Component Based Task Models for Agent Coordination

Brian Drabble
Computational Intelligence Research Laboratory,
1269, University of Oregon,
Eugene, OR 97403
drabble@cirl.uoregon.edu

Abstract

This paper describes the initial development of an intelligent tasking model which has been designed to enable complex systems, human agents and software agents, to be tasked and controlled within a reactive workflow management paradigm. The task models exploits recent advances within the AI community in reactive control, scheduling and continuous execution. The Dynamic Execution Order Scheduler (DEOS extends the current workflow paradigm to allow tasking in dynamic and uncertain environments by viewing the planning and scheduling tasks as being integrated and evolving entities. DEOS is being applied to the domains of Air Campaign Planning (ACP) and Intelligence, Surveillance and Reconnaissance (ISR) management. These are highly reactive domains in which new tasks and priorities are identified continuously and plans and schedules are generated and updated within a temporal and resource constrained setting.

Introduction

DARPA has identified workflow as one of its key "must have" technologies and is prepared to invest heavily in developing the next-generation workflow systems for the military. However, in order to do this there needs to be a quantum leap in the current capabilities of workflow engines. To date workflow systems have been applied to fairly static environments in which the execution of activities follows a fairly predictable path, e.g. mortgage application processing. This is not the case in the domains of interest to DARPA e.g. logistics, crisis management, mission planning, where new events give rise to changes both in the task and the status of the domain. This requires technologies which are capable of reacting quickly to changes e.g. enacting new processes, editing and deleting existing ones and rebalancing the current resource assignments, i.e. new agents are added and removed, new capabilities evolve.

Similar needs are now starting to arise in the business area with an increasingly competitive marketplace, along with widespread automation and the availability of online information. This has sparked interest in the re-engineering and automation of business processes. The field of workflow management (WFM) has emerged as an outgrowth of this interest in recent years. The workflow community advocates the use of explicit models and representations of processes, along with automated tools to support the activation and ongoing management of workflow processes. In a similar development, automated systems are themselves becoming more complex and are required to interact in more flexible and intelligent ways. Software systems of the future may be viewed as intelligent workflow enabled products supporting the highly responsive businesses of the future. Many domains of interest to the workflow community are characterised by ever-changing requirements and dynamic environments. However, traditional workflow systems provide only limited reactivity and flexibility. Within the AI community, work on reactive control has led to the exploration of techniques for intelligent process management to meet the requirements of adaptivity for dynamic and unpredictable environments.

This paper describes a revolutionary approach to workflow management using advanced AI planning, scheduling, and reactive control techniques. The system described is currently being developed for a number of different DARPA needs, e.g. ISR management, air mission planning and crisis management. The paper is structured as follows, Firstly, it describes the motivation behind the need to develop reactive models and gives a brief overview of the state of workflow systems. Secondly, it describes how rich models of activities and tasks are essential in building reactive workflow systems. Thirdly, it provides an overview of the ACP domain and fourthly, describes the component task models developed for reactive workflow. Fifthly, it describes the information model which has been developed to support the task models and sixthly, describes the DEOS scheduling engine developed to manipulate the task and information models. Finally, it provides a summary of current progress and describes a mapping from the military domains used as examples to a manufacturing one.

Motivation

Through a number of different programs DARPA is exploring the benefits and use of workflow support. There are two mains reasons why DARPA's is looking at workflow support:

- shrinking defense budgets means that there fewer assets e.g. planes, men, tanks, etc available while the demands for more efficient and more timely planning continue to increase.
- the nature of military planning is changing from large scale global conflicts to more localised ones. These local conflicts require fast responses e.g. from an air campaign perspectives missions need to be flown within 24 hours of the situation arising and must be planned with less than 90

Current DARPA research in workflow is being undertaken in two main areas: air campaign planning and ISR management. Research to support air campaign planners was funded through the ARPI with the development of the ACP planning tool (ACPT) [Hoffman *et al.* 1996]. The function of the ACPT was to take a number of high level objectives and break these down to a series of sub-objectives, tasks and finally targets to be attacked. This requires the coordination of multiple assets, e.g. fighters, bombers, tankers, SAED, etc, to meet these high level objectives, e.g. gain air superiority, isolate command and control centers, etc. Research to support ISR planners is being funded by Advanced ISR Management (AIM) program, which is supporting intelligence planners with tools for more effective and efficient management of the assets and information in the system. These tools address the complete ISR management problem from the strategic development of ISR objectives to the management of individual ISR assets. All of these tools and the humans in the ISR domain are viewed as agents that are capable of adding value to the emerging ISR plan.

For these highly interconnected domains to work efficiently, they must be organised and managed; that is, a workflow plan that describes how the problem will be solved and identifies the agents necessary to support it must be developed. This will involve the development of intelligent workflow techniques that identify when an agent should be called and which tasks are appropriate for the agent to solve. Without intelligent workflow management, the integration of information discovery, acquisition, exploitation, and dissemination would be impossible. This situation parallels many business and manufacturing situations where the timely acquisition and delivery of information, within time and resource constraints, is vital.

Current Workflow Systems

Workflow is a fast-evolving area that has evolved primarily from the desire to understand, organize, and (often) automate the processes upon which a business is based. These roots are reflected in the following definition of workflow management from the Workflow Management Coalition (WFMC), an international organization focused on the advancement of workflow management technology and its use in industry, *http://www.aiim.org/WfMC/*:

> The automation of a business process, in whole or part, during which documents, information or tasks are passed from one participant to another for action, according to a set of procedural rules. [Hollingsworth 1994]

The requirements and capabilities inherent to workflow apply to many process coordination and control problems. In this paper, we use the term *workflow* to refer to this more general notion of process management.

The role of a *workflow management system* is to provide the services required for automating workflow processes. The WfMC presents a generic WFM system in its Reference Model [Hollingsworth 1994]. WFM systems can be characterised by the following functional components:

- Modeling and representation of workflow processes and their constituent activities
- Selection and instantiation of processes for activation in response to a user request or key events
- Scheduling of activities to agents and resulting tasking of the agents
- Monitoring and adaptation of executing processes

At the heart of a WFM system lies a library of templates that encode explicit *process models* for the problem domain. Templates can be represented as explicit sets of tasks to be undertaken, or more abstractly as collections of constraints on allowed activities. Activities can encompass a broad array of operations, including transmission of data, tasking of software agents, or communication with human operators. Process modeling and representation are typically limited to build time within current WFM technology. However, process creation and adaptation will inevitably migrate from build time to runtime as application domains demand flexible adaptation to environment dynamics.

Templates are selected for *activation* based on current tasking and environmental conditions. Selected templates can be tailored to a range of situations through appropriate instantiation of template variables. Typically, activation results in the addition of the constituent activities of the instantiated template to an *activity list*, which contains information about all current and pending tasks. As part of activation, resource allocation is performed for new activities based on current and projected resource availability.

The term *enactment* is used generally to refer to the execution and ongoing management of activated processes. For many domains, process execution will occur within highly unpredictable and dynamic operating environments. For this reason, enacted processes should evolve and adapt over time in response to changes in the environment, the addition of new tasks, partial execution results, and other factors. Monitoring plays a critical role during enactment, to detect key events that may necessitate adaptations to current processes, or enactment of new or different processes. Similarly, process templates should evolve over time, to reflect improved models of the domain obtained by analyzing information gathered from previous enactment episodes.

Why Rich Models Are Important

In the ACP domain, as in many complex domains, the underlying process controlling the flow of information from user requirement to satisfaction defaults to a basic stovepipe. Requirements are pushed in one end, and after a fairly linear path, pushed out the other end. This encourages batch processing and creates an inflexible and unresponsive system. For the workflow manager to have a significant impact on the dynamic responsiveness of the system it must be able to manipulate and create context-dependent processes, thus, the requirement for rich models.

> Effective workflow management requires representations that makes the process logic explicit, thus allowing processes to be readily understood and adapted. [Myers & Berry 1999]

AI provides techniques to enhance SWIM's capability to both select processes based on context and adapt processes to the dynamics of the environment. For example, hierarchical models are desirable for modeling sophisticated processes because of their capacity to simplify complex tasks. Basic constructs that should be recorded for a process include the purpose, expected effects, applicability conditions, resource requirements, scheduling constraints, participants, and subprocesses that may be invoked. Process representations should support the definition of metrics relating to the time, cost, or quality of performing a process, thus allowing comparisons and improvements to be made. Concepts such as authority and accountability are also useful and essential in the AIM domain.

Another characteristic of a good process representation is the ability to support a rich set of control metaphors, including iteration, sequencing, concurrency, monitoring, testing, and suspension/resumption. Given the unpredictability of the operating environments, the ability to represent uncertainty is critical. As discussed in the workflow literature [Cichocki *et al.* 1998; Lawrance 1997], the models must also be rich in low-level constructs allowing transactional capabilities, synchronization, and information exchange.

Overview of the ACP Process

The ACP process defines the mechanism for translating high level objectives e.g. "gain and maintain air superiority by D+2", "destroy weapons of mass destruction by D-5", etc set by the commander into actual targets and missions to be flown. This involves breaking down the objectives into sub-objectives and further refining these sub-objectives to tasks and missions. At each step a large number of agents (human and/or software) are involved to gather, refine and communicate information regarding the needs of the process. Within the current ACP process the main focus is the development of the Air Tasking Order (ATO) which describes which targets will be attacked, when and how. The USAF themselves have identified a number of problems and shortcomings in the ATO generation process and include:

- the need to move away from the "stove piped" way in which military plans are generated. For example, the generation of an ATO takes many hours with a new ATO issued every 12 hours.
- inability to rapidly responding to changes and requirements. The fixed nature of the ATO does not allow units to be retasked on the fly because there is little dependency and plan rationale recorded.
- lack of integration between different plans and forces into a fully integrated battlespace. Separate plans for ISR, tankering, ammunition, etc, mean that opportunities and problems can be missed leading to inefficient and potentially costly ATOs
- to make better use of the shrinking number of resources they have at their disposal. For example, by better allocating resources in means less tasks need to be redone and the quality of the final product increases.

Although the ATO planning process involves activities and their coordination, they are described in terms of the activities that take place in the planning process itself (such as plan, change, review, publish) rather than containing activities that relate to military effects (such as destroy, paralyse, delay, etc). Thus the ACP is a planning process whose outcome is a plan, i.e. the ATO. The ATO provides the low level detail which the pilots and mission planners need to plan at the domain level. At the domain levels the planning involves allocating a number of resources, i.e. aircraft and weapons to specified targets at a time designated in the ATO. The problem is complicated by the ability of the aircraft to be reconfigured to suit the mission. For example, a flight of 4, A-10 thunderbolts carrying AIM-65 Maverick anti-tank missiles would be a perfect match for missions against armoured formations. However, the same flight of A-10s could be re-armed with MK-82 bombs and sent on a different mission. The problem is further complicated by the fact that weapons have a probabilities of both hitting a target and destroying it. The planning process aims is to find an optimal assignment of aircraft to targets so as to avoid having to restrike the target or causing uneccessary risk of aircraft loss or collateral damage. There are a number of different ways in which the optimality can be measured, e.g. number of missions undertaken, number of targets destroyed, aircraft lost, etc.

In order to overcome many of the problems involved in generating large ATOs and then repairing them when changes occurred the USAF and DARPA are looking for more reactive models of coordination and workflow between the process and domain levels. The areas being explored with this approach include air campaign planning, logistics management and intelligence, surveillance and reconnaissance management. The aim is to turn both the process and the domain levels into workflow processes in which targets are assigned to aircraft on an as needed basis and the results and outcomes of the mission are fed straight back into the process levels. If the target has been successfully dealt with then it is removed from the list. If not then process level activities are initiated to deal with the outcome, e.g. restrike immediately, integrate with a later mission, rerouting air to air refuelling tankers and SEAD[1] aircraft, retask an inflight mission, etc. DARPA refer to this as "just in time targetting" and has alot of similarities with tasks found in manufacturing domains. This will be discussed in Section .

The original task models were developed for the domain level (to improve coordination and feedback to the process level) but is was soon discovered that these models could be generalised to support planning and task management at the process levels as well. Details of both of these models are now provided.

Task Models

Tasks models define a natural breakdown of a task into its constitiuent parts. Rather than have the task as a single component it is held as a series of subcomponents which reflects the generic sub-activities neccessary to support the task. This allows the tasking agent to create a better model of the processing needed and allows a scheduling algorithm to better understand how to allocate resources, identify tradeoffs, asses changes and modify the workflow. To date two task models have been identified and applied to both the ACP domain and the ISR domain.

[1] SEAD aircraft protect other aircraft from SAM attack

Domain Domain Task Model

A generic domain task is broken down into 5 sub-tasks and is referred to as the PRFER task model.:

- Plan: Time taken for the pilot to plan the mission. Once a plan has been identified it is inserted in the slot for other workflow tasks to examine and check.
- Ready: Time taken to prepare the plane for the mission
- Fly: Time taken to get to the mission objective [2]
- Execute: Time taken to execute the mission, e.g. drop weapons, unload food pallets, etc.
- Reconstitute: Time taken to turn the aircraft round once it has returned to base.

Each task is associated with a task specification blocks (TSB) which is allowed to "breath" as changes in the domain are reflected as changes in one or more of the TSB's sub-blocks. For example, if the aircraft chosen for the mission develops a failure during its ready time then that sub-task will expand and accommodate the extra time. Alternatively the workflow engine may decide to substitute the aircraft for another if a spare aircraft exists or another can be re-weaponed in time. If no other aircraft is available then the workflow engine may try and reduce the time of the execute block to recover the lost time. For example, if the aircraft is tasked with a food drop and the current method is to land and off load the supplies then the execute block would be three hours. If it was changed to an air drop then the execute block would drop to 30 minutes but the ready time would increase due to the time take to change the food pallets to an air drop configuration. By breaking the task into sub-components it allows the workflow engine to focus on ways to improve the schedule and recover from changes occuring in the domain and task. New TSBs can be added to the schedule as needed and removed just as easily should the decision be reversed. For example, if the drop is being carried out using C-141 Starlifter aircraft then it must be loaded using a K-1 lifter but if C-5 Galaxy aircraft are used then a K-1 is not needed. Once an aircraft is chosen then the workflow engine can examine the ready block and examine needs for the aircraft type, in this case a K-1 lifter. Other needs such as fuel can be attached to other sub-tasks e.g. fly. Full details of the DEOS workflow engine which supports these tasks models is provided in Section .

[2] This can be replaced by a "drive" or "sail" block for operations using land or sea transport

Process Task Models

The generic process task is broken down into 4 sub-tasks and is referred to as the PAER model:

- Plan: Time to plan the task. Once a plan has been identified it is inserted in the slot for other DEO tasks to examine and check.
- Acquire: Time to acquire the information e.g. process products [3] necessary to carry out the task. This also specifies the resources e.g. platforms, software packages, etc, needed to run the task.
- Execute: Time to carry out the alloted task.
- Report: Time to file or report the results of the task.

Again each task is associated with a TSB and can be handled with the same workflow engine as the PRFER model. During the planning sub-task a number of requirements are identified and posted to the acquire block. The workflow engine can generate new TSBs for these if there are none in the current system. Alternatively, if another TSB is expected to generate the required document then its report block can be modified by the workflow engine to provide an addition copy. In this way the execution of the PAER block is a partial order with some information gathering being carried out before all planning is complete. In addition, by identifying the information passing between process level TSBs and domain level TSBs it becomes possible to route the right information through the hierarchy.

Information passing through the system is used to coordinate and trigger different TSBs. For example, once a target list moves from recommended to approved (through a vetting task) then other TSBs are triggered. However, the workflow engine could identify that a task can execute with the recommended target list to get a "jump start" and can finish its processing once a check has been made against the approved target list (in case changes have been made). This allows the workflow engine to trade off accuracy for time i.e. the recommended target list may change but if not the process can be considerably shortened. It is this type of adaptive workflow that DARPA is looking for.

Examples of Task Model Use

The PRFER and PAER models can be used to handle a number of different situations occurring in the workflow process.

- **Change in a major process product e.g. commanders guidance**:
  This triggers an event in the workflow manager that a primary process product has changed status unexpectedly and that action should be taken. This means identifying the TSBs in the process which have the commander's guidance in their "Acquire" block. This would result in some tasks being suspended and others re-tasked to make use of the new information. If a secondary process product supporting the development of a new commanders guidance is delayed then the "Acquire" block of any task using it would be extended by the time required. This will have potential "knock on" effects with other tasks in the process. Should the support document be delayed indefinitely an alternative or inferior (the previous days) would be used instead.

- **Initial loss of a domain asset, e.g. U2 aircraft**:
  As with the previous example the assumption is that the change requires one or more current/planned tasks to be modified. This would be achieved by identifying the tasks impacted by the lost asset. The workflow engine would identify the process products associated with the lost asset. These could then be updated and changed as appropriate (see above). An alternative view would be to change the type of processing the tool is carrying out. For example, if the tool was carrying out a full schedule then the option might be to reduce that to a feasibility estimate to try and determine the consequence of the lost asset. This means reducing the "execute" block to reflect a feasibility probe rather than the full schedule.

- **Loss of a process level asset, e.g. loss of asset scheduler:**
  The loss of a process level asset allows a number of interesting options to be explored. The options include adding news tasks and in some cases decomposing tasks to lower levels should there be no equivalent asset available. This can be handled simply by adding the failed task to the agenda and modifying its process product needs. For example, if some process products have been created or updated then the new task can ignore this need. If however, the task is yet to start then a simple substitution may be called for, it depends on when the asset goes down in the process schedule.

- **Upgrade in the forces in the conflict and the need for feasibility estimates of the ISR needs**:
  The above examples show workflow engine repairing events occuring in an already assigned series of tasks. There will be situations in which new tasks and requirements are added to the system. Such a situation would arise if a feasibility estimate is needed for ISR requirements while trying to maintain the overall picture of the ACP process. This would mean identifying ways of scaling back current effort through changing "Plan" and "Execute" blocks of needed TSBs.

[3] Process products are the orders, documents, reports, letter, etc which are used to coordinate the workflow process

Information Model

Each of the tasks at either the process or domain level is specified using "task verbs" indicating the task to be performed. The tasks are described in the following form:

- **Verb**: the task to be carried out (e.g., analyze, develop, refine)
- **Noun Phrase(s)**: one or more noun phrases describing the object(s) or products on which the activity is being performed (e.g., prioritised target list)
- **Qualifier(s)**: zero, one or more qualifiers constraining how the activity is performed (e.g., time/resource limits)

The process products are modeled as resources that are created, modified, used, and authorised within the process. Examples of process products are the documents, reports, orders, letters, and communications (formal or informal). Authority relationships and other conditions are also modeled and can be used as an extension of the basic mechanism. Current task models and process products are encoded in the ACT representation [Myers 1993], which can be directly executed by the Procedural Reasoning System (PRS) [Myers 1996; Georgeff & Ingrand 1989]. It is hierarchical and provides a rich scheme for both the representation of normative processes and the derivation of new processes based on AI reasoning and planning.

Details of the development of the verb and process product models can be found in [Berry & Drabble 1999]. This framework is general enough to be applicable across various different military domains as well as those found in manufacturing, logistics and supply chain management. The *verb/noun(s)/qualifier(s)* (VNQ) model was originally developed for the ACP domain [Drabble, Lydiard, & Tate 1998]. Full details of the verbs and process products are given in the following sections.

Verb Descriptions

A full list of the verbs and there definitions is given in [Drabble & Lydiard 1996] and a portion of the hierarchy is shown here which describes the decomposition of the major verbs **Analyse** and **Decide**.

- **Major Verb: Analyse**
  - **Notes**: **analyse** has notion of quantitative investigation
  - **Sub-Verbs**: predict, determine, monitor, diagnose, establish, measure.
- **Major Verb: Decide**
  - **Notes**: some of the meanings of **decide** imply finality (finalise, terminate), others seem to imply select by consideration (approve, calculate).
  - **Sub-Verb**: complete, finalise, approve, terminate, choose.

The verb model can also be use to describe the capabilities of the agents carrying out the tasks. By having the capabilities and the tasks described in the same language it makes the "match making" function a great deal easier. For example a planner could be categorised using the verbs "refine", "produce" and "modify". The capability descriptor would also define the process products and objects it required together with other resource information.

Process Objects and Products Model

A series of matrices is produced which defined for each task in the process, the process products which are created, read, updated and approved (referred to as the CRUA matrices[4]. This allowed for the identification of the process products which could be associated with a given verb and/or qualifier. For example, the verb "review" can be applied to the JFACC Guidance Letter but the verb "critique" could not. This analysis also identified potential classes of process products and associated values which could provide additional structure to the task models For example, all process products which refer to list of targets, e.g. candidate target list, target nominations list, service target nominations, JIPTL and JIPTL cut-off, etc. The CRUA matrices were also used to identify the other constraints which are involved in each step in the ACP process. The constraints are either:

- **resource**: these are the planning cell personnel required to carry out the activity.
- **temporal**: these are either qualitative, e.g. "the start of activity A precedes the start of activity B" or quantitative, e.g. "activity A must end no later than 16 hours after the start of conflict".
- **authorities**: these describe specific authorities which must be obtained before an activity can start or finish, e.g. presidential authority must be given before the operation can begin.

An example of part of the task to process product association is given in Table 1. However, some tasks refer to objects in the domain but without associating it with a specific process product. For example steps such as "group targets" and "deconflict airspace" refer to objects in the domain rather than process products. Features of the ACP domain such as airspace, targets, ground features, etc, exist in the real world and are not created by the ACP process. It was possible to alter the steps description to explicitly identify the possible process product(s) involved. For example, it was possible to describe. "deconflict airspace" as "deconflict

[4] From a suggestion by David Hess of SAIC.

Verb	Noun Phrase(s)	Qualifier Phrase
Deconflict	ACM Requests	
Finalise	Air Control Order Special Instructions Air Tasking Order	Quality Control
Produce	Air Tasking Order Target Groupings CAS Sortie Allocations Potential Target List Initial Target Nomination List Weaponeering Assessment Weaponeering Force Assessment Mission Support Requirements	Broad
Release	Air Tasking Order	
Consider	Target and Route Threats	

Table 1: Part of the Verb/Noun(s)/Qualifier(s) Table

airspace management requests" and "group targets" as "group targets of the candidate target list". However, in the case of "deconflict airspace management requests" it was not certain that this truly reflected the way in which the step was carried out. For example, it is possible to deconflict the airspace by using a 3D model and tracing the entry and egress routes of the aircraft on the model. Thus it is possible to deconflict the airspace without making reference to the airspace management requests. For this reason it was decided to introduce a new class of objects to the model referred to as process objects. The distinction between process objects and process products is that process objects are not created by the ACP process but can be used, modified and consumed in the same way as process products.

Using the process product features identified from the CRUA matrices it is possible to group features into classes and associate with each class a descriptor type. For example, the features "available" and "not available" could be grouped together to form a single feature "availability" which can take one of these values. These features were used to form an ontology of primitive process product features which could in turn be used as the building blocks for more complex reasoning about the status of process products. For example, the status of the document could be (available, compound, published, draft). These are not be part of the primitive ontology of process products but are instead composed of elements of the process products primitive ontology. In addition to identifying the features and values it is also necessary to identify the relationship between the potential values. This results in a series of relationships being defined for each features and the relationships identified are as follows:

- **Boolean**:
  The class can take one of two possible features. For example, a document can either be available or not available.
- **Scalar**:
  The class can take one feature from a scalar set. For example, a document's contents level can be either draft, CONPLAN or OPLAN.
- **Vector**:
  The class can take a number of fixed features. For example, a document's access information could be composed of a triple of modification operation, agent and date, e.g. (create, O-Plan, 25-Apr-97/22:10:00).
- **Set**:
  The class is a 0..N description composed of a number of sub-descriptors or classes. A document could have a status described in terms of, availability, set_of_contents, review_status, issue_status.

While this model has been developed from the ACP domain it is general enough to be applied to a wide range of workflow domains where system (human and/or software) are required to work together. The model involves a set of process object and product "features" which have a given type.

- **Process_Product_Type: Scalar**: Every process product has this description element. Process_Product_Type includes, ATO, ACO, JIPTL, etc.
- **Contents: Scalar**: Every process product has this feature.
- **Contents_Type: Scalar**: Every process product has this feature. It could use the MIME types as values for the field.
- **Description_of_Contents: Vector**: This describes the contents of the process product and would vary for different classes of process products.
- **Availability: Boolean**: This describes the availability of the process product and simply defines whether the process product exists or not. Examples of this class are as follows: not_available or available.

- **Type-Information:**
  Scalar: The type-information describes the type of information contained within the process product. The types defined for each class of process product must be MIME compliant.
- **Contents_level: Scalar**: The contents_level defines the different levels at which the contents of a process product can be "measured".
- **Review-Status: Scalar**: This describes the review status of the process product as it is reviewed and passed through the ACP process. Examples of review-status include: on-going, cut-off, final.
- **Approval/Recommendation-Status: Scalar**: This describes the status of a process product as it transitions from being recommended to approved. (A number of steps on the ACP process can start once a recommended JIPTL or TNL is available but must not complete until there "results" have been checked against an approved JIPTL or TNL.) Examples of approval and recommendation status include: recommended, approved.
- **Issue-Status: Scalar**: This describes the availability of the process product to other agents and systems. Examples of issue status include: Examples of issue-status include: current, unreleased, released, published, issued, proposed.

The items described above form the elements of an ontology of primitive process product feature descriptors. These features could be used in higher level reasoning to deduce new information about the process products. Details of some of the derived information is as follows:

- **Status: Set**: This describes the status of the process product as it "moves" through the planning process. The status values will be vary for different process products classes and examples of status values include: availability (boolean),description of contents (scalar), review status (scalar), issue status (scalar). For example, the status of the JIPTL) could be described as (status: available compound approved released). The status values for different classes of process product would have to be agreed between the agents in the process in order to avoid different values appearing in the slots of the status descriptor. All process products must have a status.
- **Access-Information: Set**: The access-information describes the types of access made to the process product and the agent and time at which it was carried out. Examples of access-information include: modification type e.g. create, modify, user (scalar), agent involved (scalar), date (scalar).

DEO Scheduler

Dynamic Execution Orders (DEO)s were identified and developed by Col. "Buster" McCrabb of the USAF's Force Development and Experimentation Group in response to the needs of military commanders to develop fully integrated and responsive Air Tasking Orders (ATOs) under tight resource and time constraints.

The basic concept behind a DEO is to generate schedules quickly and to update them on the fly as new requirements and changes occur in the domain. A DEO schedule uses an expressive formalism that breaks down the tasks into the constituent parts. These parts reflect a natural breakdown of the task from a user perspective.

Each task, or low-level process activity, is represented by a task specification block (TSB) composed of the PAER sub-tasks. The TSB can "breath" as changes in the domain are reflected as changes in one of the TSB's sub-blocks. For example, if an agent chosen for a task develops a failure during it's **A**cquire phase then the **A**cquire sub task will expand and accommodate the extra time. Alternatively, a second agent may be scheduled with the task inheriting the results from the initial **P**lan phase.

The more common reason for a TSB to change is due to a "knock on" effect from another TSB. For example, agents may be performing activities related by the workflow process. A change in the **E**xecute block of the first may push the **A**cquire block of the next ahead in time. By creating a dynamic link between sub-blocks it becomes possible to quickly identify the impact of a change and to identify an appropriate set of repairs. Failures may also cause new TSBs to be added to schedule to deal with schedule repairs. The dependency links usually reflect an interchange of information between the tasks.

A TSB can be generated in response to decision made by other parts of the schedule. For example, if an ISR task is scheduled to a U2 (aircraft) to take pictures then a film processor is needed to develop the negatives. If an alternative asset is used (i.e., collect information using infra red imaging) then perhaps a different film processor is required or none at all. The scheduler dynamically launches a new TSB to deal with the film processing task and provides feedback to the user should the availability of a processor be a problem. This approach also has the advantage of allowing the scheduling problem to be broken down into a number of different perspectives. For example, the staff operating the photo development laboratory do not need to know the details of the aircraft which is flying the mission, just the time at which the cameras should be available. By using the DEO model to break the scheduling problem into smaller pieces it allows large scheduling problems to become tractable and maintains the necessary dependencies between the subproblems.

Note that the PAER model is a generalization of the original task breakdown designed for the air campaign planning domain, the PRFER (Plan, Ready, Fly, Execute, Reconstitute) model. However, the PAER model is a great deal more generic. The PAER model is applicable to a wide variety of domains including those that involve human agents. One difference from the PRFER model is that once planning is complete the "plan" is posted in the plan sub-block so that it is available for inspection by other TSBs. In addition, where information needs have been identified during planning which need to be acquired a new TSB will be created. Thus, information acquisition and planning can proceed in parallel.

An example of the PAER model has been studied in the area of mortgage application processing with a large Scottish Bank. The existing problem was associating information arriving in support of an application to the original case (e.g., a credit report to the letter from the applicants employer). By explicitly adding information from the **P**lan, **A**cquire, and **R**eport phases to subsequent sub-tasks it becomes possible to create the necessary dependency chains. Further investigations in this domain are now being planned to examine resource utilisation, information throughput and its implications on the Bank's development of call center operations.

Squeaky Wheel Optimization (SWO)

SWO is a scheduling technology developed for application to real-world scheduling applications [Joslin & Clements 1998]. The insight behind SWO is that in any real-world problem it is impossible to capture all associated constraints and that in most cases the context in which the constraints apply cannot be easily determined. SWO uses a priority queue to determine the order in which tasks should be released to a greedy scheduling algorithm. The priority queue is determined by how difficult the task is to deal with that is, the higher the task is in the queue the harder it is to handle it correctly and not by some external priority identified by the user. On each iteration of the algorithm, SWO quickly creates a schedule and then examines it to identify the parts that were handled badly, for example, task was completed too late or by an unsatisfactory agent. Any task that "squeaks" is promoted up the priority queue, with the distance it is promoted determined by the the extent of the problem. The new priority queue is then used to generate another schedule that is analysed for problems. This process continues until no significant improvement in the schedule is noted over several iterations. SWO is extremely fast with each cycle of generate, analyze, and reprioritize taking less than a second, even for large problems.

SWO has the advantage of moving through the search space in a coherent manner, looking for better solutions. It also has the advantage of allowing changes in the environment and task to be easily integrated into an ongoing solution. For example, a new task can be added to the priority queue and dealt with on the next cycle.

Summary and Further Work

We have presented an overview of a system for dynamic task management. The task models are work in progress but already provide the foundation for workflow enabled reactive control. This includes an agent-based architecture, rich modeling and representation schemes for processes and their constituent actions, flexible integration of process instantiation, task allocation and execution, and highly reactive scheduling techniques. To date the task model has been applied to large scale ACP problems i.e. 2500 targets and 200 aircraft over a 10 day period and to process level task in the ISR domain. Further work with the Smart Workflow for ISR Manager (SWIM) will further validate the approach and techniques.

The PRFER and PAER models were originally developed for military applications. However, being generic models they are applicable across a wide number of domains including manufacturing, logistics and assembly. Work with Intel is intended to verify the models on a number of semi-conductor assembly problems and the Table 2 shows a mapping between the ACP domain and the semi-conductor manufacturing domain.

The reward for striking a target in the ACP is a number of points. This can be replaced by the retail price of the order. As with the ACP problem the price should be down graded depending on the lateness of the order. The evaluation function for the ACP problem is the number of targets attacked, the suitability of the weapon for the target and the time at which the target was attacked. In the semi-conductor domain the evaluation function would be to minimize the number of late orders. Other evaluation functions include minmising the inventory and/or maximising the work in progress. This simple comparison shows there is a great deal of scope for using the PRFER and PAER models for large scale manufacturing problems.

Acknowledgements

This ideas in this paper have been refined through continual discussions with the staff and students of CIRL. The works has also benefited from the insight and enthusiasm of Col Buster McCrabb (USAF) (Ret).

This research is supported by DARPA Contract: DABT63-98-C-0069 "Intelligent Workflow for Collection Management" and Contract: F30602-97-1-0294 "Understanding and Exploiting Hierarchy". The U.S. Government is authorised to reproduce and distribute reprints for Governmental purposes notwithstanding

DEO	Semi-conductor Manufacturing
Strike Missions	Orders (size of lot and product (e.g. 500 chip 4567)).
Popup Targets	New orders and changes to existing orders (e.g. I need 600 chip instead of 500).
Attack Window	Time from product insertion to delivery.
Aircraft	Machines on which the chips/PCBs are manufactured.
Weapons	Auxiliary resources, e.g. wafers, connectors, etc. There would need to be an inventory kept of components and their specifications.
Prob (hit) & (kill)	The expected profile of the chips made on a particular machine, e.g. the number of failures, wrong speed, etc. This will vary from machine to machine.
Re-weaponeering	set up times for the machines.
Air to Air Refuel	Delivery of raw materials to the factory.

Table 2: Military and Manufacturing Domain Comparison

any copyright annotation hereon. The views and conclusions contained herein are those of the author and should not be interpreted as necessarily representing official policies or endorsements, either express or implied, of DARPA, Rome Laboratory or the U.S. Government.

References

Berry, P. M., and Drabble, B. 1999. The AIM Process Modeling Methodology. Technical report, AI Center, SRI International, Menlo Park, CA.

Cichocki, A.; Helal, A. S.; Rusinkiewicz, M.; and Woelk, D. 1998. *Workflow and Process Automation: Concepts and Technology.* Kluwer.

Drabble, B., and Lydiard, T. J. 1996. Model of the ACP Process Verbs and ACP Tool Capabilities. ISAT Technical Report ISAT-AIAI/TR/1, AIAI, Division of Informatics, University of Edinburgh.

Drabble, B.; Lydiard, T. J.; and Tate, A. 1998. Workflow Support in the the Air Campaign Planning Process. In Myers, K., and Ferguson, G., eds., *Proceedings of the Workshop on Interactive and Collaborative Planningheld as part of Fourth International Conference on Artificial Intelligence Planning Systems (AIPS-98).*

Georgeff, M. P., and Ingrand, F. F. 1989. Decision-Making in an Embedded Reasoning System. In *Eleventh International Joint Conference on AI.*

Hoffman, M.; Griffith, A.; Shoop, J.; and Rumanes, L. 1996. Developing a Transition Path for ARPI Technology into the Air Campaign Planning Domain. In Tate, A., ed., *Advanced Planning Technology,* Technology Achievements of the ARPA/Rome Laboratory Planning Initiative.

Hollingsworth, D. 1994. The Workflow Reference Model. Technical Report TC00-1003, Workflow Management Coalition.

Joslin, D. E., and Clements, D. P. 1998. Squeaky-wheel Optimization. In *Fifteenth National Conference on Artificiall Intelligence.* Menlo Park, CA: AAAI Press.

Lawrance, P. 1997. *WfMC Workflow Handbook 1997.* Wiley.

Myers, K. L., and Berry, P. M. 1999. Workflow Management Systems: An AI Perspective. Technical report, AI Center, SRI International, Menlo Park, CA.

Myers, K. L. 1993. *The ACT Editor User's Guide.* AI Center, SRI International, Menlo Park, CA.

Myers, K. L. 1996. A Procedural Knowledge Approach to Task-Level Control. In *Third International Conference on AI Planning Systems.* AAAI Press.