Exploiting AI Technologies to Realise Adaptive Workflow Systems

Peter Jarvis[†], Jonathan Moore[‡], Jussi Stader[†], Ann Macintosh[†], Andrew Casson-du Mont[†], and Paul Chung[‡]

Artificial Intelligence Application Institute[†]
Division of Informatics,
The University of Edinburgh
80 South Bridge, Edinburgh, EH1 1HN, UK
{Peter.Jarvis, Jussi.Stader, Ann.Macintosh, Andrew.Casson-Dumont}@ed.ac.uk

Department of Chemical Engineering[‡]
Loughborough University, Loughborough,
Leicestershire, LE11 3TU, UK
{J.P.Moore, P.W.H.Chung}@lboro.ac.uk

Abstract

In this paper we describe how we are exploiting AI technologies to infuse workflow systems with adaptive capabilities. This work is part of an ongoing applied research programme between AIAI and a number of industrial and academic partners. We begin by presenting the requirements of adaptive workflow within a taxonomy consisting of the layers of domain, process, agents, organisation, and infrastructure. We then show how each level can be substantially addressed with AI technologies. Specifically, infrastructure adaptation is addressed with multi-agent toolkits, agent adaptation through knowledge-based capability matching, organisational adaptation through authority based capability matching, process adaptation through AI planning and execution architectures, and domain adaptation through rationale capture. We conclude by identifying important challenges for further work as being the improvement of rationale capture and the support for the evolution of the process models that underlie executing processes.

Keywords: Adaptive Workflow, AI Planning and Execution Systems, Capability Matching, Organisation Modelling, Authority Modelling,

Introduction

In this paper we describe how we are exploiting AI technologies to infuse workflow systems with adaptive capabilities. The work reported here is part of an ongoing applied research programme within AIAI at the University of Edinburgh which has been exploring the use of AI techniques in the realization of adaptive workflow systems since 1993. Two projects in particular are relevant: the Enterprise project (Fraser & Tate 1995; Stader 1996; Uschold et al. 1998) which was completed in 1996 and the current Task Based Process Management project which is a collaboration between AIAI and Loughborough University, UK. Applications areas considered include the bid management process (Stader 1997) and the product innovation process. Our commercial partners include BG, IBM, ICI, Lloyd's Register, Logica, and Unilever.

This paper is structured as follows. We first discuss the background business motivation for workflow and the thinking within the workflow community that has lead to the demand for developing adaptive workflow systems. We then distil the requirements of adaptive workflow from both the workflow literature and our own case studies. We use a taxonomy that divides the requirements into five relatively independent adaptation levels. We then describe our application of AI techniques at each of these levels. We conclude by describing the implementation status of our work and outlining further work.

Background to Adaptive Workflow

In the last decade there has been a significant shift in market pressures towards products that are short lived, low priced, and highly tailored to consumer requirements. To survive this transition, businesses have had to change the way in which they operate (Vlachantonis 1998). Processes are central to the operation of a business. They critically determine the type of products it can produce, the quality of its products, the rate at which it can develop new products, and the cost of its operation. Techniques developed under the banner of "Business Process Re-engineering" (BPR) aim to guide a business in improving its processes (Dellen et al. 1997). Central to these techniques is the building of explicit process models that typically capture process logic in terms of constituent activities and temporal precedence constraints between activities. Once constructed, these models serve as artifacts that can then be analyzed and redesigned to improve the overall operation of the target business.

Workflow management systems (WfMS) support the focus of process improvement through the provision of information technology support for the co-ordination, communication, and control of business processes (Joosten 1996). WfMS achieve this through the enactment of models of the type utilized by BPR techniques. Using process models to control the operation of a business directly has two classes of benefits. First, changes to the models immediately affect the operation of the business, increasing the speed at which change can be realised. Second, they assist in the co-ordination of people and distributed heterogeneous software systems working together on a common task (Georgekaopoulos and Hornick 1995). With such substantial business benefits, there are currently hundreds of WfMS products and the market is enjoying substantial growth rate (Alonso et al. 1997).

To date, the class of processes that WfMS can support is limited to simple administrative type tasks such as routine banking and insurance operations (Alonso et al. 1997; Georgakopoulos et al. 1995; Klein 1996). WfMS (and their benefits) cannot be applied to other classes of process, as the current technology does not adequately address the dynamic nature of the world (Han et al. 1998; Sheth 1997). At the heart of current WfMS is an absolute distinction between the definition and the execution of a process (Hollingsworth 1994). This distinction makes the implicit assumption that it is possible to provide a single definition for a process that is adequate for every situation in which it will be executed and every unexpected event or exception that can occur during its execution. The workflow community has realized that this build and run time distinction is inadequate for supporting all but the most simple and process. In general, at each invocation a process must be tailored to the situation in which it is to run and it must dynamically adapt to any unexpected events. The workflow community has termed the new class of WfMS that must be developed to support these requirements as "adaptive" workflow systems (Klein 1998).

Meanwhile, the AI community has been investigating intelligent systems with the capability of achieving complex tasks in dynamic and uncertain environments for over thirty years. The community has realized that the unquestioning enactment of static process models is inadequate (cf. Ginsberg's critique of Universal Planning (Ginsberg 1989)) and has instead developed rich action representations and powerful reasoning engines for dynamically generating and repairing processes. This match of technology to application requirements could assist in the fast track development of adaptive workflow (Berry & Myers 1998).

Requirements of Adaptive Workflow

With the motivation for developing adaptive workflow systems introduced, this section details the requirements that such systems must address. Han et al. (1998) provide a useful conceptual framework for categorising the requirements of adaptive workflow that we have updated in line with our own industrial experience. The taxonomy is based on the strategy of "separating concerns". It divides the classes of change that a workflow system must handle into five levels that can be examined in relative isolation. The updated taxonomy is shown in Figure 1.

Domain Level Adaptation

A deployed WfMS is configured to support the current state of a particular business. When that business changes, the WfMS must be changed. Han et al. use the domain level to differentiate between external and internal changes with respect to a WfMS. Domain level changes are external but demand a number of internal changes. The remaining layers of the taxonomy categorise the internal changes that are required to counter external change.

We strengthen the requirements of this layer to include the need to record the dependency between the internal configuration of WfMS and the specific domain features which influenced that configuration. Such explicit links will assist in identifing the facets of a WfMS configuration that must be adapted in response to given changes at the domain level.

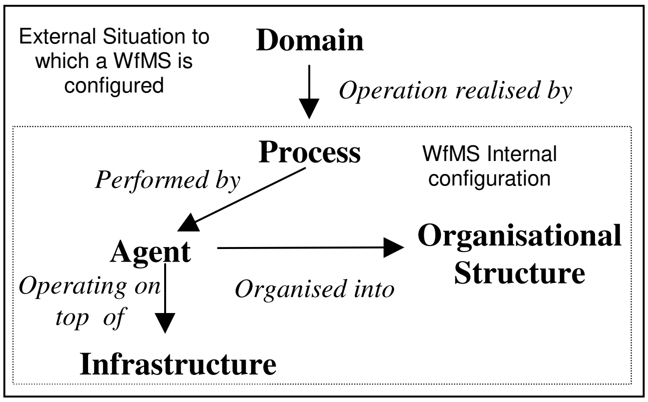

Figure 1: Levels of Workflow Adaptation, originates in Han et al. (1998)

Process Level Adaptation

The operation of an organisation at the domain level is reflected at the process level as a repository of process models. Clearly, changes in the desired operation of a business at the domain level must be reflected by changes in these models. We divide domain level change into three categories where each category poses different challenges for a WfMS.

- ***Changes in domain state between executions of a process*** result in the requirement that each execution of a process must be tailored. For example, the process to design an *artifact-a* may differ from the process to design an *artifact-b* as the artifacts are to be deployed in different countries and must be designed in accordance with different regulations. To address changes of this type, a WfMS must support the tailoring of an organisation's general "design process" to a process that meets the requirements for designing a particular artifact within a particular context.
- ***Changes in domain state during the execution of a process*** result because a process does not always proceed along the predicted path. For example, a laboratory experiment may fail to produce all the results expected of it. To address changes of this type, a WfMS must be able to adapt an executing process to changes in the domain.

- ***Explicit business process changes*** occur when a business consciously changes the way in which it operates. This poses two challenges for a WfMS. First, it must support the identification of all instances within a process repository of the process logic that is to be changed. Second, it must support adaptation of processes that are currently executing.

Additionally, the level of involvement a user wishes to have in the adaptation of a process to meet domain level changes must be considered. We have identified a continuum of user interest in the make up of a process.

- ***No concern*** is when a user has no understanding of or interest in the constitution of a process. In this case, the user expects or requires automated support for adapting a process in the light of domain changes.
- ***Full concern*** is when a user has much interest in the makeup of a process. In this case, the user expects support in considering and implementing the options available for tailoring a process.

In the processes that we have encountered, users' concern levels vary over a process. For example, a designer may wish to decide on the technical aspects of a design process yet have no interest in the make up of the financial reporting aspects.

Agent Level Adaptation

During process execution, an agent (person or software system) must be assigned to perform each activity in a process. However, the availability of a given agent is highly dynamic. Agents come and go (staff turnover, vacations, and information system updates) and become loaded with work. On each invocation of a process, a WfMS must assist in the identification of agents that are capable and available to perform its constituent activities.

Organisational Structure Level Adaptation

Agents are typically arranged into an organisational structure. For example, organisational units have people assigned to them and they may own software systems. In the context of an organisational structure, an agent may be technically able to perform an activity but not organisationally empowered to do so. A WfMS must account for these organisational norms when determining the set of agents that can perform a given activity.

Infrastructure Level Adaptation

Software systems are realised on hardware and operating system platforms. Businesses exploit technological advances by changing this underlying infrastructure. WfMS must be able to communicate with distributed and heterogeneous software systems if they are to cope with the ever-developing technical infrastructures.

The following sections outline the approaches we are taking to address requirements at each level of adaptation.

Addressing Infrastructure Adaptation with Agent Technology

Studies in distributed problem solving have considered the issue of integrating distributed heterogeneous systems. This work has resulted in the development of multi-agent architectures that provide two facilities relevant to adaptive workflow. First, they offer infrastructures that enable distributed and heterogeneous systems to communicate. This facility is directly relevant to infrastructure level adaptation. Second, they may provide a framework for dynamic capability matching. These facilities are directly relevant to agent level adaptation and will be discussed in the next section. This section focuses on infrastructure facilities.

Figure 2 illustrates the basic structure of communications within a multi-agent system. Communication between different software systems is enabled through the definition of a common Agent Communication Language (ACL) and a common message transport protocol to which all agents conform. Where existing software is required to act as an agent in the system, this is achieved by the development of a software "wrapper" which translates outgoing messages into the ACL and incoming messages from the ACL into the software's native format.

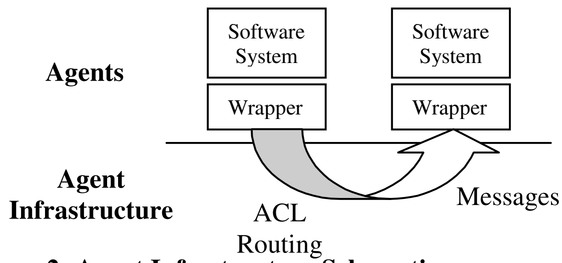

Figure 2: Agent Infrastructure Schematic

In our work, we have sought to use existing agent toolkits to address infrastructure adaptation. The common communication conduit they provide enables "wrapped" systems to communicate. We have found toolkits such as JAT*Lite* (Petrie 1996) to be adequate for research prototypes and our industrial partners have found commercial products such as GenSym Corp's ADE™ adequate for actual deployment. The one concern is the effort required in "wrapping" existing systems. There is an urgent need for "wrapper toolkits" to simplify and speed up this activity.

Addressing Agent Adaptation through Dynamic Capability Matching

Agent toolkits support the dynamic availability of agents by permitting them to register their presence and capabilities when available and to remove their registration when not available. With dynamic registration supported, the remaining challenge is the dynamic matching of the capabilities required by an activity with those of the set of agents currently available in order to find an appropriate pairing. We exploit *Knowledge-based* capability matching techniques that take into account knowledge about capabilities themselves and relationships between them.

If capability specifications are to be matched, it is important that the specifications use common and well-defined terms. We take the approach of developing a hierarchical technical capability ontology with our industrial partners, which is published as part of the Enterprise Ontology (Uschold et al. 1998). We impose more structure by splitting the capability specifications into two parts: the technical capability itself and the area (or "knowledge space") in which it can be applied. For example, if a specific database application can store data about reports, it can apply its *Store* capability to *Technical Reports*. Example terms are shown in Figure 3.

Capability Ontology	Knowledge Space Ontology
Storage Capability	Legal Entity
Store	Corporation
Store Structural	Shareholder
Store Relational	Partnership
Store Hierarchical	Document
Access	Report
Retrieve	Technical report

Figure 3: Example Ontologies

By using these hierarchical schemes in a matching function, we can not only determine which agents match the capability requirements of an activity exactly, but we can rank all agents available at the time of execution according to how closely they match the capability requirements. We apply the ranking heuristic that exact matches are best, but agents that can apply the required capability in a wider area than required are nearly as suitable. Similarly, agents that have a more general capability are suitable, although more specialised agents would be preferred because they are likely to perform the activity more effectively.

In summary, by providing a well-defined ontology of capability and knowledge space terms, statements about capabilities can be matched consistently. The use of a generalisation structure within such an ontology enables a workflow system to apply "generalist vs. specialist" heuristics to rank the available agents. The features combine to address the WfMS requirements of agent level adaptation by enabling the most effective agent for performing an activity to be dynamically identified.

Addressing Organisational Adaptation through Enhancements to Capability Matching

The importance of at least sensitizing a workflow system to the organisational structure and authority context within which it operates is well argued for in the literature (cf. Dellen et al. 1997, Joosten 1996, Kappel et al. 1995 Rupietta 1997). If this context is ignored, the system will undoubtedly break organisational conventions. We argue that rather than just being sensitized to organisational structure and authority issues, workflow systems should be provided with explicit representations of this knowledge. With this knowledge it can proactively guide a user's decision-making by highlighting how the existing organisational structure can be navigated and authority constraints maintained. Our proposed framework for modelling organisational structure and authority is outlined in the following sections.

Organisational Unit: An entity responsible for managing the performance of activities to achieve one or more purposes. An organisational unit can be used to describe departments, working groups, projects etc.
Agent: An entity that can perform an activity.
Person: A human being.
Machine: A non-human entity that has the capacity to carry out functions. A machine is similar to a person. However, it is anticipated that some functions and roles are exclusive to one or the other. For example a machine cannot be held responsible for anything.
Manages-Organisational Unit to Organisational Unit: An *organisational unit* can *manage* an *organisational unit*. With this relationship, one *organisational unit* takes on the role of the *manager* and the second organisational unit the role of the *managee.*

Figure 4: Concepts in the Organisational Structure-Modelling Framework

Organisation Modelling Language

A modelling language for describing organisational structure must contain constructs for modelling a wide variety of organisations. Our language is based upon the one published as part of the Enterprise Ontology (Uschold et al. 1998). We are confident in the generality and adequacy of this ontology as it was developed by a working group that included representatives from three international organisations and it is similar to others that have been developed, independently, for similar purposes (cf. Hoog et al 1997). The framework is centred on the *organisational unit* concept that can be used to describe departments, divisions, projects, working groups etc. The definitions in Figure 4 outline the central concepts within our organisational modelling framework.

Organisational units can be connected by a number of relationships (Figure 4). The "*manages"* relationship can be used to represent the subdivision of *organisational* *units*; a committee into working groups, for example. Both *machine* and *person* are *agents* and can be linked to *organisational units* through relationships. A *person,* for example, may be related to an *organisational unit* through the "*manages*" relationship, taking the role of a *manager*.

Authority Modelling Framework

Our authority modelling framework aims to provide constructs that can be used in conjunction with an organisational model to define the authority relationships within an organisation. The model is based upon the following authority primitives:

- **Obliged:** an agent is obliged to provide a capability.
- **Permitted:** an agent may decide itself whether or not to provide a capability.
- **Forbidden:** an agent must not provide a capability.

These primitives are taken from the field of Deontology[1]; which aims to describe the duties and responsibilities of individuals, and has been used in a number of areas within computer science (Wieringa & Meyer 1993). We illustrate these constructs with the example shown in Figure 5.

Capability:	Retrieve (Person's Salary)
Obliged:	If the person requesting the salary details is the manager of the organisational unit that the secretary works in.
Permitted:	If a person is requesting his or her own salary details.
Forbidden:	To the rest of the world.

Figure 5: Authority Context of the Secretary's Capabilities

The approach described above enables a workflow system to account for organisational and authority properties when matching activities and agents and therefore to respect organisational norms. Separating these concerns from agents' technical capabilities simplifies the adaptation of a WfMS to organisational restructuring as only the authority and organisational models need modification to reflect such changes. Our approach to authority modelling is discussed in detail in (Jarvis et al. 1999).

Addressing Process Adaptation through Automated Planning Architectures

We are working on process adaptation issues with an architecture based on the automated planning system O-Plan (Currie and Tate 1991) and the <I-N-OVA> model of activity (Tate 1996a). In the following sections we outline the operation of the research workbench we have built, called the *Task-Based Process Manager Workbench*, then show how this system is being used to explore support for process adaptation requirements. The system also supports the agent and organisational adaptation issues discussed in the previous sections. The underlying definition of process and activity used in the workbench draws from the recent SPAR (Tate 1998) standard.

Operational Overview of the Task-Based Process Manager

Figure 6 shows an example of the operation of the Task Based Process Manager Workbench (TBPM-W) in terms of the process knowledge it manipulates. In this case, the user Peter has requested that the TBPM-W support him in the achievement of an instance of the task α. In response to this request, the system creates the process structure shown, consisting of *issue*, *node* and *detailed* constraints. The *node* constraints correspond to the activities within a process. As the task has just been initiated, the node constraints contain the boundary *start* (St) and *finish* (Fn) nodes together with a node for the task just initiated, α. The *detailed* constraints relate to activities and include temporal precedence, pre and post conditions (such as information flow), and resource constraints.

For clarity, only the temporal precedence constraints are shown in the figure. There are two classes of temporal precedence constraints. *Execution Precedence* constraints determine the order in which activities are to be executed and are shown with pointed arrowheads. *Planning Precedence* constraints determine the order in which activities are to be planned and are shown with diamond arrowheads. *Issue* constraints refer to the items that must be done in order to complete task α.

As the task has only just been initiated, the only outstanding issue is the planning of the task. The issue type denotes that this is a planning issue while the status "*Ready"* indicates that this issue is ready to be addressed and the user field identifies this issue as assigned to the user "*Peter"*. The issue has been assigned by default to the user who initiated the task. However, making the assigned user explicit enables the transfer of issues to other users for handling. For example, consider the case of a manager initiating a task who then delegates its planning and execution of a task to a subordinate.

To handle a planning issue, a user must identify a *method* that further describes how the task to which the issue relates is to be achieved. The user is supported in this by being offered alternatives from a process library. Processes are indexed in the library by the name of task or higher level actions that they can be used to refine.

[1] *Greek: deon "duty", and logos "science".*

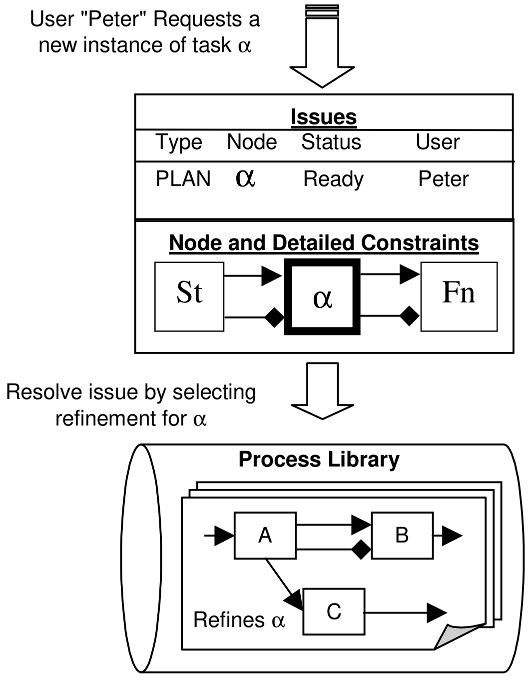

Figure 6 Task Initiation

Once the user has selected a method for refining the task, the TBPM-W updates its *Issue*, *Node*, and *Detailed* constraints to those shown in Figure 7. The α node constraint has been replaced with the constituent activities of the method the user selected. In this case, *A, B,* and *C.* The detailed constraints are also updated in line with those in the selected method. The system also updated the *Issue* constraints to note that the "*Plan α*" issue has been completed (status to "*Complete"*) and to include the new issues posed by the chosen method. In the example, nodes *A* and *C* require further refinement before an executable process will be defined. Each is represented by a planning issue. Node *B* is immediately executable and therefore raises an execution issue. The status of each issue reflects the detailed constraints between nodes. In the example, the issues associated with nodes *A* and *C* are ready to be addressed and the issue with node *B* is not ready, and is therefore assigned the status "*Not-Ready*". The planning issue with node *A* is ready to be handled as it is only related through temporal precedence with the start node of the process. The start node of a process is always considered as executed. Node *C* is only constrained to be executed after node *A*. This constraint means that it is safe to plan node *C* before the planning of node *A* is completed but the activities introduced as refinements of node *C* must wait until those of node *A* have been executed. If there had been a planning precedence relationship between nodes *A* and *C,* the issue relating to the planning of *C* would be marked as "preconditions not met". As node *B* is constrained to be executed after node *A,* the execution issue relating to *B* is marked as Not-Ready.

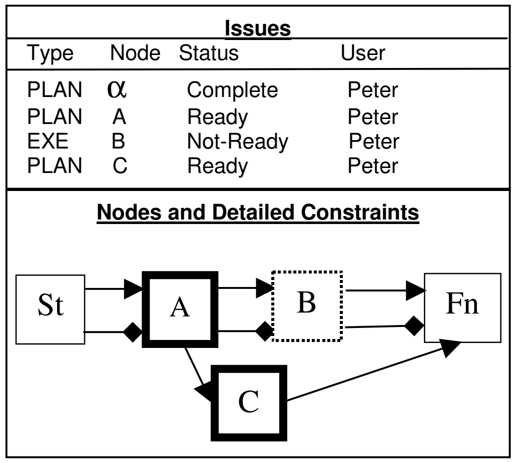

Figure 7 State after the refinement for task α has been implemented

The workbench posts the issues that can currently be addressed to the to-do list of the user to which they are assigned. The assigned user is free to assign an issue to another user. This feature gives users a fine granularity in distributing both the planning and execution of a process to other users. The assignment protocol implemented is shown in Figure 8 through a state transition chart. The person to whom an issue is assigned can either accept or reject that issue. We are exploring approaches to exploiting the authority model described earlier to tailor the dialogue styles used during this interaction. For example, one may wish to use a different dialogue style when rejecting an issue assignment request originating from a superior to one originating from a subordinate.

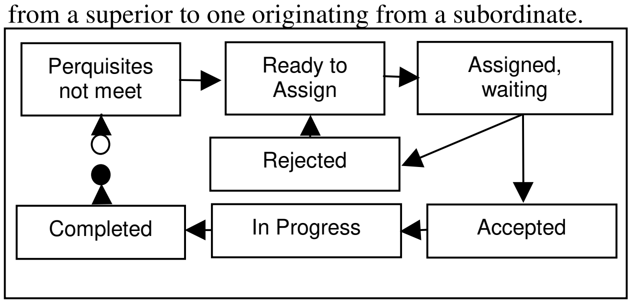

Figure 8: Issue Life History

With the basic operation of the TBPM-W introduced, we now consider how it can be used to support the process adaptation requirements of adaptive workflow.

Addressing Domain State Changes between Executions of a Process

The approach of interleaving the planning and execution of a task taken in the TBPM-W replaces the current WfMS distinction between the definition and execution of a process. In TBPM-W, the user is supported in the assembly of small process fragments or methods to form a process configuration tailored to the current situation. The workbench uses the detailed constraints attached to methods to assist the user in choosing between the methods. The hierarchical approach taken supports the user in making high level "strategic" decisions about the form of a process before moving down to lower level decisions. Decisions at any level are constrained with respect to the framework established at the higher levels

Domain State Changes during the Execution of a Process

Domain state changes invariably occur during the execution of a process. A WfMS must support a user in adapting his or her original process to these changes. We are investigating two mechanisms for providing this support. First, the general TBPM-W approach of interleaving planning and execution assists the user in delaying decisions about how part of a process is to proceed until the activities that affect its course have been executed and their outcome is known. Second, we are exploring plan repair strategies of the type developed by Drabble et al. (1998). An example domain state change is shown in Figure 9 and its repair is shown in Figure 10. The example is taken from the chemical engineering domain. At the top of Figure 9 is the planned execution of a process. It is assumed that "experiment one" will, amongst other things, produce "Result A: Known". This result being available is the prerequisite for a second experiment "experiment 2". The workbench monitors the execution of a process to check if the actual outcomes of activities correlate with the planned outcomes. The dotted line in the Figure 9 denotes the execution fringe, i.e. the point at which the execution of the process has reached. During the execution, activity "experiment 1" has produced the unexpected result of "Result A: Unknown". The execution monitoring examines the detailed constraints stored by the workbench to identify if this deviation affects the process in any way. In this case it does, as the expected result is a prerequisite for "experiment 2". With the "damage" to the process identified, plan repair strategies are invoked to attempt to recover the process. In the first instance the process is examined for other activities that produce the desired effect. Experiments, for example, may overlap in their results. In such cases, identifying other contributors to a precondition and using them to repair the "damage" can repair the process. In this case, assume that there are no other possible contributors. To repair the "damage", the TBPM-W must search the process library to identify activities that can be introduced to establish the precondition of "experiment two". Figure 10 shows the application of such a "patch". Here, the workbench has identified a method that takes fact "Result A: Unknown" and changes its state to "Known". In the experiment example, the patch entails repeating only the part of "experiment 1" that was necessary to produce the state of "Result A: Available".

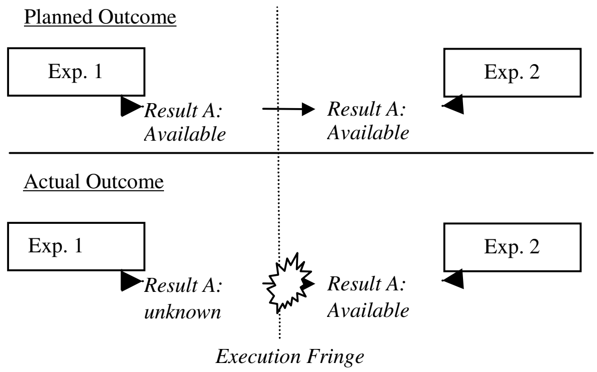

Figure 9: Planned and Actual Outcome of a Process

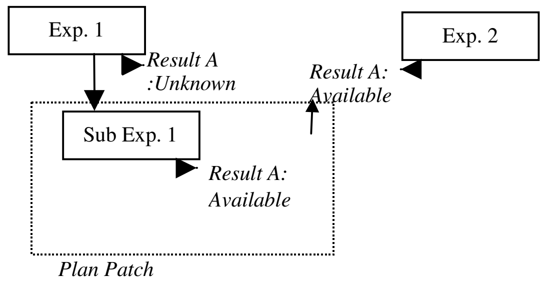

Figure 10: Example Plan Patch

Explicit Business Process Changes

In our approach, explicit business process changes imply changes in the methods available for configuring a process to achieve a task. If such changes affect parts of a process that have not been executed, it is simple to accommodate them by supporting the user in replacing the methods they have selected with new methods. The complex case is when the changes affect parts of the processes that have already been executed and parts that are currently being executed. We have not yet tackled this case. We plan to investigate the use of the "plan patch" approach to identify the difference between the new and old methods and to support the user in moving between them.

Continuum of User Concerns

Our issue-based approach gives a user freedom to select the parts of a process they wish to plan and those parts they wish to delegate to other users or to the system for planning. This is achieved through making workflow issues explicit and augmenting them with the identity of the agent currently responsible for resolving them.

Addressing Domain Adaptation through Rationale Capture

A WfMS is internally configured to support a business in its current state. When the business changes, the configuration of the WfMS must change. We are exploring mechanisms for attaching rationale to the internal configuration of a WfMS to simplify the identification of parts of that configuration that must be changed in response to a given domain level change.

To date, we have considered rationale behind the initiation of tasks. Figure 11 shows an example rationale structure. It is based on Petrie's (1993) Redux' approach to decision documentation. Here the top-level business goal of "Investigate the Scaling of Reaction X" is shown as a goal that is satisfied by the achievement of two sub goals (investigation of temperature and pressure). The top-level business goal is augmented with the assumption under which it is being carried out. In this case, the assumption is that there is a need for a new product, Y. The sub goals result in decisions to perform experiments. In the case of the pressure experiment, the assumption behind the pressure experiment is recorded. The advantage of keeping this information is that if the business changes, affected tasks can be identified. For example, if the business decides that it no longer wishes to explore product Y, the system can automatically identify "experiment 1" and "experiment 2" as tasks that need to be reconsidered and possibly stopped as a result. Likewise, if the price of water increases, "experiment 1" will be identified as a task that needs to be reconsidered. Petrie et al. (1998) are carrying out important related work in this area.

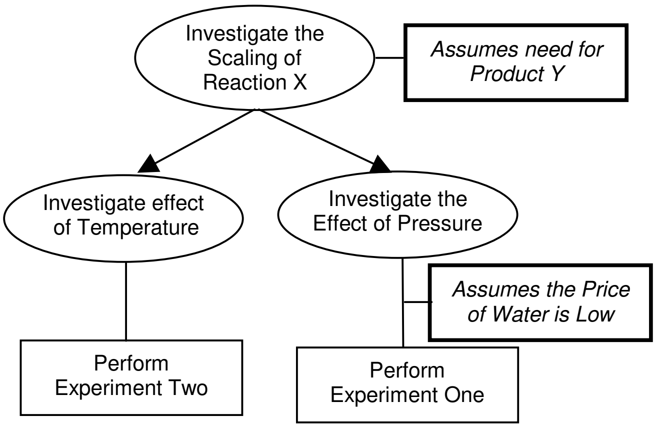

Figure 11: Example Rationale Structure

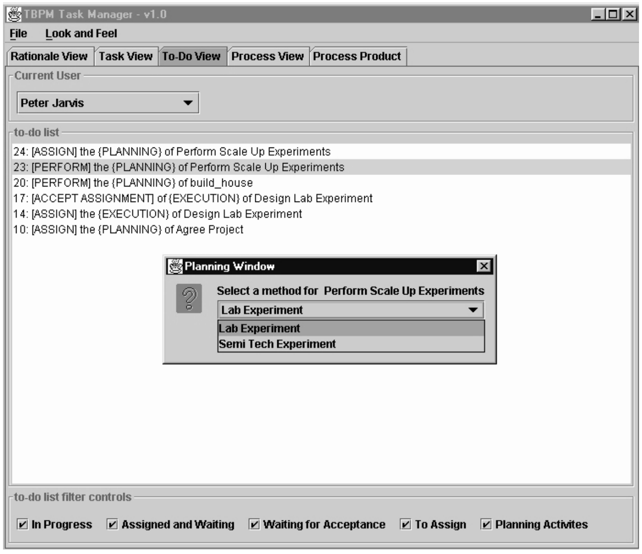

Figure 12: TBPM-W To-Do List View

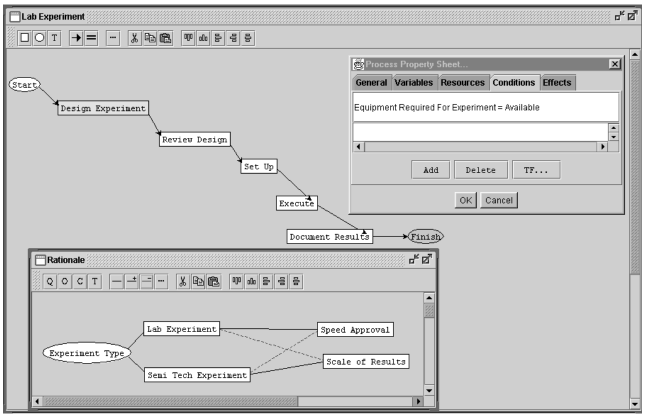

Figure 13: Integration with Polyak's Common Process Editor

Implementation Status

The TBPM-W is being implemented as a test bed for developing and demonstrating the concepts presented in this paper. Figure 11 shows the to-do list view of the system. In the figure, the user has selected the issue of planning in more detail how the task "Perform Scale Up Experiments" is to be performed. The "Planning Window" in the centre of the figure is presenting the two options available for refining this task. Once the user selects a method, the issue will be resolved and the constraints maintained by the system updated to include the consistent activities of the method. Figure 13 shows how we are integrating TBPM-W with Polyak's Common Process Editor (Tate et al. 1998) to visualise process structure. In the figure the constituent activities of the "Lab Experiment" option are shown. The editor enables the user to browse and edit hierarchical process structure and process constraints while also providing a view of the design rationale behind the artifact.

Conclusion

In this paper we have outlined how AI techniques can be used to address a significant subset of the requirements of adaptive workflow. The work reported is based on an applied research programme at AIAI that has been examining the use of AI techniques within workflow over the past six years. Figure 14 summarises the AI applicable to realizing adaptive workflow

Level	Applicable AI Technology
Domain	Rationale maintenance
Process	Planning and execution architectures
Organisation	Capability matching supported by organisation and authority models
Agent	Dynamic capability matching
Infrastructure	Multi-agent toolkits

Figure 14: Summary Of the AI Technologies Applicable to Realizing Adaptive Workflow

Our work has shown that there is a strong mapping between the requirements of adaptive workflow systems and capabilities offered by AI techniques. We plan to continue exploring the approaches outlined here. Particular future challenges lie in the linking of business rationale throughout the four internal layers of a WfMS and in the evolution of the process models that underpin currently executing processes.

References

Alonso, G., Agrawal, D., El Abbadi, A., and Mohan, C., 1997, Functionality and Limitations of Current Workflow Management Systems. *IEEE-Expert.*

Currie. K., and Tate. A., 1991, O-Plan: the Open Planning Architecture. *Artificial Intelligence*, vol. 51, part 1, North-Holland.

Drabble, B., Dalton, J., and Tate, A., 1997, Repairing Plans on the Fly. Proc. of the NASA Workshop on Planning and Scheduling for Space, Oxnard CA, USA.

Fraser, J., and Tate, A., 1995, The Enterprise Tool Set -- An Open Enterprise Architecture. Proc. Workshop on Intelligent Manufacturing Systems, held during the International Joint Conference on Artificial Intelligence (IJCAI-95), Montreal, Canada, 1995.

Georgakopoulos, D., Hornick, M., and Sheth, A., 1995, An Overview of Workflow Management: From Process Modelling to Workflow Automation Infrastructure. *Distributed and Parallel Databases*, Vol 3, pp 119-153, 1995.

Ginsberg, M., 1989, Universal Planning: an (almost) Universally Bad Idea, *AI Magazine,* part 10, pp. 49-44.

Han, Y., Sheith, A., and Bussler, C., 1998, A Taxonomy of Adaptive Workflow Management. Proc. CSCW-98 Workshop Towards Adaptive Workflow System, held during the 1998 Conference on Computer-Supported Cooperative Work in Seattle, USA,

Hoog, R., 1994, Organisation Model: Model Definition Document. University of Amsterdam Technical Report, KADSII/M6/UvA/041/3.0, Netherlands.

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

Jarvis, P. Stader, J. Macintosh, A. Moore. J., and Chung, P, 1999, What Right Do You Have to Do That? Infusing Adaptive Workflow Technology with Knowledge about the Organisational and Authority Context of a Task. Proc. First International Conference on Enterprise Information Systems (ICEIS-99), Setubal, Portugal.

Joosten, S., 1996, Workflow Management Research Area Overview. Proc. 2nd Americas Conference on Information Systems, Arizona, pp. 914-916.

Kappel, G., Lang, P., Rausch-Schott, S., and Retschitzegger, R., 1995, Workflow Management Based on Objects, Rules, and Roles. *IEEE Bulletin of the Technical Committee on Data Engineering*, Vol. 18, No. 1, pp11-17.

Klein, M., 1996, Challenges and Directions for Coordination Science. Proc. 2nd International Conference on the Design of Cooperative Systems (COOP'96), France.

Klein, M. (ed.), 1998, Proc. CSCW-98 Workshop Towards Adaptive Workflow System, held during the 1998 Conference on Computer-Supported Cooperative Work in Seattle, USA, on-line proceedings available at: http://ccs.mit.edu/klein/cscw98/.

Petrie, C.,1993, The Redux' Server. Proc. International Conference on Intelligent and Cooperative Information Systems (ICICIS), Rotterdam, May

Petrie, C, 1996, Agent-Based Engineering, the Web, and Intelligence. *IEEE Expert*.

Petrie, C., Goldmann, S., and Raquet, A., 1998, Agent-Based Project Management'. Proc. International Workshop on Intelligent Agents in CSCW, Dortmound, Sept., p. 1-17.

Polyak, S. and Tate, A., 1998, Rationale in Planning: Causality, Dependencies, and Decisions. *Knowledge Engineering Review*, 13(3), September, pp. 247-262.

Rupietta, W., 1997, Organisation and Role Models for Workflow Processes. in *Workflow Handbook*, P. Lawrence (ed.), Wiley. 1997,

Sheth, A., 1997, From Contemporary Workflow Process Automation to Adaptive and Dynamic Work Activity Coordination and Collaboration. Proc. Workshop on Workflows in Scientific and Engineering Applications, France, September.

Sheth, A., and Kochut. K., 1997, Workflow Applications to Research Agenda: Scaleable and Dynamic Work Coordination and Collaboration Systems. Proc. NATO Workshop on Advances in Workflow Management Systems and Interoperability, Turkey.

Stader, J., 1996, Results of the Enterprise Project. Proc. 16th Int. Conference of the British Computer Society Specialist Group on Expert Systems, Cambridge, UK, 1996.

Stader, J., 1997, An Intelligent System For Bid Management. *The International Journal of Project and Business Risk Management*, Vol 1, Part 3, pp 299 - 314, 1997.

Tate. A., 1996a, Representing plans as a Set of Constraints - the <I-N-OVA> Model. Proc. The Third International Conference on Artificial Intelligence Planning Systems, Edinburgh, Scotland, AAAI Press.

Tate. A., 1996b, Responsive planning and scheduling using AI planning techniques - Optimum-AIV. *IEEE Expert Intelligent Systems & Their Applications,* Vol. 11, No. 6, December, pp 4 - 12, IEEE.

Tate, A., 1998, Roots of SPAR - Shared Planning and Activity Representation. *The Knowledge Engineering Review*, Vol 13(1).

Tate, A. Polyak, S. and Jarvis, P., 1998, TF Method: An Initial Framework for Modelling and Analysing Planning Domains. Proc. Workshop on Knowledge Engineering and Acquisition for Planning held within AIPS-98, Pittsburgh, USA.

Uschold, M., King, M., Moralee, S., and Zorgios, Y., 1998, The Enterprise Ontology. *The Knowledge Engineering Review*, Vol. 13

Vlachantonis, N., 1997, Workflow Applications Within Business Organizations. *Workflow Handbook,* Ed. P.

Wieringa, R., and Meyer, J., 1993, Applications of Denotic Logic in Computer Science: A Concise Overview. *Deontic Logic in Computer Science,* R. Wieringa and J. Meyer (eds.), Wiley, 1993.

Acknowledgements

The work described in this paper can be accredited to two main projects. The Enterprise project[2] was funded under the UK Government's *Intelligent Systems Integration Programme*. It was carried out at AIAI, the University of Edinburgh, UK, with input from industrial partners IBM UK, Lloyd's Register, Logica, and Unilever. The Task-Based Process Management project[3] is funded under the UK Engineering and Physical Sciences Research Council's *Systems Engineering for Business Process Change* programme. It is a collaboration between AIAI and the Chemical Engineering Department at Loughborough University, UK. Industrial partners on this project are BG Technology, ICI, and Unilever. Steve Polyak's work is on-line at: http://www.dai.ed.ac.uk/students/stevep/

We thank Stuart Aitken, Dave Bustard, Barbara Dellen, Brian Drabble, Sigrid Goldmann, Charles Petrie, Steve Polyak, and Austin Tate for enlightening discussions

---

2 Details available at http://www.aiai.ed.ac.uk/project/enterprise/

3 Details available at http://www.aiai.ed.ac.uk/project/tbpm/tbpm.html