms of a planning language. The language is used to describe the state of the system before the operator is executed, and the state that results after the operator has finished. The user can then specify a goal state, and the planning software will find a sequence of operators that will reach the goal. For example, assume that a system has three operators: Operator A looks up the email addresses for a list of users, Operator B takes a set of email addresses and sends email to each one, and Operator C is a simulator. If a user's goal is to send a project status update to everyone in the enterprise, the planner would create a workflow consisting of Operator A followed by Operator B.

This approach makes it easier for end users to create workflows. It also results in more dynamic and fault tolerant workflows. When workflows are created manually by software developers, if an operator becomes unavailable the developer must hand-edit each workflow that uses that operator and replace it with a substitute. Similarly, if a new operator becomes available, developers must revisit each workflow that could make use of the new operator. If AI planning is used, the planner can simply be run again with the same goal, and the newly generated workflow will take advantage of all operators currently available in the collaborative environment and avoid any that are no longer available.

3.1. A. Feasibility of AI Planning in Collaborative Environments

There are several concerns that arise when using AI planners to solve real-world problems. The first and most important is that the size of the search space may overwhelm the planner, in which case the planner is unable to generate solutions in a timely manner. Another consideration is the complexity of the language used to describe the available operators. The planning language must be expressive enough to describe the relationships, capabilities, and trade-offs of the operators, while at the same time reflect the vocabulary of the problem domain so end users are comfortable with it [5]. Current research suggests possible solutions to both of these concerns. A large search space can be coped with by codifying business rules to guide the search process [14] by using templates or a plan library as a starting point [2], or by taking a mixed-initiative approach [6]. There has also been work related to simplifying the planning language by using two separate ontologies to describe problems: one for the domain-specific concepts and another for the planning concepts [2]. However, both of these issues are less of a concern in current collaborative environments.

The number of operators available in a single collaborative environment is likely to be small enough for a standard AI planner to handle efficiently using a relatively simple planning language. Most collaborative environments are organized around enterprises or communities of interest, which are focused on a single topic [7, 12]. Workflows created in these environments will consist of operators specific to this topic or from a limited collection of generic operators. This set of available operators is unlikely to be large enough to confound the planner. In addition, most current workflow management systems have not been designed with inter-operability in mind [14], which limits the possible operators to those within a single organization. This is largely due to the reluctance of most commercial organizations to use operators provided by sources external to the company. Therefore, the planning language needs to describe only a relatively narrow set of operators and does not need to be expressive enough to convey characteristics related to quality of service and trust concerns.

3.2. Difficulties Related to Collaborative Environments

The greatest challenge in incorporating AI planning techniques into existing workflow management systems is the way that individual workflow nodes have been developed. Most existing software has been written using an object oriented paradigm. In addition, current workflow systems typically use the same integrated development environment (IDE) both to add new nodes to the collaborative environment and to chain the nodes together to create workflows. This leads to workflows consisting of tightly coupled nodes that have a low possibility of being reused in workflows other than the one for which they were originally designed.

Object oriented programming has been extremely popular for more than a decade. However, operators within a workflow need to be closer to services than objects. Both services and objects are loosely coupled, but services encompass complete business functions and are meant to be reused in configurations not thought of when the services were originally developed [11]. Current systems will need to be moved from object oriented to service oriented architectures (SOA). Creating the proper services when starting from monolithic legacy systems is not always an easy or straightforward task. Moving to a service oriented architecture requires identifying which business functions should be exposed as services, determining the proper interfaces for these services, and finding the underlying code necessary to implement them. Because services represent complete business functions, the code to implement them may need to be integrated from pieces in several different applications [11].

As mentioned previously, using a single IDE to create both new operators and new workflows creates a temptation for developers to create "glue" nodes that are tightly coupled to other nodes in the workflow. This breaks the SOA paradigm and results in workflow nodes that are less reusable. As part of our work regarding using AI planning in current WfMS, we are also designing a separate interface to create new workflow operators independent of any specific workflow. This will emphasize the ideal of developing services that are generic enough to be used in many different circumstances. In addition, this new IDE will provide mechanisms to manage the planning language used to describe the workflow nodes in order to maintain consistency.

4. IMPLEMENTATION

In order to gather first-hand experience using AI planning techniques in an integrated collaborative environment, we have used PRODIGY [1] [13], a state space planner, to implement a workflow generation portlet within the KnowledgeKinetics™ framework. This portlet is a proof of concept; a more robust implementation will be part of our future work in this area.

KnowledgeKinetics™ [7] is a collaborative environment developed and commercialized by Ball Aerospace and the Air Force Research Laboratory Collaborative Technology and Applications Branch. The collaborative environment is meant to allow geographically distributed teams to collaborate on projects and decision support ranging from product design to research. KnowledgeKinetics™ is organized around *enterprises*, which are similar to the "communities of interest" discussed by Smith in [12]. The enterprise is a grouping of all the documents, data, people, schedules, and tools related to a project. It is accessed via a web portal.

The workflow system within KnowledgeKinetics™ supports both human and software operators. KnowledgeKinetics™ is based on the J2EE platform; software operators may be written in any programming language, but Java wrappers must be created for them to function within the WfMS. The human operators are integrated into the collaborative framework and can monitor the user's interactions with entities inside the portal. For example, human operators include actions such as a user filling out a form, approving/choosing an option, or uploading a document. When a developer

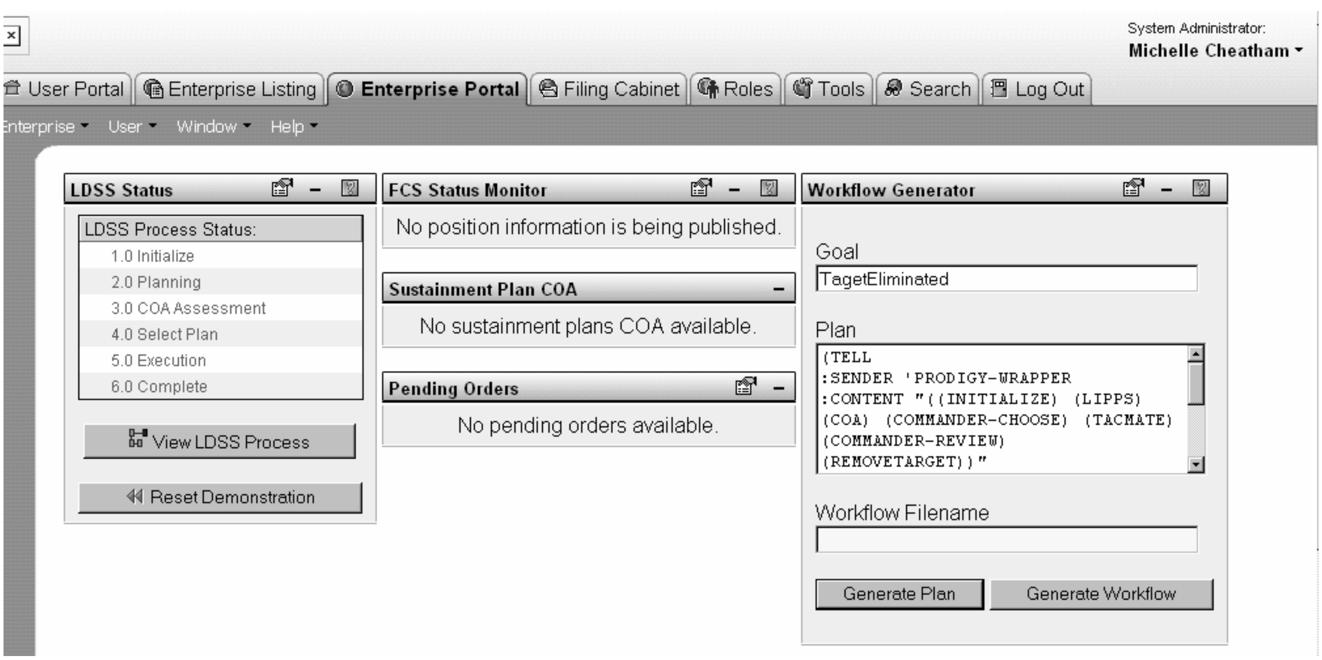

Figure 1. Combat Decision Support System Portal

creates a workflow, she first checks to see that all necessary operators are available. If not, additional applications are integrated into the system. Once all of the required operators are available, they are dragged into place using the workflow integrated development environment (IDE), along with process control nodes such as conditional branches, loops, and parallel series. Nodes in the workflow are connected by joining the outputs of some to the inputs of others.

KnowledgeKinetics™ exemplifies many of the characteristics of integrated collaborative environment WfMS discussed previously. The system supports both human and software operators. Workflows in the system are a blend between static and dynamic: some workflows represent standard business processes that seldom change, such as travel expense approval; others are more dynamic in nature, such as those created to chain together simulation tools to do what-if analyses. Software developers are required to create all but the simplest workflows due to the knowledge required about each of the available operators and the need to write scripts that act as "glue" by passing information between some workflow nodes. In addition, the KnowledgeKinetics™ server acts as a broker between all of the agents in the system. Finally, all existing KnowledgeKinetics™ workflows use agents belonging to an individual organization.

There are many different AI planners available (see [10]). PRODIGY, a domain-independent state space planning tool, was chosen for this implementation. PRODIGY has a partial order planning mode – in addition to finding a sequence of operators to achieve a given goal, it is also capable of recognizing when some operators can be executed in parallel.

The workflow generation tool we have implemented has been applied to a prototype Combat Decision Support System (CDSS). The CDSS enterprise was developed several years ago as a proof of concept demonstration of the kind of assistance that a sophisticated collaborative environment could provide to the military with respect to command and control operations. The CDSS portal serves as a focal point for a commander monitoring a battle. There are portlets available to plan a battle, simulate the plan, issue orders, monitor assets, and watch the battle unfold. Workflow nodes to support these activities, as well as standard KnowledgeKinetics™ operators, such as sending a notification message to a user, getting a user to approve a proposal, and tasking a user to fill out an online form, also exist within CDSS. By choosing the CDSS enterprise as our implementation target, we were able to examine the issues arising from attempting to retrofit an existing system to take advantage of AI planning techniques.

In order to use the planner to create workflows, details concerning pre- and post-conditions for each workflow operator were added to the information that KnowledgeKinetics™ already stores about all operators available within the system. An example of our node representation is shown in Figure 2. The operator in the example is a software tool that analyzes a set of alternative courses of action. This operator takes as input a set of potential plans and returns a risk analysis of each one. The *Resource Name* and *Resource Key* fields indicate which software agent provides this action. The *preconds* and *effects* sections indicate to the planner that this operator can be applied only after a set of plans have been created and will result in each plan within the set being evaluated. For a more thorough discussion of the PRODIGY section of the operator definition, see [1].

A screenshot of the CDSS portal is shown in Figure 1. The user interface for our system is the Workflow Generator portlet located on the right side of the portal. Using a dynamic help system that displays all achievable goals within this KnowledgeKinetics™ enterprise, the user enters the goal of the workflow into the text box at the top of this portlet. More complex goals can be made by joining individual goals with Boolean operators. The *Generate Plan* button causes the portlet to connect to the PRODIGY server and retrieve the correct sequence of operators to achieve the user's goal. The underlying technology used to accomplish this is discussed in [3] and

```
(OPERATOR COA
; Type | Resource Activity
; Name | COA
; Attributes
; Resource Name | COA Assessment
; Resource Key |
AgentProxyHome.MyCommunity.1089638838578
; Inputs
; plans | java.util.Vector
; Outputs
; risk | java.util.Hashtable
  (params <planset>)
  (preconds
    ((<planset> SETOFPLANS))
      (forall ((<plan> (and PLAN
        (gen-from-pred (memberOf <plan>
                        <planset>)))))
      (created <plan>))
  )
  (effects
    ()
    ((add (evaluated <planset>))
  )
))
```

Figure 2. Operator Representation

[4]. The KQML message containing the resulting plan is then displayed in the portlet [8]. When the user clicks on the *Generate Workflow* button in the lower right corner of the portlet, this plan is translated to an executable KnowledgeKinetics™ workflow, shown in Figure 3.

5. CONCLUSIONS AND FUTURE WORK

This paper has illustrated how the unique characteristics of workflow management systems within integrated collaborative environments – a combination of human and software agents, the limited scope of domains, a centralized architecture, and agents located within the boundaries of the organization – influence the use of AI planning techniques to facilitate workflow generation by the end users of these collaborative systems. In addition to reducing the need for software developers to create every workflow in a system, the knowledge captured by describing all of the available operators in terms of a planning language can also be used in aspects of the collaborative environment beyond the workflow management system. In particular, representing the human elements of the system using a planning language opens the possibility of adjusting the user's view of the portal based on her current goals. Opportunities such as this will be explored as part of our future work in this area. In addition, we will examine other possible uses of AI techniques within collaborative environments. For instance, the advantages of creating agents within workflow management systems that are more goal-centric could be considered, along with the possibility of integrating them by analyzing goal and subgoal relationships rather than by using domain specific knowledge and constraints.

Many researchers predict that next generation collaborative systems will have grid architectures. These systems will use web services and other emerging technologies to facilitate communication among a greater variety of operators than what is supported by integrated environments. With so many operators available, the need for techniques to find and order the services needed for a particular application will be even greater. Work involving the use of AI planning in integrated collaborative environments is not only useful today; it also provides important lessons that will be valuable when the next generation of collaborative systems arrives.

REFERENCES

[1] Carbonell, J. G. et al., PRODIGY4.0: THE MANUAL AND TUTORIAL, Technical Report CMU-CS-92-150. Computer Science Department., Carnegie Mellon University, 1992.

[2] Chung, P.W.H., L. Cheung, J. Stader, P. Jarvis, J. Moore, and A. Macintosh, "Knowledge-based Process Management – An Approach to Handling the Adaptive Workflow," Knowledge-Based Systems, Vol. 16, 2003, pp. 149-160.

[3] Cox, M. T., G. Edwin, K, Balasubramanian, and M. Elahi, "Multiagent Goal Transformation and Mixed-Initiative Planning Using Prodigy/Agent," Proceedings of the 4th International Multiconference on Systemics, Cybernetics and Informatics, Vol. 7, 2001, pp. 1-6.

[4] Elahi, M. M., A DISTRIBUTED PLANNING APPROACH USING MULTIAGENT GOAL TRANSFORMATIONS, Masters Dissertation, Wright State University, Computer Science and Engineering Department, Dayton, OH, 2003.

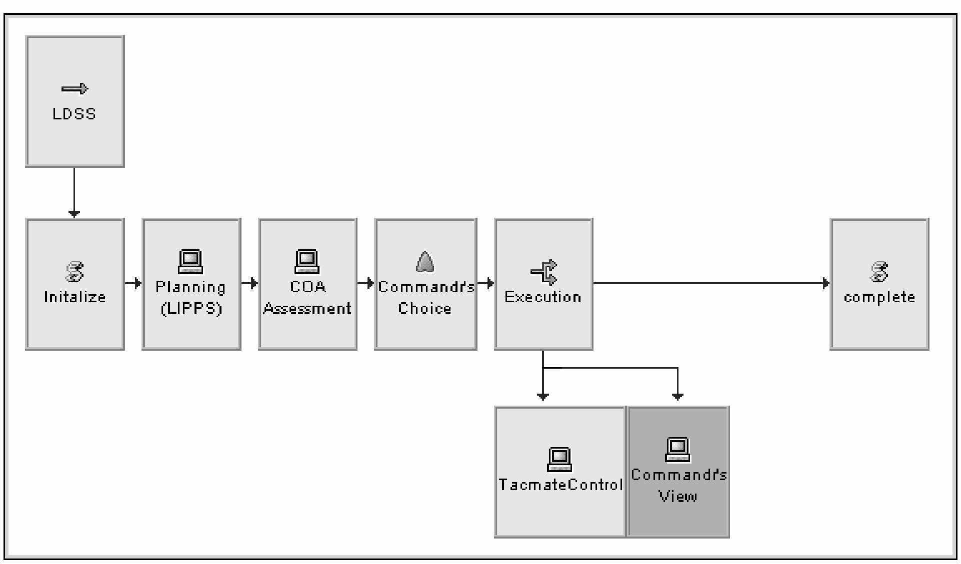

Figure 3. Ready-to-Execute Workflow

[5] Gil, Yolanda, E. Deelman, J. Blythe, C. Kesselman, and H. Tangmunarunkit, “Artificial Intelligence and Grids: Workflow Planning and Beyond,” IEEE Intelligent Systems, 2004, pp. 26-33.

[6] Kim, Jihie, M. Spraragen, and Y. Gil, “An Intelligent Assistant for Interactive Workflow Composition,” International Conference on Intelligent User Interfaces, ACM, Funchal, Madeira, Portugal, January 13-16, 2004.

[7] KnowledgeKinetics™ homepage, https://k2.knowledgekinetics.info

[8] Knowledge Query and Manipulation Language (KQML) specification, http://www.cs.umbc.edu/kqml/kqmlspec/spec.html

[9] McQuay, William K. “The Collaboration Grid: Trends for Next Generation Distributed Collaborative Environments,” Proceedings of SPIE Enabling Technologies for Simulation Science Conference, Defense & Security Symposium, Orlando, Florida, USA, April 12-16, 2004.

[10] Moreno, M.D.R. and P. Kearney, “Integrating AI planning techniques with workflow management system,” Knowledge-Based Systems, Vol. 5, 2002, pp. 285-291.

[11] Papazoglou, Mike P, “Service-Oriented Computing: Concepts, Characteristics, and Directions,” Proceedings of the Fourth International Conference on Web Information Systems Engineering, IEEE, Rome, Italy, December 10-12, 2003.

[12] Smith, Reid G. and A. Farquhar, “The Road Ahead for Knowledge Management: An AI Perspective,” AI Magazine, Winter 2000, pp. 17-40.

[13] Veloso, M. M., J. Carbonell, A. Perez, D. Borrajo, E. Fink, and J. Blythe, “Integrating Planning and Learning: The PRODIGY Architecture,” Journal of Theoretical and Experimental Artificial Intelligence, Vol. 7, No. 1, 1995, pp. 81-120.

[14] Yang, Jian, M. Papazoglou, B. Orriens, and W. van Heuvel, “A Rule Based Approach to the Service Composition Life-Cycle,” Proceedings of the Fourth International Conference on Web Information Systems Engineering, IEEE, Rome, Italy, December 10-12, 2003.