Automating distributed workflow for electronic commerce: A model for building meta-workflow components

M. Gill*, M. Goul, D. Gregg, T.S. Raghu, & D. Schuff
School of Accountancy and Information Management, Arizona State University

Business software development environments for web-based applications are at a very early stage in their potential lifecycle. The purpose of this research agenda is to develop a model for how business processes can be interpreted into primitive computer codes for web applications. We have taken a transaction perspective adapted from the distributed database approach to maintenance of integrity assuming the concepts of ACIDity (Atomicity, Consistency, Isolation, and Durability) and serializability. This perspective is consistent with emerging literature on "workflow agents" [e.g., Huhns and Singh, 1998]. In this paper, we design, develop, and show an example of a meta-level set of building blocks for a direct mapping between electronic commerce and workflow processes. To validate the sufficiency and completeness of the meta-level components proposed, a complete mapping of an electronic commerce application to meta-workflow components is proposed.

Background

Our increasingly competitive and technology-driven world has decreased the time available for most business activities. To survive in this environment, organizations are increasingly turning to artificial intelligence-based computer technologies, like intelligent agents, to perform many of the more time consuming business tasks [O'Leary, 1997]. Intelligent agents are a class of software entities that carry out a set of operations on behalf of a user with some degree of independence or autonomy, and, in doing so, employ some knowledge or representation of the user's goals or desires [Etzioni, 1997, Maes, 1994].

One application for agent technology includes agents designed to facilitate cooperation among humans working on large projects. For example, an intelligent agent system has been used to streamline the design process conducted by teams of problem solvers [Tan, Hayes and Shaw, 1996].

Similar intelligent agents have been proposed for use in workflows. A workflow is "the automation of a business process, in whole or in part, during which documents, information, or tasks are passed from one participant to another for action, according to a set of procedural rules" [Lawrence, 1997]. Workflows are complex dynamic processes that can span corporate boundaries. Electronic Commerce (EC) transactions can be defined using workflow processes. An example of an EC workflow is web-based order processing in the context of a "build-to-order" product. For instance, a customer receives product information from an on-line company and then places an order. The company receives the order, the payment source is verified, inventory is checked and the product is shipped. Conceivably, several organizations could participate in the processing of the orders. Huhns and Singh show that intelligent agents are ideally suited to this type of environment because agents can be designed to coordinate workflow activities. For example, agents can be used to negotiate with each other to ensure that global workflow constraints are not violated. They can also be designed to identify different types of exception conditions and to learn appropriate behavior from repeated instances with the same kinds of exceptions.

Workflow standards have been proposed by the Workflow Management Coalition in their "Workflow API" [Lawrence, 1997]. However, these standards are comprised of a series of granular, low level primitives. These primitives primarily deal with implementation details and are a complex and large set of functions. The functions can be used to implement workflow agents. The next section shows a higher level mapping of these workflow primitives that will facilitate the creation of workflow agents in a manner that can be directly tied to business processes.

Mapping WAPI primitive groupings to business processes

Electronic Commerce (EC) transactions can be defined using workflow processes. Mapping these workflows to the WAPI meta-primitives will facilitate the creation of EC development tools for applications such as "build-to-order" processing. With the growing importance of online purchasing, the ability to customize an order is becoming increasingly important. As an example of the concepts developed in this paper, a script describing the "build-to-order" process is used to demonstrate a framework upon which the primitives can be mapped. The script for the example process is shown in Figure 1. The WAPI primitives are grouped into aggregate meta-primitives, organized by function, with the interrelationships shown in

* corresponding author: mark.gill@asu.edu

Script for "Build-to-Order" Workflow

Order Placement Process

1) URL request made via Web
2) Send order form to user
3) Verify order completeness
4) Receive order form from user
5) Check credit information
6) If (credit is bad) then
7) Reject order
8) Else
9) Create the order request
10) Determine internal/external supply sources
11) Check inventory for each supply source
12) Check schedule dates for each supply source
13) Send confirmation request to the customer
14) Receive response
15) If (response is reject order) then
16) Return
17) Else
18) Complete order placement process (confirmed)

Figure 1 – Sample EC Script

Figure 2 [Lawrence, 1997]. Those functions are:

1) **Connection** – Manage the communication link between hosts
   a) Start communication
   b) Stop communication
2) **Process control** – Manage the processes to be executed by the system
   a) Check process availability
   b) Instantiate process
3) **Activity control** – Manage the activities to be carried out by each process
   a) Start activity
   b) Stop activity
4) **Process status** – Monitor the status of currently running processes
   a) Check if process is complete
   b) Check if all associated processes are complete
5) **Activity status –** Monitor the status of currently running activities
   a) Check if activity is complete
   b) Check if all associated activities are complete
6) **Work list** – Manage the allocation of system resources
   a) Assign activities to resources
   b) Maintain list of open activities
7) **Administrative** – Top level management of all system processes and activities
   a) Manage process
   b) Terminate activities and processes

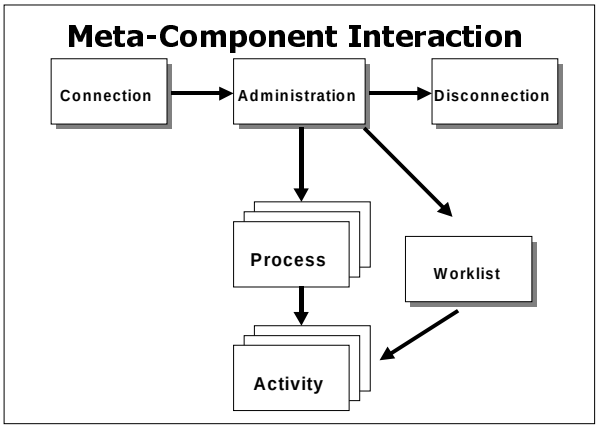

Figure 2 – Meta-Component Interaction

Attempting to map steps in the script to WAPI primitives clarifies the need for meta-primitives. For example, *Step 11: Inventory Check* of our Build-to-Order Workflow, a check to verify a product in the inventory of a supplier, can be broken down into more granular steps. Each one of those steps can

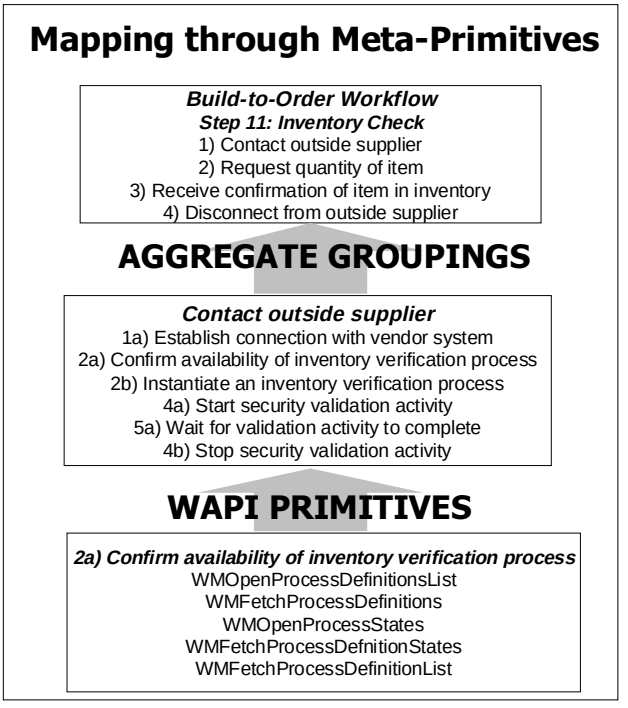

Figure 3 – Mapping through Meta-primitives

further be broken down into a series of low-level primitives as defined by WAPI. While the level of detail provided by the WAPI primitives is comprehensive, it would require significant effort to use these primitives to describe business processes.

Figure 3 illustrates how the meta-primitive layer can be used to describe a business process. Instead of representing each step in the process by a series of

primitives, an intermediate layer is used to describe the major functions the primitives will perform. Grouping the low-level primitives into meta-primitives facilitates the construction of an outline that can closely mirror the actual business process of the organization.

Conclusion

The contribution of this work is that we have integrated recent research in intelligent agents, electronic commerce, and workflows to develop a model for how business processes can be interpreted into primitive computer codes for web application development. The Web-BPAD (Business Process Application Development) Group at Arizona State University is currently developing a higher-level set of scripts that describe fundamental EC processes.

Etzioni, Oren, Moving Up the Information Food Chain: Deploying Softbots on the World Wide Web, *AI Magazine*, v18n2, Summer, 1997, 11-18,

Huhns, Micharl N. and Singh, Munidar P., Workflow Agents, *IEEE Internet Computing*, 2(4), July-August, 1998, 94-96.

Lawrence, Peter (ed), Workflow Handbook 1997, published in association with the Workflow Management Coalition, John Wiley & Sons: Chichester, England, 1997.

Maes, Pattie, Agents that Reduce Work and Information Overload, *Communications of the ACM*, v37n7, July, 1994, 31-40.

O'Leary, Daniel E., The Internet, Intranets, and the AI Renaissance, *Computer*, , January, 1997, 71-78

Tan, Gek Woo; Hayes, Caroline C. and Shaw, Michael, An Intelligent-Agent Framework for Concurrent Product Design and Planning, *IEEE Transactions on Engineering Management*, v43n3, August, 1996, 297-306.