Making Workflows Context-aware: A Way to Support Knowledge-intensive Tasks

Mitra Heravizadeh **David Edmond**

Queensland University of Technology
Brisbane, Australia
Email: mitra.heravizadeh@gmail.com

Abstract

In business processes, knowledge-intensive tasks are ones in which the people performing such tasks are involved in a fair degree of uncertainty. These people are required to apply and bring together their experience, training, expertise and judgement. In particular, they are concerned about issues or problems that might arise and how these are best dealt with or avoided. Current workflow technology does not support such tasks, as it deals only with predictable and easily automated decision making. In particular, it fails to deliver the right information to the user at the right time based on the context of the process instance, thus not taking the opportunity to forewarn users of potential problems. Context-aware workflows are a way to overcome shortcomings of workflow management systems. This paper proposes an approach for the dynamic integration of knowledge and workflow processes by offering proper support for the realtime handling of the both the current context of a process and its execution path.

1 Introduction

A business process is an arrangement of business activities designed to achieve a certain business goal. Business processes can be highly structured or relatively unstructured (Hagen et al. 2005). Some processes behave in a way that is well understood, predictable and repeatable: the tasks are discrete and compartmentalised, the flow of control is straightforward, and any decisions involved in determining the flow of control are clear and simple. Some processes, on the other hand, involve tasks that are much more open, and require discretion and judgement. The interplay between these tasks is also much more fluid. Usually, such unstructured processes are not suited to automation. A business process that is able to be represented in an executable form is called a workflow model, or simply a workflow.

A workflow management system (WfMS) is a generic software tool which deals with the definition, execution, registration and control of workflows (van der Aalst 1998). An execution instance of a workflow model is called a case or process instance. Multiple cases of the same workflow model can run simultaneously without reference to each other.

A *task* refers to a single unit of work or activity, while a work *item* is a task that is being executed in a given case. In general a work item is executed by a resource that has been allocated to the task. Resources may be computer programs and they may be human beings: some tasks within a workflow can be performed by a computer without any human interaction, while other tasks may require people to form judgements or to make decisions. However, once a task is complete, and regardless of what kind of resource carried it out, the WfMS determines, by reference to current case data and the workflow model, how the process is to be continued. It does this by delivering the next piece of work to a resource capable of executing it (Reijers 2003). A task may trigger one or several more tasks when the work item corresponding to it completes.

When people are involved in performing tasks, they obviously need whatever knowledge is necessary to execute such tasks. Some part of this knowledge resides in the user's mind, and arises from his or her experience (tacit knowledge). Other forms of knowledge can come from external sources such as documents (explicit knowledge). Finally, some forms of knowledge come from contextual information which increases people's awareness of the situation.

A process can be said to be *knowledge intensive* if its value can only be created through the fulfilment of the knowledge requirements of the process participants (Hagen et al. 2005). While a knowledge intensive process might be structured, it is more likely to be unstructured: this could because of the unpredictable nature of tasks governed by creativity. It is also possible that, within a particular process, only some of the activities concerned would be classified as knowledge intensive.

In knowledge intensive processes, knowledge certainly contributes to the overall worth of the process. However, innovation and creativity also play a major role (Eppler et al. 1999). For example, where the process involves quoting on some software development, the task of approving an indicative project plan is a knowledge-intensive task. When a manager is required to approve such a plan, the decision to approve or reject the plan is based on both tacit knowledge (experience gained over years of being a product manager), explicit knowledge about the project (project requirements, estimates, budgets, resource history etc), and the organisational context such as the availability of staff and their level of experience.

It has been claimed that organisational knowledge management – the creation, transfer and application of organisational knowledge – has the potential to provide enormous benefits to business (Sullivan 1998, Alavi & Leidner 2001, Lichtenstein & Swatman 2003). Currently, workflow support for knowledge intensive tasks is not very sophisticated: the operation of a workflow relies heavily on the informal exchange of knowledge (Gronau et al. 2005). To operate properly, the workflow relies on personal or inter-personal knowledge management, which is certainly not formally integrated with the workflow. This separation of KM systems from everyday organisational work practice and business processes is of concern (Lichtenstein & Swatman 2003). The main reason for this concern is identified as a lack of context and personal intent in existing knowledge management repositories, inhibiting the usefulness of knowledge management systems. The integration of knowledge management and workflow systems is one of the goals of this research.

In most knowledge-intensive tasks, the user is required to make judgments or decisions. Knowledge management (Nonaka 1994) is useful when the right knowledge is distributed to the right person at the times needed. Therefore a workflow system should be fully aware of the context of the process to be able to provide appropriate knowledge at the right time to the right user who is working on a knowledge-intensive task.

A workflow system can assist the user who is working on a knowledge-intensive task to make a better judgements and decisions:

- It can forecast what *might* be expected downstream. Some events, happening at an early stage of the process, may raise unexpected difficulties downstream. In order to achieve an acceptable outcome, the user may want to take certain actions either to avoid or to rectify the issue. In this way, the WfMS is adapting its communication to the user based on an accumulated context of the current case.
- It can present, to the person facing an issue, knowledge gleaned from previous cases in which similar issues have been dealt with.
- It can propose appropriate solutions in case of an issue based on the context of the process instance.

The distribution of the knowledge to the user at the right time is another goal of this research.

For a workflow system to deliver all of the above, it should be fully aware of the current context of the process instance. To use context effectively, there must be a better understanding of what context means with respect to workflow systems, as there are a number of different definitions and uses of the term context (Maamar et al. 2005). A general definition of context is any information that can be used to characterise the situation of an entity, where an entity can be a person, a place, a physical or computational object (Lee & Helal 2003). The notion of 'context-aware business processes' has been discussed in (Rosemann et al. 2006). In their view, context is 'the relevant subset of the entire situation of a business process that requires a business process to adapt to potential changes in the context variables'. The authors have established that while most process modelling techniques are able to handle immediate context of a process in form data, applications and resources, a wider consideration of the contextual information is still not supported. The authors have proposed a framework for context which helps a better understanding of different types of context and their impact on business process, but no integration of the context reference framework with an existing process modelling technique has yet been proposed.

As most of the popular business process modelling techniques have little or no support for contextual information (Rosemann et al. 2006), the purpose of this research is to automatically integrate appropriate contextual knowledge into the tasks within a workflow. Accordingly, the research questions are:

- What are the contextual variables for each task within a workflow process?
- How these context variables are linked to knowledge within the process?
- How do different values for these variables impact the dynamic integration of the relevant information and knowledge to a task?

The paper is organised as follows: section 2 presents a motivating scenario involving a knowledge intensive process; section 3 introduces an approach to context-aware workflow systems; section 4 discusses that approach in some detail, and section 5 discusses how the rules that alert users are evaluated.

2 Motivating scenario

This section outlines a scenario that highlights the importance of introducing context-awareness within workflow systems to support knowledge-intensive tasks. The scenario is a relatively unstructured knowledge intensive process known as a *Request for Proposal*. An RFP is a tendering process used by both businesses and governments for the procurement of goods and services. In this scenario, we will assume that the company submitting the RFP is seeking the delivery of some software. The process commences with the receipt, by a software house, of an RFP from the client company. The RFP documents the high level requirements of the system and asks that software houses respond by providing a proposed solution, an estimate of the time and cost to complete the work and an indicative project plan. The particular RFP process we now describe is intended to highlight how context-awareness can improve the effectiveness of the process.

The process is outlined in figure 1. There, we have used the YAWL notation (van der Aalst & ter Hofstede 2005) to describe two of the main tasks involved: the top-level task and the task of developing an indicative project plan. The process is described entirely from the viewpoint of the software house. First, its marketing team receives the RFP and a review is performed to determine the parameters of the response. It may be determined during the RFP review that the software house is extremely keen to obtain the business as the prospective client is a large spender on software and success with this proposal could lead to more work in the future. It is decided during the review that the software house is prepared to do the business at cost.

One important piece of contextual information in this first activity is the *"the rating of the prospective client"*. It is important that this information is recorded and is made available for subsequent tasks, as those tasks may be performed by another team. For example, in this case, any project manager responsible for finalising the indicative project plan needs to be warned about this important contextual information in order to make appropriate adjustments to the project plan (perhaps to make the total project cost more attractive) to increase the chance of being awarded the work.

The next task in the process is to formulate some kind of solution which is then reviewed. Estimation is obviously a crucial part of responding to an RFP. Because of importance of estimation, if it is felt that the solution lacks detail or credibility, an attempt is made to gather more information, if time allows. If time is running out and the solution remains incomplete, the estimation will go ahead anyway. The formulation of the estimate takes as inputs the original request itself, the proposed solution, information about the proposed client, knowledge of past and current software projects and knowledge of previous RFP responses. The accuracy of the estimate is an important issue, and is determined by (i) *the method used* to carry out the estimation, (ii) *the experience* of the person making the estimate, and (iii) whatever knowledge of *the client, its project requirements and the proposed solution* is made available to that person.

The *"accuracy of estimate"* is an issue that a project manager needs to deal with before finalising an indicative project plan. Therefore, to be really useful the workflow system should warn the project manager what it knows about factors that may affect the accuracy of the estimate.

From the scenario, it can be seen that various kinds of contextual information may help a person working on a knowledge-intensive task to better understand the current situation before making decisions. What is required is a workflow system that can warn the user of possible issues based on given context and also provide an appropriate solution accordingly.

3 Context-aware workflows

In this research, we would like to use contextual information to bring to bear, to each stage of the execution of a given process, the knowledge that most appropriate to that stage. Workflow technology, which is the technology devoted to the automation of business processes, currently fails to exploit the possibility. We believe that, for a system to be context aware, there must be a model that provides the means for categorising, naming, storing, retrieving, reasoning with, and the binding of contextual information to tasks. We call this the **context model**.

3.1 Context categories

The categorisation of contextual information is an important stage in the design process, because each category is parameterised by a specific set of characteristics. At the most basic level:

1. *Resource-oriented* contextual information is concerned with whatever resource or resources are involved in performing a given task. This might include the user's role, job title, and length of service. For example the *"the experience level of the estimator"* in the RFP process is contextual data in this category. However, information in this category might be extended to include the availability of potential resources at the time the task became available for execution, and the unit cost of each such resource.
2. *Method-oriented* contextual information is concerned with the *way* a task is being executed, and the *time taken* to perform the task. For example, the *"estimate method"* – the technique by which the *"estimation"* task was carried out within the RFP process – is contextual information of this kind.
3. *Environment-oriented* contextual information describes the conditions that applied outside the process at the time a task was being carried out. This will include such information such as the location or locations at which the task was performed, the date or dates involved, and the time of day. This category will also include the specific business environment within which the owner of the process operates. Thus, the *"rating of the prospective client"* is an example in this category.

Each of the above categories provide *task related contextual information*, which arises at the time of execution of a particular task. On top of these basic forms, we have *execution path contextual information*, which arises from the particular path that the execution of the current case has taken (out of all the possible paths) prior to the execution of task. Such information allows a task to respond to potential difficulties. For example the *"is requirement detailed"* is context data of this type.

3.2 The context relevance space

The relevance space represents task dependencies in a workflow model based on the context.

Figure 2 shows an example of context sensitive task dependencies. Four consecutive tasks (Tasks 1 to 4) are involved in the process. Context variables are labelled $CV_{i.j}$ which indicates that it is the jth context variable of *Task i*. Issues are labelled $issue_{i.j}$ which indicates that it is the jth issue of *Task i*. Issues are linked to the tasked via a solid line, while dotted lines represent an issue's dependency on context variables and other issues. The thick solid arrowed lines depict a task dependency which has been derived from issue dependencies. The line points to the source of dependency. For example *Task 3* depends on *Task 2*, and this has been derived from the fact that *Issue 3.1*, which is an issue related to *Task 3*, has dependency on context attribute *CV2.1* which belongs to *Task 2*. The Ⓐ symbol represents the *and* operator. For example, *issue 3.1* depends on context attribute *CV2.1 and* context attribute *CV3.2*.

As shown in figure 2, a task may depend on a preceding task because (i) its issue depends on a context attribute of the preceding task (ii) its issue depends on an issue linked to the preceding task. A task may be depend on more than one preceding task. This pattern of dependencies is called the **context relevance space**, and it should be identified and explored, essentially, by experts in the application domain, with the assistance of the business process modeller. The actions to be taken are as follows:

A1: For each clearly identifiable task within a process model, establish all the possible issues that may arise. These are difficulties

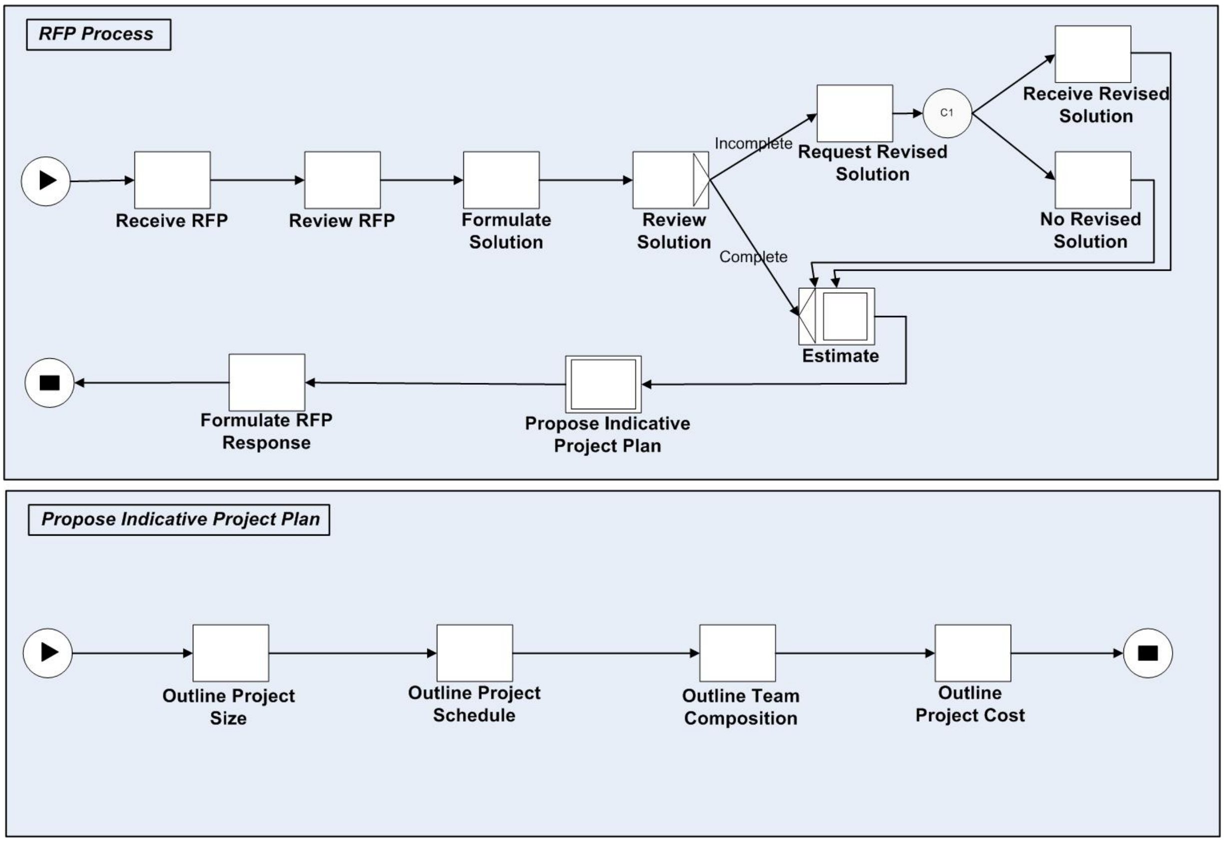

Figure 1: RFP process

that domain experts can forecast, based on their particular knowledge of the area.

A2: For each of these issues, identify those context attributes which might help decide whether the issue is sufficiently real to be concerned about. This may be facilitated by considering all the categories of contextual information discussed above.

A3: For each context attribute define important properties. The importance of these properties lies in their precise description of how each context attribute is to be observed at execution time.

A4: Establish conditions by which a context attribute may be judged to be at a critical level. For example, in the RFP process, we might express the following condition:

(?user experienceLevel is low)

This would help build a case that, for example, the accuracy of any estimates made by this user should be treated with caution.

A5: Express any rules over the possible values of the context attributes (at instance level). For example, we might express the following deduction:

(?user experienceLevel is low) $\wedge$ *(?estimateMethod is highLevel)* $\rightarrow$ *(inaccurateEstimate)*

This would allow us to determine, with some genuine confidence, whether the accuracy of the estimate is in question. If a rule is successful, then a case has been made that an issue needs attention.

A6: A set of possible solutions (that is, ways of rectifying the issue). This allows the process designer to refine the process model to include corrective decisions and tasks. In this way, the original model becomes more defined and able to be automated.

In the following section, we provide a more detailed discussion of the various activities just introduced.

4 Design activities

A1: Identifying issues

Identifying all the issues related to a task is crucial part of developing a context-aware workflow. Knowledge of identified issues and how to handle these issues will help a user to make an informed decision or judgment while working on a knowledge-intensive task.

This research offers a new perspective on the use of value-focused thinking framework in identifying task related issues.

The role of organisational objectives as the first step of business decision modelling using value-focused methodology has been well established (Keeney 1992). According to Keeney (1992), an objective is "a statement of something that one desires to achieves" and values are "what we care about". According to Keeney's model of decision making, value-focused thinking always leads to smarter choices. In many situations, by identifying and structuring values first decision makers are forced

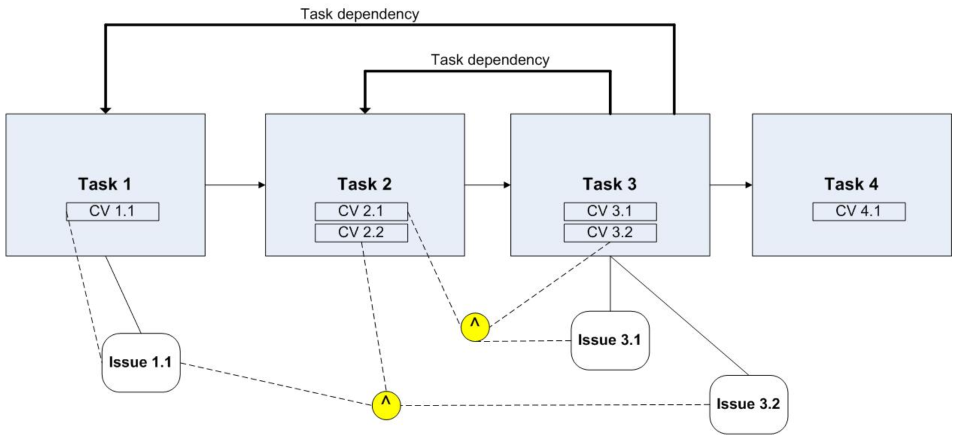

Figure 2: Relevance space – task dependencies

to more clearly define the problem. These people cannot easily identify and clarify values, however, without an adequate understanding of the problem. The process of defining the problem and identifying values is usually iterative. Keeney argues that value-focused thinking expands options and improves the likelihood of selecting the optimal outcome because decision makers can use the values as guides for alternative generation and evaluation.

The VFT framework was intended to be a general framework that can be applied to any decision situation (Heravizadeh & Edmond 2003). VFT has been used (Neiger & Churilov 2003) as a framework to facilitate goal-oriented business process modelling. In their framework the objectives have been used as a link between two disciplines of business process modelling and decision science. The VFT framework has been used to guide the design of business process models and to structure process objectives to ensure that the broader business objectives and values are expressed in the integrated model.

In this research, a VFT framework is used to identify issues. As part of process model design, an expert in the business domain needs to define an objectives hierarchy for each task. The hierarchy consists of objectives (each objective may have sub-objectives), values, and issues. Objectives represent the purpose that a task is intended to achieve. Because a task is typically a unit of work or activity, there might well be just one objective for a task. Values are derived from objectives and are complex way of thinking about the world and importance of things with respect to the objectives. Figure 3 depicts an objective hierarchy for the RFP *"estimation"* task. After values are identified, it is easy to identify issues that are linked to values, as each issue is derived by simply negating a particular value. Considering Keeney's definition of values – "what we care about" – the values in the objectives hierarchy defined for a task are the ultimate desired outcome of the task. For example for *"estimation"*, one of the desired outcomes is an *"accurate estimate"*. Therefore, it is a safe to decide that the prospect of an *"inaccurate estimate"* is an issue with respect to that particular value.

The values linked to an objective are weighted using the same scale, which in this example will be from 0.1 to 1, such that the sum of weights in a branch across a tier must sum to one. Issues inherit the weight from values: because if you value something highly, for example, then you would also be highly concerned about it not happening

A2: Identifying context attributes for an issue

Having now identified issues, the next step is to identify the context attributes that will allow the software to decide, for any active case, whether an issue is of significance.

Measuring the achievements of the values and objectives and developing a value model using these values is an important step in the VFT framework. The degree to which an objective is achieved is measured by what Keeney refers to as an **attribute**. For example, the attributes chosen to measure the *"accurate estimate"* are the *"experience level of estimator"*, the *"estimation method"*, and whether *"requirements are detailed "*. The selection of attributes to help measure a given value is, in itself, a decision problem facing the modeller.

The attribute should be measurable, operational and understandable. An attribute that is measurable defines the associated values in more detail by embodying implicit value judgements that are appropriate and avoiding those that are inappropriate (*"issues"* in this research).

All the attributes linked to a particular value are weighted consistently. The relative ratio beside each attribute (figure 4) indicates the level of the contribution of each attribute towards the linked value. The weights assigned to attributes which are linked to a value are not in any way dependent on the objectives to which that particular value may be linked. For example, the attribute *"experience level of estimator"* is weighted 0.3 with respect to the value *"accurate estimate"*. This weighting (0.3) remains the same no matter to which objective the value *"accurate estimate"* is linked. In the other word values are measured independently of the objectives.

The relationships between objectives and values in the objective hierarchy are modelled using an ORM Halpin (2001) schema (figure 4). In particular:

- An *objective* may be influenced by several *values*. Conversely, a value may be a factor in the achievement of a number of different objectives. Every objective is tempered by at least one value (plays a mandatory role, in the terminology of ORM).
- An objective may be achieved by the corresponding achievement of a number of separate (sub-) objectives.

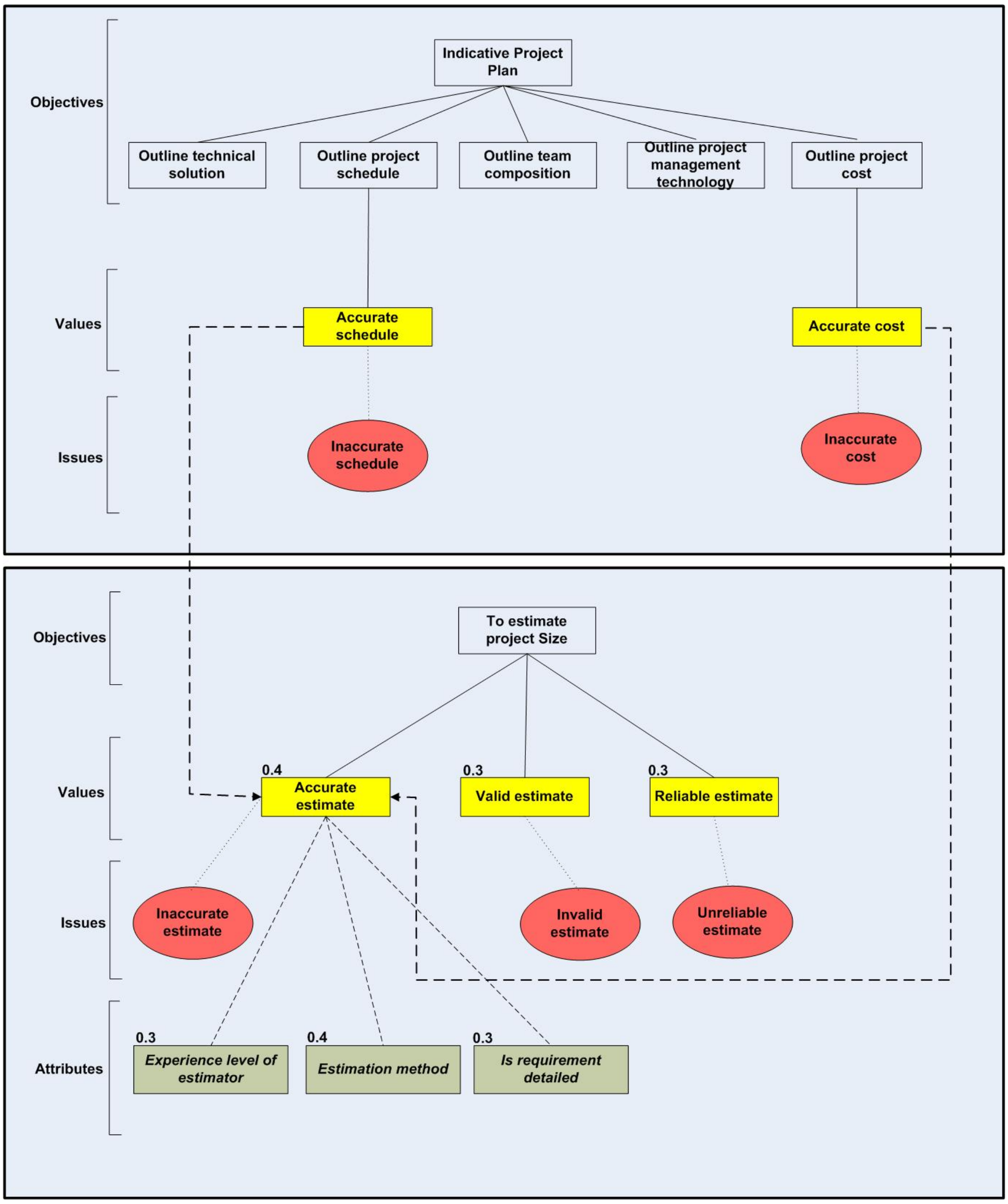

Figure 3: The value model hierarchy

- An *issue* is simply the converse (negation of) one specific value.
- A value may be measured by the conjunction of several *context attributes*.

A3: Defining context attributes

Each context attribute has set range of features that need to be defined by the modeller at design time. These properties are:

- The attribute's source task: the task from which the context attribute originates. For example *"estimation method"* is a context attribute belong to the *"estimation"* task.
- The attribute's category (whether the attribute arises from the method by which the task is carried out, or from the resource that carries out the task, or from some environmental factor applying at the time of execution).
- An appropriate name for the context attribute.
- A collection method: the means by which a value is acquired for the attribute (e.g., by user input or by a web service call).
- A storage method: the means by which the attribute is stored within the workflow system.
- Allowable values: a set of valid values for the attribute.

The importance of these properties lies in their precise description of how each context attribute is to be observed at execution time.

A4: Establishing conditions over context attributes

Now that we have a better understanding of (i) our objectives, values and issues, and (ii) the attributes by which we judge it possible to sense these matters,

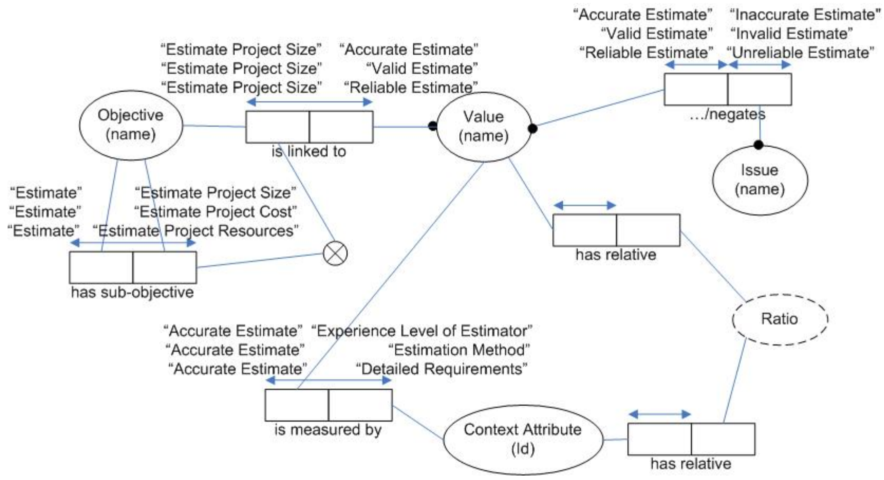

Figure 4: Objectives, values and issues schema

Valuing the Accuracy of Estimation				
Value	**Attribute**	**Acceptable minimum**	**Acceptable maximum**	**Relative ratio**
Accurate estimate	experience level of estimator	medium	high	0.4
Accurate estimate	estimation method	low level	medium level	0.3
Accurate estimate	is requirement detailed	yes	yes	0.3

Table 1: Value model – values, attributes and acceptable limits

we are in a position to decide exactly which observations would cause us to take action. This is the last step in defining the value model using the VFT framework. According to Keeney, a value model is a model with qualitative and quantitative relationships. The keys to building a value model are the set of objectives, values, and attributes to measure values. The data needed for parameterising a value model should be gathered from people knowledgeable about the domain.

Table 1 shows a value model for the example depicted in figure 4. For each attribute which measures *"accurate estimate"* value, a minimum acceptable level and a maximum desirable level have been defined. As an example, for the *"experience level of estimator"* attribute to contribute to achieving the *"accurate estimate"* value, a minimum acceptable level of *"medium"* and a maximum desirable level of *"high"* is required. Anything out of the acceptable range will raise the issue linked to that value.

A5: Reasoning with respect to an issue

Having now defined the value model, the next step is to define the rules over possible values of the context attributes. Considering the value model depicted in table 1, the following deduction can be made:

(?user experienceLevel low)) $\wedge$ *(?estimateMethod is highLevel)* $\wedge$ *(? not isRequirementDetailed)* $\rightarrow$ *(inaccurateEstimate)*

A detailed discussion of this aspect of the model is provided in the following section.

5 Rule evaluation

The purpose of this work is to enhance workflow management systems by providing support for knowledge-intensive tasks. This is achieved by providing valuable and relevant warnings to the user at the right time based on the contextual information captured through the process. Knowledge of identified issues and how to handle these issues will help a user to make an informed decision or judgment while working on a knowledge-intensive task.

As soon as a user starts working on a task, he or she needs to be notified of any issues raised either by preceding tasks or through the execution of certain paths. Therefore, an evaluation is required, as soon as a task is initiated (pre-evaluation). Another evaluation is required whenever a task is finished (post-evaluation). The post-evaluation identifies issues raised from the task just executed. Any issue identified during the post-evaluation will be posted to a form of noticeboard (for ensuing tasks).

Rule evaluation plans (pre & post) will automatically be generated after the value model has been fully defined for a task. A *pre rule evaluation plan* will be generated if (i) there is a task dependency on preceding tasks (either through issues or context attributes), or (ii) there is an issue which needs being evaluated by an execution path context attribute. A *post rule evaluation plan* will be generated if there is an issue which needs to be evaluated by context attributes related to the task being executed.

As depicted in figure 3, *"inaccurate estimate"* can be determined by evaluating values of context at-

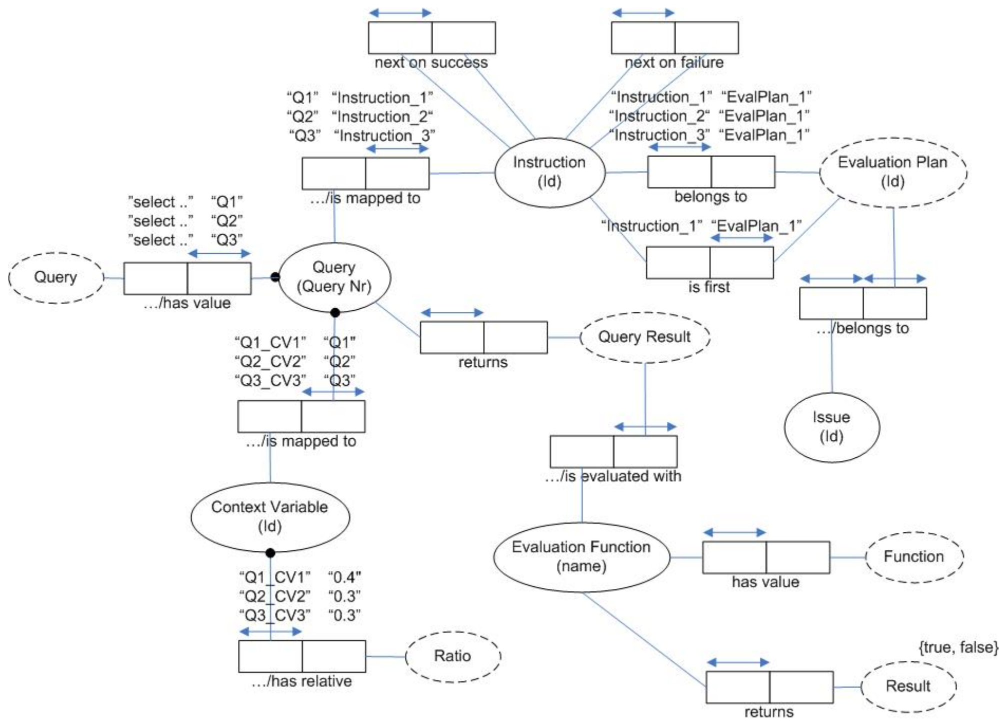

Figure 5: Instruction schema

tributes *"experience level of estimator"*, *"estimation method"*, and *"is requirement detailed"*. Based on theory above, two evaluation plans will be generated from this value model:

- A pre rule evaluation plan needs to be generated because the context attribute *"is requirement detailed"* is an execution path context attribute.
- A post rule evaluation plan needs to be generated to evaluate the issue after task being executed.

A rule evaluation plan, in general, is a set of instructions for evaluating either (i) context attributes, or (ii) secondary issues linked to a major issue. Each clause in *(?user experienceLevel is low)* ∧ *(?estimateMethod is highLevel)* ∧ *(?isrequirementDetailed)* → *(inaccurateEstimate)* will be evaluated by one instruction.

Instructions have several parts, as depicted in the schema (figure 5). The purpose of an instruction is to describe which context attribute or issue from preceding task a value should be collected for. Each instruction contains a query which retrieves either value for a context attribute or issue. The *query result* is evaluated using the linked *evaluation function* identified by name. If the evaluation returns true, the instruction identified by *next on success* is selected for processing and the query result is stored. If the evaluation fails, the instruction identified by *next on failure* is selected for processing and its *query result* is stored. The intuition behind having only two choices of which instruction to select next (next on success and next on failure) is that the query associated with a given instruction only ever results in either a *true* or a *false* outcome by the relevant evaluation function:

- In the case of a *false* result, the plan can proceed to the next instruction, if there is a further context attribute linked to the *issue*. Otherwise the plan has reached its conclusion *FinishedFalseIssue*. This means the evaluation plan has concluded that the relevant issue does not apply in current instance.
- In the case of a *true* result, there are two alternatives: either (i) the evaluation of the context attribute is enough to allow the conclusion that the issue exists in the current instance (*FinishedTrueIssue*); or (ii) the plan proceeds to the next instruction to evaluate the next context attribute.

The relative ratio beside each attribute plays an important role to determine either way. If the relative ratio of the context attribute being evaluated is more than 0.5 determines the evaluation of this variable is enough to conclude the issue exists in the current instance. Otherwise more context attributes need to be evaluated.

A tree view of the evaluation plan for the *"inaccurate estimate"* issue is shown in figure 6. Each numbered box represents an instruction: instruction 1 is to evaluate the context attribute *"experience level of estimator"*; instruction 2 is to evaluate the attribute *"estimation method"*; and instruction 3 is to evaluate *"is requirement detailed"*. The instruction on next on success (s) and next on failure (f) is listed in each box representing an instruction. The box marked as FTI represent the conclusion *FinishedTrueIssue*, while FFI represents the conclusion *FinishedFalseIssue*.

6 Conclusion

This paper was motivated by limitations of the support currently provided, by workflow systems, for processes involving knowledge-intensive tasks. These kinds of tasks require people to marshall and apply their experience, training, expertise and judgement in situations of some uncertainty. In particular, these people are concerned about issues or problems that might arise and how these issues are best dealt with or avoided. Current workflow technology does not support such tasks, as it deals only with predictable

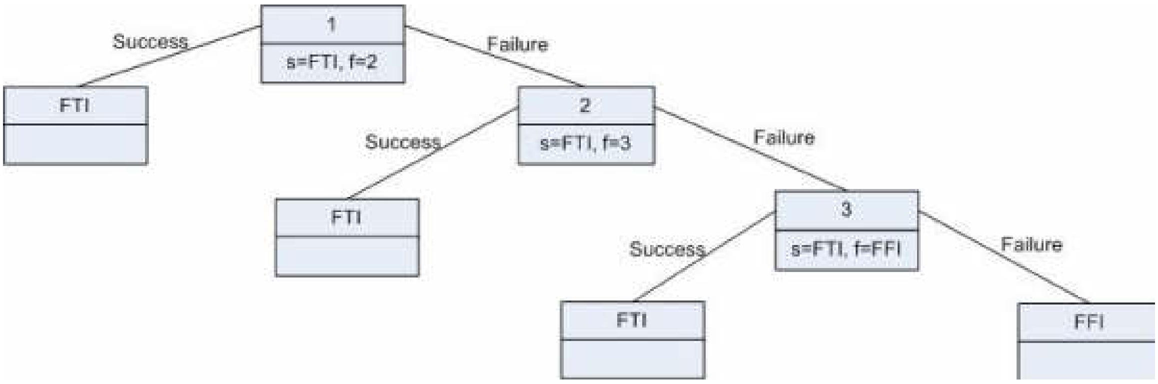

Figure 6: A tree representation of the "inaccurate estimate" evaluation plan

and easily automated decision making. But, we are not seeking to replace people. Instead, we are seeking to to deliver, to them, the right information at the right time: such things as who performed certain previous tasks, what experience and skills these people brought to these tasks, the way the tasks were performed, when and where they were performed, and so on. We believe that, by being context-aware, by noting this information as the process proceeds, a system will have the ability to forewarn users of potential problems. Context-aware workflows have been introduced as part of the solution:

- The notion that, for people involved in knowledge-intensive tasks, the focus is on *issues*. Issues are problems that may require their attention.
- Issues are closely linked to the idea of *values*. Values are what "we care about". Issues are what we are concerned about.
- *Objectives* are what we strive to achieve: their successful achievement may be tempered or hampered by our concerns over issues.
- The notion of a *context attribute* is introduced as a concrete way of recognising when an issue is significant.
- Three categories of such attributes have been identified within workflow systems.
- The *context relevance space* has been proposed as a framework to support context-aware workflows.

Context-aware workflows are a way to overcome shortcomings of workflow management systems. By proposing proper support for the realtime handling of the both the current context of a process and its execution path, this paper offers an approach for the dynamic integration of knowledge and workflow processes . Further work will include formalising models for the context relevance space and the annotation of tasks with contextual information.

References

Alavi, M. & Leidner, D. E. (2001), 'Review: knowledge management and knowledge management systems: conceptual foundation and research issues', *Management Information Systems Quarterly* **25**(1), 107–136.

Eppler, M. J., Seifried, P. M. & Ropnack, A. (1999), Improving knowledge intensive processes through an enterprise knowledge medium, *in* 'Proceedings of the ACM SIGCPR Conference on Computer Personnel Research', ACM Press, pp. 222–230.

Gronau, N., Muller, C. & Korf, R. (2005), 'KMDL – capturing, analysing and improving knowledge-intensive business processes', *Journal of Universal Computer Science* **11**(4), 452–472.

Hagen, C. R., Ratz, D. & Povalej, R. (2005), 'Towards self-organizing knowledge intensive processes', *Universal Knowledge Management* **0**(2), 148–169.

Halpin, T. (2001), *Information Modeling and Relational Databases*, Morgan Kaufmann.

Heravizadeh, M. & Edmond, D. (2003), On the role of value-focused interfaces for online shopping, *in* 'Proceedings of PACIS, Adelaide, Australia'.

Keeney, R. L. (1992), *Value-Focused Thinking: A Path to Creative Decisionmaking*, Harvard University Press, Cambridge MA.

Lee, C. & Helal, S. (2003), Context attributes: An approach to enable context-awareness for service discovery, *in* 'Proceedings of the Symposium on Applications and the Internet (SAINT)', pp. 22–30.

Lichtenstein, S. & Swatman, P. (2003), Sustainable knowledge management systems: Integration, personalisation and contextualisation, new paradigms in organizations, markets and society, *in* 'Proceedings of the 11th European Conference on Information Systems', pp. 1–8.

Maamar, Z., Benslimane, D., Ghedira, C. & Mrissa, M. (2005), On tracking personalized web services using views, *in* 'Proceedings of the 2005 IEEE International Conference on e-Technology, e-Commerce and e-Service', pp. 432– 437.

Neiger, D. & Churilov, L. (2003), Structuring business objectives: A business process modeling perspective, *in* 'Proceedings of BPM 2003, Eindhoven, The Netherlands', Vol. 2678 of *Lecture Notes in Computer Science*, Springer.

Nonaka, I. (1994), 'A dynamic theory of organizational knowledge creation', *Organization Science* **5**, 14–37.

Reijers, H. A. (2003), *Design and Control of Workflow Processes: Business Process Management for the Service Industry*, Springer.

Rosemann, M., Recker, J., Flender, C. & Ansell, P. (2006), Understanding context-awareness in business process design, *in* 'Proceedings 17th Australasian Conference on Information Systems, Adelaide, Australia'.

Sullivan, P. H. (1998), *Profiting from Intellectual Capital: Extracting Value from Innovation*, John Wiley & Sons.

van der Aalst, W. (1998), 'The application of Petri Nets to workflow management', *The Journal of Circuits, Systems and Computers* **8**(1), 21–66.

van der Aalst, W. & ter Hofstede, A. (2005), 'YAWL: Yet Another Workflow Language', *Information Systems* **30**(4), 245–275.