WORKFLOW MANAGEMENT FROM A SCHEDULING PERSPECTIVE

Stephen F. Smith and David W. Hildum and Marcel A. Becker
Intelligent Coordination and Logistics Laboratory
The Robotics Institute
Carnegie Mellon University
Pittsburgh PA 15213-3890
412.268.{8811,2357,8164}
{SFS,HILDUM,MB81}@CS.CMU.EDU

Abstract

In this paper, we examine the workflow management process from a scheduling perspective. Recognizing that effective workflow management requires an ability to efficiently allocate limited resources to tasks over time, we concentrate on characterizing this domain as a continuous distributed scheduling problem and on understanding the requirements and opportunities for providing workflow scheduling support within multi-agent environments. Our goals are twofold: (1) to relate the characteristics of the workflow management problem to scheduling models previously developed for other domains, and (2) to identify the issues and challenges surrounding application of previously developed scheduling technology to this problem.

Introduction

In this paper, we examine the workflow management (WFM) process from a scheduling perspective. Recognizing that effective workflow management requires an ability to efficiently allocate limited resources to tasks over time, we concentrate on (1) characterizing this domain as a continuous distributed scheduling problem and (2) understanding the requirements and opportunities for providing workflow scheduling support within multi-agent environments. Our goals are twofold: (1) to relate the characteristics of the WFM problem to scheduling models previously developed for other domains, and (2) to identify the issues and challenges surrounding application of previously developed scheduling technology to the WFM problem. Our starting hypothesis is that the concepts, structure, and constraints that define this scheduling problem are not substantially different from those of other domains. But although we expect commonality in modeling and scheduling requirements with respect to previously addressed application domains, there are likely also unique (or perhaps previously under-emphasized) scheduling issues and constraints that constitute solution drivers in the workflow domain.

Our longer-term focus is application of the scheduling capabilities currently embodied in the OZONE[1] application framework (Smith, Lassila, & Becker 1996) to the workflow management domain. OZONE assumes a dynamic, constraint-based problem-solving model and provides three high-level components: (1) a library of primitives (organized as a class library) for specifying domain constraints and constructing domain models, (2) a companion set of problem-solving components, including constraint-propagation/management tools, search components and procedures, a set of scheduling and rescheduling methods and heuristics, etc., and (3) a blackboard-based control infra-structure for configuring and arbitrating the use of different scheduling and analysis tools.

Methodologically, our approach to analyzing the WFM domain is to consider the principal elements of the workflow scheduling problem in terms of the OZONE scheduling "ontology" (Smith & Becker 1997). The OZONE ontology provides a conceptual framework for mapping the characteristics and constraints of a given application environment into the concepts and techniques provided by the class library, and hence a conceptual framework for constructing executable domain models (i.e., working scheduling applications). As such, it gives an indication of what information must be determinable in a given application domain for application of correspondent scheduling capabilities to be viable.

Characterizing the Workflow Scheduling Domain

In this section, we utilize the OZONE scheduling ontology to develop a domain ontology for workflow management within multi-agent environments. We proceed by introducing the OZONE scheduling ontology and then mapping the basic workflow domain components onto it. Finally, we discuss the inherently distributed nature of the WFM problem and its amenability to a multi-agent solution approach.

The OZONE Scheduling Ontology

The OZONE scheduling ontology adopts an activity-centered modeling viewpoint and is biased towards constraint-based solution approaches. Scheduling is defined as a process of feasibly synchronizing the use of RESOURCES by ACTIVITIES to satisfy

---

The work described in this report was sponsored by Teknowledge Corporation under a subcontract within their larger "JFACC-after-Next Workflow Management" Applications Services contract in DARPA's JFACC program. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Advanced Research Projects Agency or the United States Government.

[1] OZONE = O^3 = Object-Oriented OPIS. OPIS (OPportunistic, Intelligent Scheduler) was an earlier, constraint-based scheduling architecture (Smith 1994) developed at Carnegie Mellon University.

DEMANDS over time, and application problems are described in terms of this abstract domain model.[2] Figure 1 illustrates the base concepts involved and their structural relationships.

A DEMAND is an input request for one or more PRODUCTS, which designate the goods or services required. The DEMAND is the interface that allows an external client to state the objective to be achieved as well as certain user-specified restrictions and/or preferences on this objective. The objective specified in the DEMAND is the expected output of the system. These expected system outputs are the PRODUCTS. The product can be a physical entity, the satisfaction of some conceptual specification like the evaluation of feasibility, or even a more abstract goal that has no actual physical meaning.

The ability to generate the expected output according to specifications is a property of the system. A computer-manufacturing company can assemble only a certain range of configurations, a transportation company can support only certain types of cargo, and a workflow participant can perform only certain kinds of tasks. The different types of objectives that can be accomplished by the system characterize the set of PRODUCTS available to the user of that particular system. In the OZONE ontology, the PRODUCT entity represents the knowledge required by the scheduler to generate an executable schedule which results in its production. A scheduling system does not generate any physical object nor produce any change in the real world. The production, transportation and workflow management systems are the entities responsible for the actual accomplishment of the objective. In a scheduling system, the PRODUCT encodes the internal information about resources and physical characteristics of the process which, when combined with the context provided by an input DEMAND, allows the generation of a set of process and resource requirements over time. Therefore, the PRODUCT can be seen as the template plan for accomplishing a certain goal or a certain set of goals. The DEMAND provides the parameters that map this prototypical plan into an ACTIVITY network that when executed should accomplish the specified objective.

The satisfaction of DEMANDS centers around the execution of ACTIVITIES. An ACTIVITY is a process that uses RESOURCES to produce goods or provide services. An ACTIVITY can only be executed if certain conditions, like resource availability, are satisfied. The execution of an ACTIVITY produces changes in the state of the real world. Notice that although PRODUCTS are *produced* as a result of the execution of ACTIVITIES, they play a different role in the OZONE ontology. They represent the set of valid objectives that can be specified in a DEMAND: the set of objectives the system knows that can be satisfied with the set of RESOURCES available. The PRODUCT entity acts more as a link, connecting DEMANDS to ACTIVITIES through RESOURCES, than as a means of describing the result of executing activities. When a DEMAND is input into the scheduling system, the PRODUCT has the information necessary to create a process plan that when executed should produce the required product. The scheduler merely allocates time on the RESOURCES specified in the plan.

The use of RESOURCES and the execution of ACTIVITIES is restricted by a set of CONSTRAINTS. These CONSTRAINTS can be specified by the DEMAND, like release date and due date, can be inherent to the PRODUCT characteristics, like technological restrictions and design parameters, or can be a result of the RESOURCE limitations, like capacity, speed and accuracy.

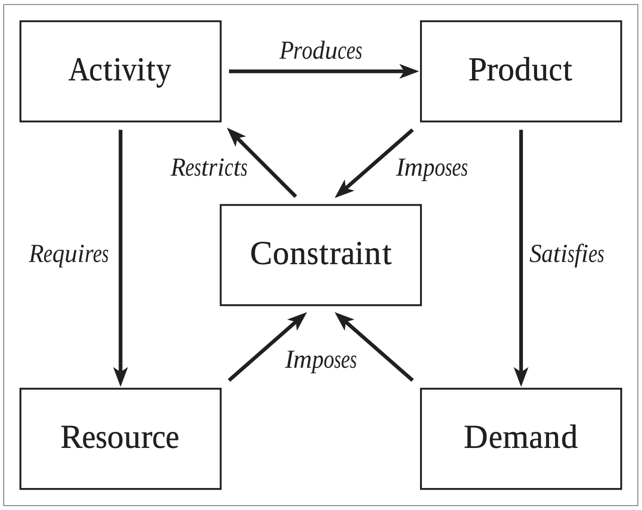

Figure 1: OZONE Abstract Domain Model

These five basic concepts of the ontology—ACTIVITY, RESOURCE, DEMAND, PRODUCT and CONSTRAINT—together with the interrelationships depicted in Figure 1, define an abstract model of a scheduling domain and a framework for analyzing and describing particular application environments. Associated with each concept definition are terminologies for describing basic properties and capabilities. Properties define attributes or parameters of relevance to specifying an executable scheduling model. The abstract model and its properties are extensible through concept specializations to define more specific models for various subdomains.

Workflow Domain Components

The process of workflow management is focused on the passing of tasks and information between participants who perform actions to (1) enable further activities and (2) contribute to the production of the final product. Workflow activities can be grouped into WORKFLOW PROCESSES (akin to OZONE's ACTIVITY networks), which when executed, are intended to achieve a particular WORKFLOW OBJECTIVE (or PRODUCT).

The primary RESOURCES in a workflow domain are WORKFLOW PARTICIPANTS and ARTIFACTS. WORKFLOW PARTICIPANTS are *individuals* or *teams* of individuals whose membership can vary over time. They are schedulable RESOURCES subject to the usual capability and capacity constraints. An ARTIFACT, like (for example) the draft of a document, is produced through the execution of one or more workflow ACTIVITIES. ARTIFACTS are typically *infinite-capacity* RESOURCES (i.e., they can be copied and distributed freely to ACTIVITIES as necessary), and it is therefore not often necessary to consider their allocation to other ACTIVITIES. Their basic role is that of *enablers* of other ACTIVITIES. Secondary resources, like (for

---

[2] By convention, we use capitalization to distinguish specific concepts that are included in the ontology.

example) data, software, and other types of equipment, may or may not require scheduling, depending on whether or not their availability is limited.

WORK ITEM (or ACTIVITY) instances are assigned to WORKFLOW PARTICIPANTS in the form of WORKLISTS. It is the dynamic, distributed development and management of these WORKLISTS, for both individual and team WORKFLOW PARTICIPANTS, for which the scheduling components in a multi-agent workflow management hierarchy are responsible.

Multi-Agent Environments

In our view, scheduling in workflow domains is inherently a distributed problem. In the component structure of a typical business environment or military command, responsibility for different parts of the problem at different stages of the process is naturally distributed among multiple agents. Schedules (or WORKLISTS) for the various teams and individual workflow participants involved are produced by different agents along different lines of decomposition. Each agent is responsible for producing schedules through either local scheduling of its own resources or delegation of scheduling responsibility to lower-level agents in charge of specific resource sets (i.e., WORKFLOW PARTICIPANTS).

The hierarchical description of resources (and resource constraints) advocated by the OZONE modeling framework, and warranted by the team and individual workflow participants of the workflow domain, provides a natural basis for decomposing and structuring solutions to the overall workflow scheduling problem. It provides a basis for specifying schedules at different levels to support decision-making at different stages of the planning/scheduling process. It likewise provides a structure for decomposing and distributing problem-solving responsibility, where different agents are responsible for allocation/apportionment of specific sets of resources at a given level of detail.

Building from this basic problem-decomposition perspective, we can articulate a specific model for distributed, multi-level generation and management of workflow schedules. The model assumes a hierarchical organization of scheduling agents, with each agent having access to specific levels of the underlying hierarchical domain model (in effect, the "full" hierarchical model is distributed among the scheduling agents). Thus, there is heterogeneity in the portion and level of description of the overall problem accessible to each agent.

Given the scale of the overall problem and the use of abstractions of resource-allocation constraints as a basis for specifying problems and solutions at different levels, two further decomposition assumptions follow directly:

- **Decision-making scope and granularity** - The portion of the overall problem that is visible and of concern to the decision-maker, and correspondingly, the level of detail of supporting models, can be seen in relation to particular stages of the overall process. For example, high-level resource-allocation decisions require a global (and necessarily coarse) view of the whole problem. Management of day-to-day activities involving teams and personnel, alternatively, requires much more detailed models of temporal process constraints and resource constraints, but only with respect to activities involving those resources.
- **Decision-making horizon** - Corresponding to decreasing scope and increasing model detail is a decrease in the temporal horizon of decision-making. This assumption is supported by two considerations: problem scale and presence of environmental uncertainty. The problem solver's computational burden can remain almost invariant at each level by balancing decreasing scope and increasing model detail. The extent of uncertainty in the operating environment makes the executability of more detailed models more suspect further into the future. Thus a given decision-maker's horizon must balance the computational burden of maintaining the solution over time (or equivalently the extent to which it really provides a useful projection of future events).

These collective assumptions lead to a distributed model that resembles the organization and roles of current workflow scheduling structures. Within this model there are two basic types of agent interaction:

- **Vertical**: The results of a given agent's (re)scheduling actions are communicated downward as scheduling constraints/objectives; an agent's ability to satisfy imposed constraints/objectives, or responses to lower-level results, are communicated upward. At each level of abstraction, an agent produces the best solution it can, given currently imposed global constraints and objectives, and the currently known results communicated from lower-level agent results (or the execution environment).
- **Lateral**: Agents at the same level communicate to resolve local conflicts and produce solutions within the bounds of constraints that have been imposed through downward constraint communication.

Given these assumptions, coordination of the entire agent organization can then be achieved through the development of the necessary "interaction policies" (or protocols) to govern agent behavior.

Issues and Challenges in the Development of Workflow Scheduling Capabilities

In the preceding section we have established a correspondence between elements of the workflow management domain and the concepts, characteristics and constraints defined in the OZONE scheduling ontology. The OZONE ontology reflects our prior experiences in constructing domain models (and schedulers) for a range of other planning and scheduling applications. The fact that most aspects of the workflow scheduling problem can be expressed naturally within this ontology indicates the level of commonality between this and previously addressed problems, and hence suggests the applicability of the scheduling techniques and tools that were developed in prior application contexts. In fact, the workflow management problem is not that different from other complex, continuous, distributed planning and scheduling environments.

At the same time, our analysis of the workflow scheduling problem brings to the forefront some specific modeling issues that have either not been present in previously addressed domains or have not been important enough solution drivers to warrant substantial attention. In expanding the basic OZONE ontology to address these issues, we have identified a set of extensions to core constraint-management and scheduling techniques that are required to treat all important aspects of the workflow scheduling problem and develop an effective workflow scheduler. In addition to these functional extensions, there is also a set of more general design decisions that need to be addressed, relating to the type and level of scheduling that makes sense in this domain, and the distribution of this functionality among multiple workflow agents.

In the paragraphs below we summarize the issues and challenges that we see in providing effective scheduling support for managing the WFM process. We work our way from basic design decisions to more specific functional capabilities.

- Level and Type of Scheduling Decisions to be Managed - There is a range of approaches that can be taken with respect to representation, interpretation and management of workflow schedules. In the simplest workflow scheduling scenario, the scheduling problem might be seen simply as one of maintaining ordered worklists for each resource. In this case, scheduling decisions are concerned strictly with resource selection and activity sequencing (e.g., prioritization). In contrast, the scheduling models used in prior applications of OZONE-based schedulers have additionally tied activities (and resource assignments) to specific time intervals on the timeline, dictating absolute start and end times in addition to sequencing constraints. Though it perhaps makes sense to start with a simple "worklist" (or sequence-based) schedule representation, there are some obvious limitations of this approach with respect to managing deadlines and making most efficient use of scarce resources. There is no explicit mechanism for measuring progress, for anticipating in advance that deadlines will be missed and for accurately assessing the magnitude of assigned workloads as other unassigned activities are considered. On the other hand, there is also some question as to whether the principal resources of concern in the WFM domain (i.e., individuals working independently and teams of individuals working collectively) really present that complex of an allocation problem; perhaps the generation and management of detailed time lines of scheduled activities adds greater complexity than it does advantage.

- Hierarchical Resource Models - Effective delegation and distribution of workflow tasks by higher-level agents requires "capacity" models of the agents and resources to which tasks are being delegated and assigned (respectively). In some cases, the mapping of atomic resources to aggregate structures is straightforward. For example, a team with four members will have the capacity to simultaneously handle four independent tasks. In other cases, however, the mapping can be more complex. What is the capacity of a multi-functional team? We might assume that team members work jointly on each task that is assigned and hence have a "capacity" of one. On the other hand, capacity will also be a function of the temporal granularity that is assumed in the schedule. If the temporal granularity of the schedule is one hour, then it is quite possible that this same team may be able to accomplish more than one activity in a given time tick (and hence have a capacity greater than one). The same capacity-modeling question arises with respect to the modeling of individual workflow participants; can they do more than one activity at a time? Or are they better modelled as continually context switching (i.e., preempting one activity in favor of another and vice versa)? The question is really one of what level of temporal granularity and detail is appropriate. It could also be the case that agents at higher levels in the organization operate at a coarser temporal granularity than do lower-level agents, in which case resources at the lowest level that behave as unit-capacity resources may in fact appear as having capacity greater than one at higher decision-making levels.

- Scheduling Horizon - Another, somewhat inter-related design issue concerns determination of the scheduling horizon (or set of scheduling horizons if it is assumed that certain global workflow agents plan and schedule increasingly further out into the future). One aspect of the scheduling horizon decision relates to the level of emphasis to be placed on predictive scheduling versus dynamic scheduling (when needed at execution time). For example, suppose a workflow agent has, as one of its resources, a team consisting of M individual members. A schedule of N delegated activities ($N > M$) can be generated in advance over some horizon, complete with individual team-member assignments and preferred execution intervals. At execution time, this schedule is simply followed, signalling the need for reactive change whenever a given constraint in the schedule cannot be respected (e.g., if a team member becomes delayed in executing a given task and falls outside of its execution interval). On the other hand, if activity durations are uncertain with high variance, then it perhaps makes more sense to defer specific resource assignments to execution time and allocate instead relative to team capacity models. Similarly, if workflow processes themselves are rather dynamic and uncertain, then the likelihood that the schedule will remain "executable" for very long might be questionable and it perhaps does not make much sense to schedule too far into the future. Our current sense is that the workflow processes are in fact fairly dynamic (with a good deal of inherent conditionality), and that consequently detailed schedules should not be expanded over very long horizons.

- Distributed Workflow Scheduling Framework - In the previous section, we outlined a multi-level framework for distributed scheduling that appears consistent with the distributed nature of typical workflow management environments. At the same time, many issues relating to its implementation remain unclear. For example, in order to maintain correspondence between delegated and assigned tasks belonging to agents at adjacent levels in the organization, it seems unlikely that an agent's scheduling capability can be treated simply as a "subroutine." As assigned tasks are scheduled at lower levels, and decisions (and execution results) are propagated upward, this information must be assimilated into the schedules of higher level agents; this is what provides a basis for recognizing problems and opportunities in the current schedule, and serves to trigger rescheduling processes. Another basic assumption, from the perspective of a given "scheduling" agent, is the need to support a mixed-initiative process, and hence interactive schedule visualization and manipulation capabilities are clearly important. However, scheduling of workflow is intimately intertwined with construction and instantiation of "processes" (or activity networks) and somehow these interfaces must be accessible from scheduling interfaces and vice versa.

- Generating and Managing (Sub)Demands - Related to the above integration issues are the functional capabilities needed to support a distributed scheduling capability. As the ACTIVITIES associated with producing a given PRODUCT are expanded and instantiated, an agent may choose to either carry out a given ACTIVITY locally or delegate it to another agent. In this latter case, a DEMAND whose objective (PRODUCT) corresponds directly to the final outcome (PRODUCT) of the ACTIVITY in question is generated and sent to the other agent. Thus, a given "leaf" ACTIVITY of the delegating agent is mapped to the "root" ACTIVITY of the agent to which the (sub)DEMAND is assigned; and this correspondence can be used as a means of communicating constraints and execution results between agents. Realization of this capability requires support for asynchronous task assignment (to other agents) and the development of task-delegation procedures.
- Limited-Lifetime Discrete-State Resources - One fairly common type of resource in the WFM domain is a so-called "limited-lifetime" discrete-state resource (referred to earlier also as an ARTIFACT). The existence of an artifact serves as a prerequisite (or enabling condition) to various activities, but it also may have a utility that decays over time (i.e., it will become outdated), like (for example) a weather report. Just as capacitated resources have *capabilities* for managing and querying their available capacity, there is the need for analogous capabilities for representing, maintaining and querying the existence and decaying utility over time of these types of discrete-state resources. For example, while some artifacts are probably updated periodically by default, the decision to reassess high-level objectives is more dependent on the current situation, and must be made as the situation warrants.
- Cyclic processes - A final modeling (and scheduling) issue concerns the treatment of cyclic processes. Many WFM processes involve repetitive processes with conditional exit structures. Within the current OZONE scheduling infrastructure, there is no capability to model conditional outcomes of activities and hence no capability to represent cyclic processes. We have, however, developed conditional plan representations within previously developed manufacturing scheduling systems (Smith 1989), and this provides a logical starting point for extending the OZONE ontology and modeling framework in this direction. The second important question regarding cyclic processes is of course how to handle their scheduling within a deterministic scheduling framework. Perhaps the most straightforward approach is to simply schedule only those activities preceding the next conditional branch point, and upon receiving the results of executing this conditional activity, resume scheduling of subsequent activities along the appropriate branch (up to the next conditional branch point). Our current feeling is that this might be a sufficient approach in the workflow domain. A second approach, which we utilized in our previous manufacturing scheduling work, is to use the probabilities associated with alternative outcomes (with appropriate discounting if the probability of repeating a process decreases each time through the loop) to *probabilistically instantiate* activity networks from "process definitions" (i.e., in effect, hypothesizing which branches will be taken and how many iterations will occur). These instantiated activity networks are scheduled deterministically, and then subsequently revised whenever actual outcomes differ from "expected" results. This more sophisticated approach could be more appropriate if it is important to maintain a reasonably accurate picture of available resource capacity.

Final Remarks

In this paper, we have characterized workflow management as a scheduling problem. The mapping of the workflow domain ontology onto the OZONE ontology suggests that the workflow management problem is not significantly different in structure from other types of scheduling problems. In addition, the distributed nature of the WFM problem makes it amenable to a multi-agent solution approach. Nevertheless, the WFM problem brings to the forefront several important modeling and design issues that still need to be addressed by workflow scheduling solutions.

References

Smith, S., and Becker, M. 1997. An ontology for constructing scheduling systems. In *Proceedings, AAAI-97 Spring Symposium on Ontological Engineering*, 120–129.

Smith, S.; Lassila, O.; and Becker, M. 1996. Configurable, mixed-initiative systems for planning and scheduling. In Tate, A., ed., *Advanced Planning Technology*. Menlo Park CA: AAAI Press. 235–241.

Smith, S. 1989. The OPIS framework for modeling manufacturing systems. Technical Report CMU-RI-TR-89-30, The Robotics Institute, Carnegie Mellon University.

Smith, S. 1994. OPIS: A methodology and architecture for reactive scheduling. In Zweben, M., and Fox, M. S., eds., *Intelligent Scheduling*. San Francisco CA: Morgan Kaufmann. chapter 2, 29–66.