1

A Framework for Equipping Workflow Systems with Knowledge about Organisational Structure and Authority

Peter Jarvis[ϕ], Jussi Stader[ϕ], Ann Macintosh[ϕ], Jonathan Moore[ψ], and Paul Chung[ψ]

[ϕ]Artificial Intelligence Applications Institute,
Division of Informatics, The University of Edinburgh
80 South Bridge, Edinburgh, EH1 1HN, UK.

[ψ]Department of Chemical Engineering
Loughborough University
Loughborough, Leicestershire, LE11 3TU, UK.

{Peter.Jarvis, Jussi.Stader, Ann.Macintosh}@ed.ac.uk
{J.P.Moore, P.W.H.Chung}@lboro.ac.uk

ABSTRACT

To achieve more widespread application, workflow systems need to be developed to operate in dynamic environments where they are expected to ensure that users are supported in performing flexible and creative tasks while maintaining organisational norms. We argue that in order to cope with these demands, the systems must be provided with knowledge about the organisational structure and authority context of tasks. We support this argument by identifying a number of decision points that an adaptive workflow system must support, discussing how these decisions can be supported with technically oriented capability specifications, and describe how this support can be enhanced with the inclusion of knowledge about organisational structure and authority. We outline how such knowledge can be captured, structured, and represented in a workflow system. We then demonstrate the use of such knowledge by describing how the task initiation, task planning, activity scheduling, and agent interaction functions within a workflow system can be enhanced by it.

Key Words: Adaptive Workflow, Capability Matching, Organisational Modelling, Authority Modelling.

INTRODUCTION

There is a consensus in the workflow community that the application of systems based upon the unquestioned enactment of process models is inherently limited, particularly in dynamic working environments [1] [4] [8] [13] [14]. To move forward, the workflow field needs to develop techniques that support people in making decisions about the activities required to achieve a task, the execution order of those activities, and the specific agents that will perform each activity. In order to provide such support, knowledge about tasks and agents is required. To provide a frame of reference we have identified three main stages of workflow in which the decisions above are made:

- <u>Task Initiation</u>: is the stage where a user instructs a workflow system to begin a new task. There are choices of which task to initiate.
- <u>Task Planning</u>: is the stage where the activities required to achieve a task are determined. There are choices of which method to use.
- <u>Activity Scheduling</u>: is the stage where a specific agent is assigned the role of performing an activity. There are choices of who the activity should be assigned to.

We use these stages throughout the paper to show how different types of knowledge can provide different levels of decision support during the stages.

In this paper, we introduce our approach of “knowledge-based capability matching” which concentrates on the use of knowledge related to capabilities to support the decisions above. We first introduce the basic support that can be provided using knowledge about technical capabilities. However, organisations, are typically arranged into an organisational structure that defines the division of labour and communication channels that together provide a regulated co-operation between its members [10]. Adding knowledge about this context of an activity in terms of organisational structure and authority, we can provide more sophisticated support. We show how this can be done by outlining a specification schema for describing the knowledge required and discussing how a workflow system can make use of it.

Throughout the paper, the term “task” corresponds to a goal or objective that is to be achieved. Typically to achieve a task, a number of “activities” have to be performed by “agents” (people or software systems). There may be different “methods” specified of how to achieve a task, each consisting of a number of activities. Once a “method” is selected, its constituent activities define the “process” for achieving a particular task.

The work reported here is part of an ongoing applied research programme at AIAI, University of Edinburgh, UK, investigating and developing techniques for supporting complex tasks within dynamic environments. Two projects in particular are relevant: the Enterprise project [3] [11] [16] which was completed in 1997 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 [12] and the product innovation process. Our commercial partners include: IBM, Lloyd’s Register, Logica, Unilever, ICI, and BG.

KNOWLEDGE-BASED TECHNICAL CAPABILITY MATCHING

In a capability matching function, the capabilities required by an activity are matched against the capabilities held by available agents in order to identify suitable agents for performing the activity. *Knowledge-based* capability matching refers to the more sophisticated matching that takes into account knowledge about capabilities themselves and relationships between them. The reason for using any such matching function in a workflow context is that it is impossible to predict the exact environment in which a task is executed. Similarly, specific agents may not be available at the time of execution (people take holidays or leave the organisation), or more suitable agents may have become available (people are hired and new software systems are developed). Similarly, activities may not be required in the specific context of a task's execution. Availability of agents not only has an impact on assigning activities to agents, but also on the decision of which method is chosen to achieve a given task. If a method for carrying out a task requires a particular capability but there are currently no agents available with that capability, then the task must be achieved using an alternative method.

Clearly, capability matching requires specifications of capabilities both for activities and for agents. If capability specifications are to be matched, it is important that the specifications use common and well-defined terms that can be related to each other. Our approach during the Enterprise project was to develop a technical capability ontology with our industrial partners which is published as part of the Enterprise Ontology [16]. This technical capability ontology provides a hierarchy of capabilities. For example, databases provide a *Store* capability and - more specifically - relational databases provide a *Store Relational* capability. This hierarchical structure can ease the task of specifying capabilities (required or available) because specifying a high-level capability implies that all its lower-levels are covered too.

However, in our experience organising all terms required for specifying capabilities of agents and activities into such a hierarchy is too big a task for any realistic application area. We decided to impose more structure by splitting the 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 skills, it can apply its *Store* capability to *Technical reports*. Each of the parts uses its own hierarchy of terms shown in part in Figure 1.

Capability Ontology

- Storage Capability
  - Store
    - Store Structural
      - Store Relational
      - Store Hierarchical
  - Access
    - Retrieve

Knowledge Space Ontology

- Legal Entity
  - Corporation
  - Shareholder
  - Partnership
- Document
  - Report
    - Technical report

Figure 1: Example Capability and Knowledge Space Elements from the Enterprise Ontology

Using this specification schema in a matching function, the workflow support we implemented during the Enterprise project can not only determine which agents match the capability requirements of an activity exactly, but it can rank all agents available at the time of execution according to how closely they match the capability requirements. Exact matches of a capability specification 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 made and matched consistently. The use of a generalisation structure within the ontology simplifies the specification of capabilities. It can also be exploited by a workflow system to apply "generalist vs. specialist" heuristics and make the best use of the agents available during a task's execution. Our experience with the Enterprise project shows that a technical capability matching function can underpin the three workflow stages as follows:

- Task Initiation: Using specifications of technical capability requirements of tasks, users can be assisted in choosing only tasks that they themselves can initiate and manage.
- Task Planning: A technical capability matching function can help a user rank alternative methods for tackling a task by considering how much agent support is currently available for each approach.
- Activity Scheduling: A technical capability matching function can help a user dynamically identify and rank potential performers of an activity so that the most suitable agent can be selected.

Motivation for Adding Knowledge about Organisation and Authority

The importance of at least sensitising a workflow system to the organisational structure and authority context within which it operates is well argued in the literature (cf. [2], [6], [7], [10], [15], and [10]). If this context is ignored, the system will undoubtedly break organisational conventions. For example, activities may be assigned to people who are not related to the organisational unit responsible for the overall task. We argue that a workflow system can be much more than just sensitised to organisational structure and authority issues. With this knowledge it can proactively guide a user's decision making by highlighting how an organisational structure can be navigated and authority constraints maintained. Using our three stages of workflow, the following additional support can be provided:

- Task Initiation: After determining that a user has the technical capability to initiate and manage a task, the system can use authority knowledge to determine if the agent has the authority to take this action. If an agent does not have the necessary authority, the system can advise the agent on who to ask for it.
- Task Planning: After determining if there are agents technically qualified to complete a task, the system can assess relationships between the agent

managing the task and the agents that will be asked to perform activities. Methods for achieving a task can then be ranked using an "organisation and authority metric". For example, a method where all the agents required are employed in the same department as the agent managing the task may be considered preferable to one where the agents are distributed across departments.

- Activity Scheduling: After determining the set of agents available for performing a task through technical capability criteria, the set can be ranked according to the authority held by the agent requesting the assignment. For example, it may be considered better to assign activities to agents that one has direct authority over.

Activity scheduling requires agents to communicate not just about results of activities but also about the delegation of activities between agents. With knowledge about organisational structure and authority we argue that support for such communication can be enriched as follows:

- Interaction Styles: If a system understands the relative authorities between two individuals, it can assist in the selection of appropriate interaction styles for the communication between them. For example, the system can prioritise entries on a to-do list according to the seniority of the individual requesting the action, and arranging dialogues so that "accept" becomes the default option when responding to a superior's request.

ORGANISATIONAL STRUCTURE MODELLING LANGUAGE

A modelling language for describing organisational structure must contain constructs that can be used to model a wide variety of organisations. Our language is based upon the one published as part of the Enterprise Ontology [16] [17]. 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. [5] [10]). The language is centred around the *organisational unit* concept which can be used to describe departments, divisions, projects, working groups etc. The definitions in Figure 2 outline the central concepts within our organisational modelling language.

Organisational units can be connected by a number of relationships (Figure 3). 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*.

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, project 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 the relationship, one *organisational unit* takes on the role of the *manger* and the second organisational unit the role of *managed by.*

Figure 2: Definition of the Concepts in the Organisational Modelling Language

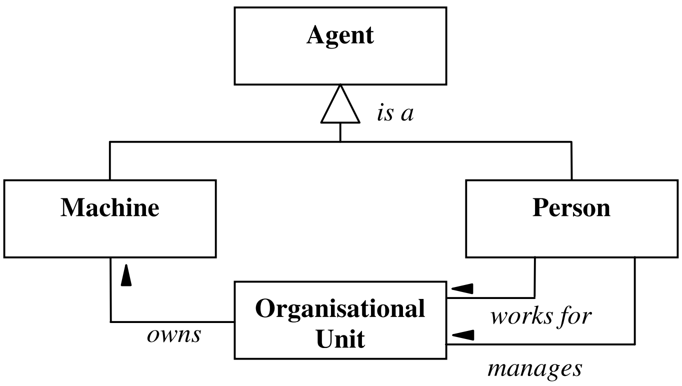

Figure 3: Graphical Representation of the Organisational Structure Modelling Language

AUTHORITY MODELLING LANGUAGE PROPOSAL

Our authority modelling language 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 perform an action.
- Permitted: the agent may decide itself whether or not to perform an action.
- Forbidden: an agent must not perform an action.

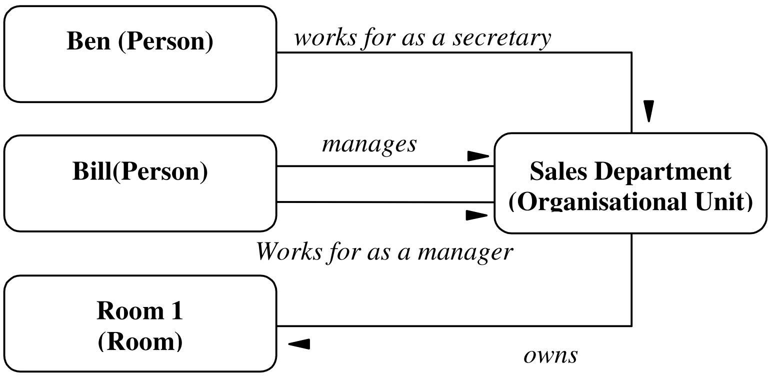

Figure 4: Example Organisational Model

These primitives are taken from the Deontology[1] field; a field aiming to describe the duties and responsibilities of individuals that has been used in a number of areas within computer science [18]. The primitives have been successfully applied to authority modelling issues in other areas of Computer Supported Cooperative Work. Liu and Dix [9], for example, use them to endow computer agents with knowledge about their responsibilities.

We illustrate these constructs with an example. Consider a sales department that has a manager, a secretary, and a meeting room. This situation is shown using our organisational modelling language in Figure 4.

Consider two capabilities that a secretary can provide: the retrieving of salary details, and the booking of rooms. These capabilities are specified below using our technical capability and knowledge space approach.

---

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

Capability Specification: *Secretary*
Retrieve (Person's Salary)
Book (Room)

Figure 5: Example Capability Specifications

The example shown in Figure 6 adds the authority context to the secretarial capabilities introduced in Figure 5. Considering first the *Retrieve* capability, the “obliged” slot states that the secretary must provide the capability if the person requesting it manages the organisational unit for which he or she is a secretary. The “permitted” slot states that the secretary is free to choose between answering or not answering a request made by an individual who is asking for their own salary details. The “forbidden” attribute states that the secretary must not return salary details to anybody who does not meet the criteria stated in the “obliged” or “permitted” attributes. The authority statements added to the *Book* capability state that the secretary must book the room for people working in the organisational unit that owns the room. The secretary is permitted to accept bookings from individuals who work for the overall organisation that the secretary’s organisational unit is a part but is forbidden to accept bookings from individuals who do not meet the “obliged” or “permitted” criteria.

Authority Specification
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.
Capability: Book(Room)
Obliged: If the person requesting the room works in the Organisational unit that owns the room.
Permitted: If the person requesting the room works in the company that owns the organisational unit that owns the room.
Forbidden: To the rest of the world.

Figure 6: Authority Context of the Secretary’s Capabilities

The “Culture” Perspective

The authority specifications in Figure 6 are from what we term the “Capability” perceptive as they each apply to a single specified capability. Our “Culture”

Perspective is designed to complement the "Capability" Perspective with the specification of general authority statements that reflect the authority culture within an organisation. For example, a manager may have authority over all the agents working in or owned by the department(s) he or she manages and thus the capabilities provided by these individuals. Figure 7 outlines such a specification. The essential difference between this culture related specification than that in Figure 6 is the inclusion of "for all" type statements that are applicable over a set of capabilities. In the detailed perspective above, authority statements are specified in terms of a single and explicitly stated capability.

Capability**:** For all capabilities possessed by people who work in the organisational unit that is managed by a particular manager.
Obliged: to accept requests from that manager

Figure 7: Example Capability Specifications for a Manager Role

Whilst the culture perspective allows a single authority statement to be applied to many technical capabilities with the benefit of reducing modelling effort, the perspective is less precise than the detailed perspective. The two specifications given in Figure 6 and Figure 7 are currently consistent. Figure 6 states that the secretary is obliged to answer salary retrieval requests from his or her manager. Figure 7 implies that the secretary is obliged to answer all requests from his or her manager. If the statement in Figure 6 is modified to state that the secretary is permitted only to return salary information to the individual to which it applies, then the two specifications contradict each other. The capability perspective is now stating that the request cannot be answered whilst the cultural perspective is stating that the request must be answered. To resolve such conflicts, a system working with authority specifications must employ some conflict resolution strategy. A general approach can be to always favour the capability oriented statement as this perspective is more detailed. However, the system may simply identify the conflict and allow the users to resolve it.

Using Organisational Structure and Authority in Workflow

The specifications of organisational structure and authority described above can be used to support our identified main stages of workflow:

- <u>Task Initiation</u>: If task specifications include statements of authority, a workflow system can ensure that only agents who meet the requirements specified in those statements can initiate a task. If an agent does not have the necessary authority, the system can identify agents that do and suggest that the initiating agent discuss the task with one of these agents.

- Task Planning: Considering the availability of agents for performing each method a workflow system can rank methods for achieving a task using an "authority metric". A method where activities can be assigned to agents with a high level of obligation to perform their activities may be considered preferable to a method where agents are only "permitted" to perform the activities. Methods for which the only agents available are forbidden to perform their activities can be discounted.
- Activity Scheduling: The "authority metric" of task planning can be adapted to rank potential agents during scheduling. The set of available agents can be ranked according to the authority held by the agent who wants the activity to be scheduled. Agents "obliged" to perform the activity may be considered preferable to those only "permitted" to perform it.
- Interaction Styles: Interaction styles could be modified by examining the relative authorities between two agents. If during an activity delegation, the receiving agent is "obliged" to perform the activity then a workflow system could offer a different communication style to the case where the agent is "permitted" to perform the activity.

Further Authority Modelling Issues

To enable a clear introduction, the description of our authority language so far has not described the relative "viewpoints" from which authority can be specified. Authority can be specified and viewed from either the viewpoint of the agent providing a capability or that of an agent requesting a capability. Considering the example in Figure 6, the secretary's authority is specified from the "provisions" viewpoint as the authority statements are attached to each of the capabilities that the secretary provides. Considering the example in Figure 7, the manager's authority is specified in terms of the capabilities that he or she can request and is therefore specified from the "requesting" viewpoint. Both the "provisions" and "requesting" cases are simply points from which a single authority model can be described and viewed. We anticipate that these perspectives will provide two useful viewpoints from which to present authority models within tools that support people in understanding, critiquing, and maintaining such models.

CONCLUSION

If workflow technology is to move forward into applications that are not based upon prescriptive process models, then techniques must be developed for supporting users in making decisions about the activities required to achieve a task and the agents that can perform those activities. In this paper, we have argued that knowledge about technical capabilities and the organisational structure and authority context within which they exist are important factors used by people when making such decisions. Our introduction to technical capability matching outlined how such a function can support important workflow decisions. Complementing this support with knowledge about an organisation's structure and authority context allows a workflow system to take organisational norms into account and to help users maintain these norms. The following points summarise how technical capability matching can support a number of workflow decisions and how this support can be enhanced with knowledge about organisational structure and authority:

- Task Initiation:
  Technical Only: ensure that users can only initiate tasks that they have the technical skills to manage.
  With Authority: ensure that only users with the appropriate authority start tasks. If a user does not have the authority to initiate a task, the workflow system can suggest a user who does so that initiation can be discussed.
- Task Planning:
  Technical Only: rank the methods available for achieving a task using the availability of agents that are technically qualified to perform the constituent activities.
  With Authority: rank the methods available for achieving a task using the authority and level of obligation of available agents that perform the constituent activities.
- Activity Scheduling:
  Technical Only: identify the agents that have the technical capability to perform an activity.
  With Authority: identify the agents that have the authority to perform an activity and rank them according to their level of obligation to perform that activity.
- Interaction Styles:
  Technical Only: no support.
  With Authority: modify the interaction style for dialogue between agents in accordance with the authority relationship between them.

We have noted how authority can be defined from both “capability” and “cultural” perspectives. The capability perspective lends itself to precise authority specifications that require significant modelling effort. The cultural perceptive lends itself to more general authority specifications that reduce the modelling effort with an (often acceptable) loss in precision.

We plan to implement the organisational and authority modelling languages outlined in this paper and experiment with them within an adaptive workflow system. This work should produce precise modelling languages and empirical evidence of their usefulness.

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.

We thank Stuart Aitken, Barbara Dellen, Sigrid Goldmann, Charles Petrie, and Austin Tate for enlightening discussions.

REFERENCES

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

[2] Dellen, B., Maurer, F., and Pews, G., "Knowledge-based Techniques to Increase the Flexibility of Workflow Management." *Data and Knowledge Engineering*, 1997.

---

[2] Details available at http://www.aiai.ed.ac.uk/~entprise/

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

[3] Fraser, J., and Tate, A., "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.

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

[5] Hoog, R., et al. "Organisation Model: Model Definition Document." University of Amsterdam Technical Report, KADSII/M6/UvA/041/3.0, Netherlands. 1994,

[6] Joosten, S., "Workflow Management Research Area Overview." *Proc. 2nd Americas Conference on Information Systems,* Phoenix, Arizona*, 1996.*

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

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

[9] Liu. K., and Dix, A., "Norm Governed Agents in CSCW." *Proc. 1st Int. Conference on Computational Semiotics*, Paris, France, 1997.

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

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

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

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

[14] Sheth, A., and Kochut. K., "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, 1997.

[15] Tate, A., "Authority Management - Coordination between Planning, Scheduling and Control." *Proc. Workshop on Knowledge-based Production Planning, Scheduling and Control*, held during the International Joint Conference on Artificial Intelligence (IJCAI-93), Chambery, France, 1993.

[16] Uschold, M., et al. "The Enterprise Ontology". *The Knowledge Engineering Review*, Vol. 13, M. Uschold and A. Tate (eds.), 1998.

[17] Uschold, M., "The Enterprise Ontology Ontolingua Encoding." [Available From] http://www-ksl-svc.stanford.edu:5915/FRAME-EDITOR/&sid=ANONYMOUS&user-id=ALIEN, 1998.

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