Towards an Integrated Life-Cycle for Business Process Management based on Learning and Planning

Hugo Ferreira[a], Diogo Ferreira[b]

[a]INESC Porto, Campus da FEUP, Rua Dr. Roberto Frias 378, 4200-465 Porto, Portugal
[b]Faculty of Engineering U. P., Rua Dr. Roberto Frias, 4200-465 Porto, Portugal
{hmf@inescporto.pt, drf@fe.up.pt}

Abstract

While focusing on the automation of business processes, Business Process Management (BPM) has process modelling as one of its fundamental premises. This is both a requisite for automation and the source of several problems, as workflow systems face challenges such as model completeness, exception handling, and dynamic changes. A survey on current research suggests that most of these issues have their roots on the need to provide process management with the ability to react to dynamic changes by planning, and to adapt to unstructured processes by learning. This paper explains how the continuous interplay between learning and planning will allow BPM systems to automatically adapt to new tasks and re-plan current processes, effectively providing a fully integrated life-cycle for business process management. Before that can be achieved, however, a set of fundamental problems needs to be overcome.

Keywords: Business Process Management (BPM), Workflow Systems, Machine Learning, Planning, Inductive Logic Programming (ILP)

1 Introduction

Business Process Management (BPM) allows for the automation of complex human-oriented workflows. Its ultimate goal is to greatly facilitate the coordination of tasks that humans and machines alike need to accomplish in the work environment. Over the years many researchers have worked on and proposed solutions to workflow management issues. The problems that have kept this research community busy for so long are well known and have been aptly described for example in [Alonso et al 1997] and [Alonso and Mohan 1997]. Non-trivial issues such as modelling language, availability, scalability, robustness and dealing with heterogeneity have been enumerated. In an effort to solve these issues, work in this area includes topics such as decentralization, transaction process monitoring, extended exception handling, rollback, compensations and alternate execution [Alonso et al 1997, Alonso and Mohan 1997].

More recently, research has focused on enhanced flexibility (e.g. by taking advantage of the interplay between agent-based and workflow systems [Huhns and Singh 1998]), modelling (automated consistency checking, automated deployment and execution [Chen-Burger 2001]), knowledge management (ontologies, proactive knowledge, etc. [Decker et al 1996]), and agent-based (or agent-enhanced) systems [Baral and Lobo 1997].

Most of these issues, as will be argued, have its roots in the need to provide process management with the ability to react to dynamic changes by planning, to adapt to unstructured processes by learning, and to make use of a single formal modelling technique in both. Once these capabilities are in place, it will be possible to devise a fully integrated life-cycle for business process management based on a continuous interplay between learning and planning – a system that learns new tasks and re-plans current processes.

This article is organized as follows: Section 2 reviews the traditional classification of workflow processes, while section 3 identifies the main topics of concern in workflow research. Section 4 then elaborates on the requirements arising from those concerns and on the relevance of planning and learning, and section 5 presents a broader view on how concepts from artificial intelligence can relate to business process management. In particular, it describes how learning and planning are the key enabling factors for an integrated life-cycle in business process management. However, as section 6 will show, there are some major obstacles in applying those AI techniques to this end.

2 Process Types

[Alonso et al 1997] classifies workflow processes into four types: *administrative*, *production*, *collaborative* and *ad-hoc*. These process types are ordered according to: business value, process uniqueness, task complexity and task structure. This classification reflected the set of research issues in the workflow area at that time. Some of those issues – such as fault tolerance [Alonso et al 2000] – have already been thoroughly discussed, whereas other issues such as ad-hoc processes still remain to be tackled [Voorhoeve and Aalst, 1997].

Workflow research has focused fundamentally on modelling processes and enhancing the run-time behaviour of these systems, while resorting to different modelling languages and formalizations [Aalst 2000, IFIP–IFAC 1999, WfMC 2002, Adams et. al. 2003]. Process modelling concerns the relationship between activities and flow control, while the run-time behaviour is more focused on the activity itself. From this we could argue that workflow research can be seen along two dimensions: *process complexity* and *task complexity*. Process complexity deals mainly with flow control (verification), exception handling (consistency) and resource allocation (optimization) amongst other issues. Task complexity on the other hand is closely related to domain knowledge. It is specifically concerned with topics such as knowledge management (controlled vocabularies, dictionaries and ontologies) and information integration (heterogeneous data sources, conversion, consistency).

We can use these two metrics to classify the four types of processes into two main classes: *structured* and *unstructured* processes (see figure 1).

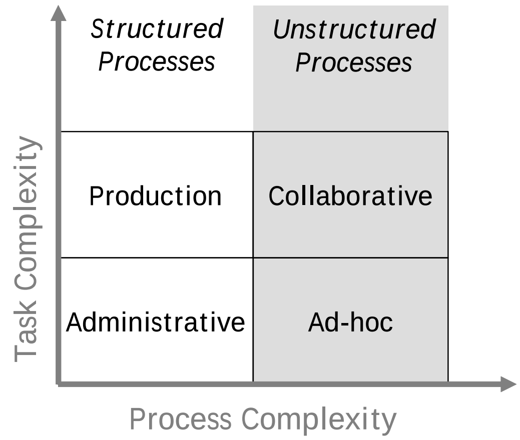

Figure 1. Complexity-based classification of processes.

This article will focus on the unstructured type of processes, with an aim towards supporting complex processes such as collaborative workflow. It is important to note that the framework and issues discussed herein are applicable to structured processes as well.

Looking back, one can see that research has progressed from administrative and production type processes (structured) to ad-hoc and collaborative processes (unstructured processes). Investigation has naturally moved from basic flow control problems (structured) to the more difficult and richer dynamic environments of cooperation and negotiation (unstructured) [Jennings et al 2000]. However, the problems associated with these dynamic environments have been a major impediment to the application of business process management itself. There is also the tacit conviction that current process management techniques remain unsuitable for unstructured processes, since the research community still actively investigates these and other related issues. Comparing [Nutt 1996] with [Adams et al 2003] suggests that the challenge of supporting ad-hoc processes is still a present-day issue. In the next section we will identify a framework for future investigation based on a simple but essential set of problems that need to be dealt with. Hopefully this will provide new research paths and reinvigorate the investigation activity in this area in collaboration with other research communities.

3 Fundamental Issues

The first step in understanding what is at stake in BPM research is to decompose the subject into its fundamental themes. The following list enumerates these basic themes, most of which have already been investigated and reported on in some form or another.

Completeness: any process must be modelled before it is executed. To be able to fully automate a process the resulting model must be complete in the sense that it should reflect reality. Such completeness is not possible, even if we restrict ourselves to a very limited subset of the world. On the other hand, if we did have all of the required knowledge to come up with a complete model, the world we live in is ever changing and evolving and such a model would need to be continuously revised. Humans in general deal with this by being aware of changes in context, identifying patterns and regularities and using this information to learn.

Consistency: a process model must be consistent in that all the information and state it holds represents the current view of the world. To do this requires any and all changes in the environment to be used to change the model state itself. The modeling languages should also enable one to maintain (or at least verify) the models' consistency in an automatic fashion, e.g., by assuring that the model is sound [Aalst 1998] or that it has no deadlocks.

Flexibility: a modelling language must be general and expressive enough to describe any of the elements required for process enactment. This includes, but is not limited to: flow control, business rules, legal constraints, strategies, goals, queries, resource description, negotiation protocols, components and domain knowledge. It should also allow for easy incorporation of new knowledge, preferably in a way that does not require the process to be fully revised before being enacted again.

Process life-cycle: consists of two basic phases: build-time and run-time. During build-time a model that describes a process is created and deployed for execution. During run-time the process model is instantiated and this model is then used to manage process execution. Process models need to be altered and redeployed due to changes in goals, strategy and the environment. The objective here is to significantly reduce the turn-around time between modelling a process and its deployment.

Dealing with complex processes (adaptability): for a process to reflect reality in a usable fashion means that such model will be large and complex. Because these models will sooner or later be changed, its modelling language should therefore have a theoretic base that will enable one to freely change the model but ensure that it remain consistent.

Dealing with unforeseen tasks (exceptions): process management has automation as its main objective. Ideally, therefore, the process should be able to deal with exceptions without human intervention.

Optimisation: another goal of process management is to increase effectiveness and efficiency. Whatever the criteria for effectiveness and efficiency, process modelling should support basic constraints satisfaction and resource usage optimisation.

Knowledge management: unstructured processes include all those human-oriented process with complex, dynamic and creative behaviour. In such a scenario, humans executing the tasks dictate much of the flow control themselves because they hold the domain knowledge required to make the correct decision. In such circumstances, identifying and modelling flow control becomes unfeasible. The key ingredient here is domain knowledge. Such knowledge must be collected, managed and distributed amongst the users in a consistent manner.

The themes above provide the framework that will be used for future research. All these issues have been dealt with before in one way or another, but have never been presented together as a whole. We believe that these issues constitute the fundamental building blocks that one must be aware of before unstructured processes can be successful supported via automated management. As such, the next two sections present the basic tools and techniques that may be employed to partially solve most of the above issues.

4 Resolving the Issues

4.1 Formal Modelling and Automation

A closer look at the list of fundamental issues, one can see that modelling languages and the modelling phase itself plays a key role [Nutt 1996]. Much work in process management has been focusing on this and related issues [Aalst 1998, Chen-Burger 2001, Baral and Lobo 1997, Tang and Hwang 1998]. [Lin et al 2002] provides a good survey on process modelling and proposes a generic structure that encompasses all modelling perspectives (functional, behavioural, informational, organizational, verification/validation and modelling procedure) that are required for successful modelling. It is interesting to note that [Lin et al 2002] does not consider verification and validation as essential. It is a general belief, however, that this functionality is in fact essential for BPM systems [Sheth et al 1999].

The main goal of modelling and supporting processes is automation. The level of automation is directly dependent on the ease with which software may manipulate and process information. It is therefore imperative that the modelling languages and the resulting models themselves be amenable to automatic manipulation. To this end we have identified the following characteristics of the modelling language:

- **Possess formal description**: the language must have a solid theoretic base that will allow for automated manipulation according to a set of well-known axioms or rules. Examples of formal languages include propositional logic, first-order logic, descriptive logic, event calculus, situation calculus, Petri Nets and any Finite State Automata in general. Such languages deal with issues such as grammar, parsing, undecidability, intractability and complexity.

- **Allow inference**: many formal languages and models exist, each with its own set of symbols and operators. Theses elements enable us to pose a query, refute a fact or prove a conjecture.

- **Provide automated resolution/verification techniques**: the selected formalism should provide the means for automatic resolution. Two important characteristics of such a resolution are completeness (all true statements have a proof) and soundness (dual of completeness, i.e. if a fact can be proved it is true).

The use of formal languages and models, however, is complex and requires much training. Naturally we expect tools to provide user-friendly access to and manipulation of the process models. Such tools may include: graphical front-ends, wizards and on-line help systems.

4.2 Reacting By Planning

Process modelling deals with tasks (also referred to as activities or actions) and describes the static order in which these tasks are executed and the relationships between them. Such a model attempts to identify all possible actions, foresee all possible interactions amongst these actions and consider all conditions under which such actions may occur. However, as already pointed out, generating such a model may be quite difficult. In process modelling one usually resorts to simpler workable models and let the humans intervene when exceptions do occur.

In unstructured processes the set of possible tasks cannot be anticipated, let alone attempting to describe static relationships between them. This means that we must effectively deal with unexpected situations. As a corollary, we could argue that process modelling as done by most of today's modelling languages becomes unfeasible.

Automated planning (or scheduling) has been identified as a solution and some work has already been done in this area [Smith 1999]. Planning differs from process modelling in that it does not explicitly encode all possible flows. A plan represents a subset of what process management refers to as the process model - it represents the flow control only for a given world state. On the other hand, planning assumes that all action's effects are known. Any changes in the world state not foreseen in the operator description means the plan is invalid. Therefore, the effective use of planning requires techniques such as plan repair, re-planning and sensory planning [Howe 1995]. The advantage of such a system is that it would be inherently reactive and adaptable. It could effectively explore multiple sequences of actions and find new, improved solutions without the need for intensive repeated manual labour of process modelling and administration.

4.3 Discovering and Adapting By Learning

It is a well-known fact that the single greatest bottleneck in the deployment and use of automated process management is process modelling (build phase) [Aalst et al 2003, Chen-Burger 2002]. In response, research has taken two different approaches in dealing with this. The first attempts to facilitate modelling. This includes, but is not limited to: easy-to-use modelling languages (graphical diagrams, specialized views), automated checking of the completeness and consistency of multiple models [Chen-Burger 2001], extension made to alternate but well known languages to encompass process modelling [Penker and Eriksson 2000] and using higher order abstractions [Medina-Mora 1993, Aalst 1998].

The second approach is an attempt to automate modelling. The objective is to analyse how a process is executed and from such information identify the process [Klingemann et al 1999]. An example of such an approach is workflow mining [Aalst et al 2004]. Although workflow mining, as it has been presented in [Aalst et al 2004], requires process audit trails from a working automated process management system, it does present an interesting concept that can be used specifically to support unstructured processes.

In addition to the initial modelling phase, constant modifications to processes are required due to exceptions that occur during process execution or to environmental changes that require an immediate change in the process execution policy. Current solutions provide mechanisms for the detection and flagging of exceptions, which then requires manual intervention from human agents [Tang 1998].

From the previous statements it is evident that such changes require the repeated application of the lengthy modelling and deployment phases. For short-lived, dynamic unstructured (almost unique) processes, such a costly effort of remodelling is not justified. In this work we propose to use learning as a means to eventually bypass the modelling phase and significantly shorten the processes management life-cycle. In this manner, the system will be able to incorporate new knowledge and adapt itself to novel situations.

Ultimately we are aiming at *zero-administration process management*. The objective is to learn how to plan task sequences required in attaining a goal and thus be able to automatically generate (and change) processes.

4.4 Goal-Oriented Modelling

When humans execute a set of tasks, they do it with the intention of partially satisfying a goal. We can therefore say that a process is identified by a goal and that each task in its turn identifies the sub-goals that need to be accomplished in order to carry out the process successfully. It is important to note that activities may represent processes themselves so that the relation between goal and sub-goals is a hierarchical one.

The relationship between goal and action, however, is not a simple "one-to-one" relationship. The same action may be taken for a number of reasons. What is more, one may go about reaching the same goal with a different set of tasks. Unstructured process management will only be successful if we can identify, record and disseminate information about these goals in a consistent manner. Knowledge acquisition can only be made possible if people associate a given action to a specific goal that has a well-defined meaning. Modelling by goals thus enables users to establish a common ground of understanding when executing tasks. It also allows them to provide this information to the system so that learning and planning may be applied successfully.

It is important to note that workfow participants are responsible for identifying and using a set of goals having a well-understood and unambiguous meaning. To do so requires the use of a dictionary or ontology, but the use of such a dictionary or ontology should require little or no initiative from the user. For example, queries for plan generation via goals should automatically search thesauri for equivalent goal terms and ask the user if the definition recorded in the system is the intended one. If the user agrees then the system proceeds, otherwise the user is guided by the system to add to or correct the knowledge base. Such a knowledge base is intimately related to the domain knowledge and is greatly influenced by the expressiveness of the modelling language. Ultimately this will determine the success of planning and learning phases.

On the other hand, the relationship between a goal and a set of preconditions is critical. Goals are a means for a user to interface with the system (ease of use) while preconditions are the "rules" that determine the success of planning (automated inference). The user interacts with the system in three different ways:

- The first is as a standard workflow participant that selects activities to be executed. Such a (non-prescriptive) task is nothing more than one of the steps that are required to complete a plan.

- The second is when a user queries the system for a plan in order to reach a certain goal. That goal may be already described in the system, but the user may need either to add or to alter its definition. The planner will attempt to satisfy the preconditions for that goal and produce a plan, which is in fact a deployed workflow model.

- Lastly the user may be required to assist the system in identifying, defining or correcting an operator used by the planner. The purpose here is to support the definition of such operators in a semi-automated way via the use of a learning sub-system.

5 AI Techniques

In order to adequately support unstructured processes, process modelling must be automated to the point where constant change and instant deployment is supported. Our view of the future is that of a *fully automated process management life cycle* based on a continuous cycle of learning and planning. The fact that these techniques had its origin within the scope of *Artificial Intelligence* (AI) suggests that potential usefulness of AI techniques in the context of managing unstructured processes should be further investigated.

The use of AI in process management is not new. Some work in this area does exist [Tate et al 2000, Tate 2004, Lin 2002, Decker et al 1996, Myers and Berry, 1999]. However, AI techniques will be even more helpful when dealing with unstructured processes. In fact such potential has already been identified [Jarvis et al 1999], although far too little attention has been given to it. In the following subsections we will attempt to show that AI possesses the means to (partially) solve the various fundamental issues that were previously identified.

5.1 First Order Logic

One of the basic requirements we have identified was the need for a formal language with a theoretical base. Having learning and planning in mind, First Order Logic (FOL) is a good candidate since it possesses the following characteristics [Nilsson and Maluszyński 1995]:

- Automated resolution (based on logic);
- Allows for the use of inference (deduction, abduction, induction);
- Has high expressive power.

This set of characteristics allows us to effectively ensure *consistency*, have *flexibility* of expressiveness, and use it as a language for *knowledge management*. In other words FOL is appropriate for three of the eight basic issues that were listed.

5.2 Planning

Planning is a means of attaining reactiveness and adaptability. AI research in this area has progressed much in the last few years to the point where it may be practical to use it in solving real-world problems [Zimmerman and Kambhampati 2003]. Planning as a research topic, however, is extensive. Its techniques (based on state-state and plan-space planners) and algorithms (Constraints Solving Problem – CSP [Barták 2001], Satisfiability Problem – SAT [Marques-Silva 1998], Integer (Linear) Programming, Binary Decision Diagrams - BDD and heuristics) are numerous and include, just to name a few: mixed initiative planning [Cox 2003], planning by analogy, contingency planning, refinement planning, hierarchical task decomposition and learning in planning [Kumar 2002].

Today's advanced planning systems can handle dynamic and incomplete worlds [Veloso et al 1998]. Current development has taken a step further by attempting to handle time and resources. State-of-the-art planning aims to efficiently handle durative resource consuming actions. At this point planning is starting to incorporate scheduling [Kumar 2002]. This means that optimisation of resources (as opposed to simple plan quality metrics) in real-world problems may be a reality in the near future.

Interestingly enough, planning has been identified as a key technology that can be applied to workflow specifically [Kearney et al 2003]. In addition, planning has been used together with Grid computing and the Semantic Web to facilitate scientific data analysis by means of automatic generation of workflows [Gil et al 2004, Blythe et al 2004]. Although planning has yet to be used to facilitate the enactment of human-oriented processes, some research is now unveiling its potential in this respect [Madhusudan et al 2004].

5.3 Learning

In process management, the purpose of using learning techniques is to identify operators together with their pre- and post-conditions so that, ultimately, we may use planning to generate process models, as shown in figure 2. Because learning the operators cannot be fully automated, the user will also be able to modify existing operators or introduce new ones. The operators represent the actions that planning will organize into a new process model. Such a process model can be automatically deployed; then workflow participants will perform each of its activities. Learning will thus enable domain-specific knowledge to be disseminated amongst the users of the system (an operator identified by one user is used by another user) and support the evolution of domain-specific knowledge (adapting operators to new business rules, adding or removing operators).

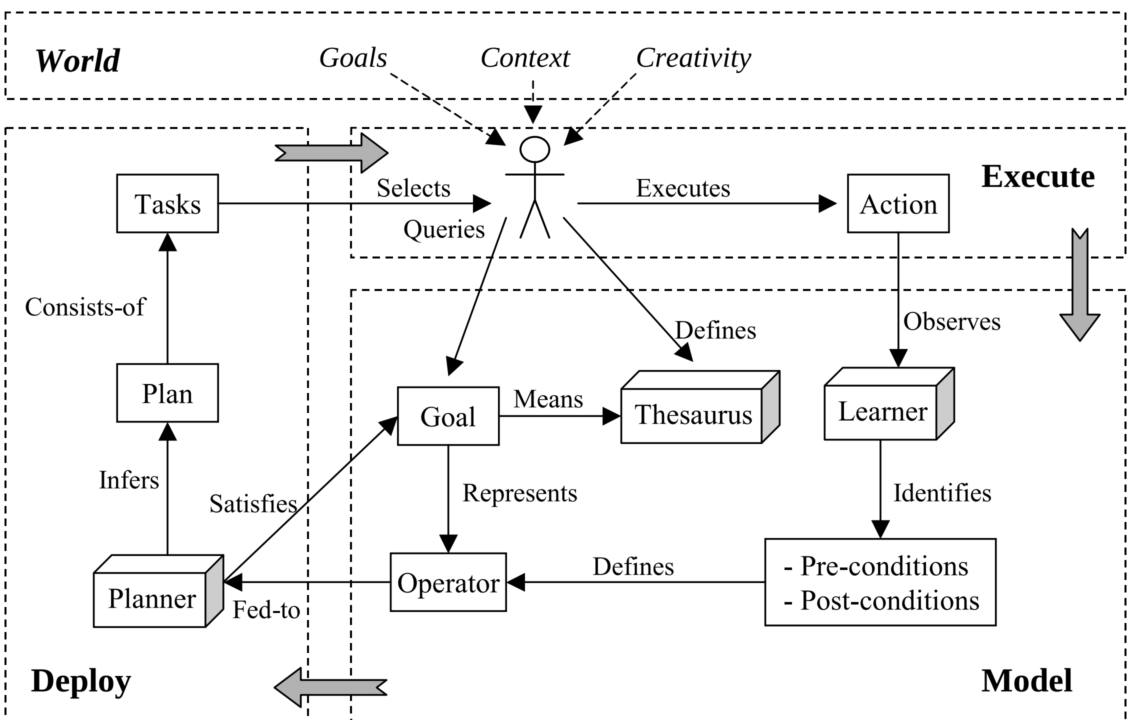

Figure 2. Learning and planning in the process management life cycle.

Planning uses STRIP-like operators [Russell and Norvig 1995] that describe actions (a more expressive language known as Planning Domain Definition Language or PDDL [Fox and Long

2003] is now in use, but we have adhered to the simpler STRIP operators in our experiments). Planning is the “simple” composition of these actions according to a set of rules. The set of techniques under the umbrella of knowledge learning are well suited for identifying operator rules. Explanation based learning (EBL) and inductive logic programming (ILP) [Muggleton and Raedt 1994, Lavrač and Džeroski 1994] are two means of obtaining the required planning rules.

ILP seems especially appropriate for two reasons (for more information on the advantages of using ILP as a learning technique please refer to [Bain 1994]). The first is that ILP generates FOL like expressions that can very easily be used to express the STRIPS operators. The second is that current ILP research is focusing on optimisation issues and can now handle larger and more difficult problems [Page and Srinivasan 2003]. Although much work needs to be done before any results may be presented, we have come to realize that learning is the most critical element in order to attain full automation of the process management life cycle.

There is already some work in applying learning to process management, although not specifically targeting unstructured processes, or using planning and learning at the same time. Research that is very closely related to learning (although not presented in the AI context) is workflow mining [Aalst et al 2003]. Some investigation in the use of machine learning techniques in process management can also be found and include [Herbst 2000].

5.4 Learning and Planning

Much work has been done in relation to learning and planning [Veloso et al 1995, Wang 1996, Tae et al 1999, Gopal 2000, Estlin 1998]. Initial investigation in the use of learning in the planning arena had two basic objectives: to increase plan quality (algorithmic effectiveness) and to decrease the time needed for planning (algorithmic efficiency) [Zimmerman and Kambhampati 2001]. Recent developments in planning, however, have made learning under these two settings unattractive. The time spent in learning rules that will speed up inference or result in better quality plans is counter productive (this is known as the utility problem) [Zimmerman and Kambhampati 2001].

In contrast, “learning to plan” has a different focus [Wang 1996]. It attempts to obtain information that will allow planning to proceed successfully. It has been seen as a difficult problem and has therefore not attracted a lot of attention [Tae et al 1999].

In general, it is possible to take action without planning – one may recognize by experience that a certain situation requires a given action and simply act. Taking action then becomes a problem of identifying policies and circumstances under which set of appropriate actions should be taken [Pearson 1996, Lent and Laird 2000, Könik and Laird 2004, Khardon 1999]. Research in cognition can be used to identify *when* a set of actions are usually taken and then use this information to implicitly identify the sequence of actions that *should* be taken as the situations arise. Unlike planning, such policies are static and do not allow for the reactiveness we are looking for. In contrast, however, learning policies becomes a computationally more tractable problem because one is only required to identify the minimal set of conditions that require an action to be taken as opposed to identifying *all* conditions under which an action can be taken.

Very little has been done in order to combine learning and planning for the management of unstructured processes. This may stem from the fact that AI research has not presented many results in "learning to plan" with the explicit goal of identifying planning operators. Nevertheless, the usefulness of combining these two techniques has already been identified [Moreno et al 2000].

6 Basic Hurdles

Any system that attempts to deal with real-world environments faces a number of basic pragmatic problems. In order to better illustrate these practical issues, we will refer to the sample claim-handling process shown in figure 3.

The process begins when the company receives a customer claim about a defective part. The claim is sent to the Quality Department for technical analysis, so as to ascertain whether it is plausible or not. Should the claim be accepted, the Quality Department may either send the defective part for repairing at the Production Department or, if it is not possible to repair it, the Quality Department asks the Sales Department to compensate the customer. Sales can decide either to ask Accounting to reimburse the customer or replace the defective part with a new one from the Warehouse. However, if the company is out of stock for that specific item, only the first option is applicable. In any case, the customer will be notified of the ultimate decision concerning the claim.

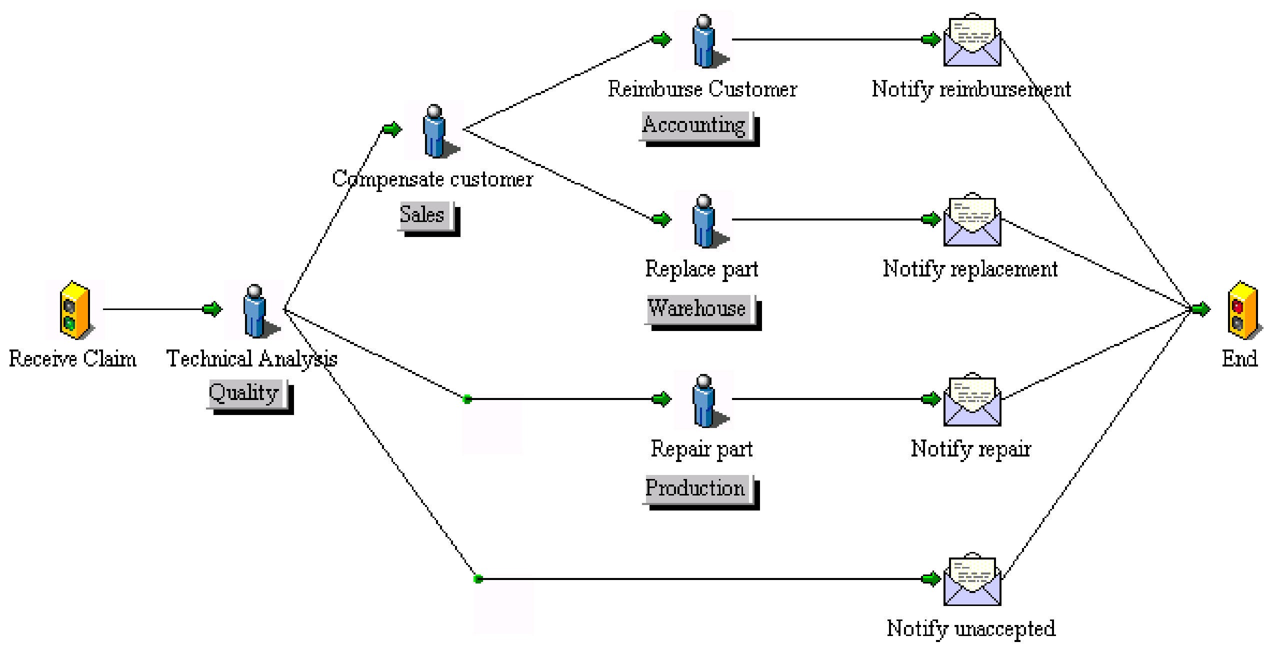

Figure 3. Customer claim processing model (Ultimus Process Designer)

The following sections list a number of elementary and highly interrelated issues that cannot be solved by simply resorting to sophisticated formal theory.

6.1 Observability

A fully automated BPM life-cycle requires an information system that will be able to pickup on every action of a user and learn from those actions. In human-oriented processes where many of the activities are not supported by information technology (IT), data collection presents a very difficult job. It becomes difficult to identify relevant from irrelevant actions, let alone collecting data from all applications being used.

Assume, for example, that Accounting is reimbursing the customer, according to the process shown in figure 3. How do we know that the employee at the Accounting Department has actually reimbursed the client? Is it because she opened a spreadsheet and changed a value under the column "Accounts Receivable"? Or was it the database update that she performed just five minutes ago?

As an initial approach we assume that the system can only observe the world model that is directly manipulated by the user. Additionally, we view this world model as sequence of discrete frames. Each frame represents a world state, in other words it is a snapshot of the real world at a given point in time. Any action taken requires the human actor to indicate the goal and the action taken, and to flag the start and finish of that action during execution, thereby enabling us to mark the frames in order to automatically learn the action's pre- and post-conditions.

6.2 Consistency

When users execute tasks using an information system, they are effectively using data that represent concepts. These concepts depict real objects that live under a set of constraints and restrictions determined by business rules. This is not obvious in the data models that represent them. Thus, people would be free to erroneously and unknowingly manipulate the information resulting in an invalid world state. Such a world state would deceive the learning and planning engines.

To deal with this issue, we must guarantee that the user may only change the world state in a consistent manner. As an example, suppose that the Warehouse replaces the defective part by a new one, according to the process shown in figure 3. If the employee retrieves parts from the Warehouse without updating the stock level for those items, the process will reach an inconsistent state; then Sales may decide to replace a defective part when that kind of item is actually not available in stock. To solve this problem, the employee must use an application that will correctly update stock level information whenever parts are stored or retrieved from the warehouse. With this approach, however, we now have the problem of generating and representing the restrictions that ensure consistent state manipulation.

6.3 Completeness

For a system to be adaptable it must be able to incorporate new information. This is effectively a open-loop system since the users' input may result in the change of process models, which in turn will influence the users' actions. Such a system suffers from a fundamental problem that all AI systems have to deal with: representation [Russell and Norvig 1995]. In other words, how can we best represent knowledge so that it is amenable to automated processing?

As an example, suppose that the process shown in figure 3 is actually unaware of the concept of "stock". In that case the model would be incomplete, since it would be lacking a precondition for the replacement of defective parts: the precondition is that a new part must be available in stock.

We will limit ourselves to extending the model that will allow for the incorporation of new planning operators only. We conveniently admit that expert users will model the essential background knowledge. However, graphical user interfaces (GUI) wizards will allow for the input of planning operators much like [Ai-Chang et al 2004] do it, only we will attempt to automate this as much as possible. Because the addition of background knowledge will require expert users it becomes evident that zero administration is not possible.

6.4 Qualification Problem

Qualification has to do with the problem of ensuring that an action is guaranteed to work [Russell and Norvig 1995]. The qualification problem arises when considering the following question: if I perform an action, need I not observe the results to confirm that it did in fact succeed?

As an example from the process shown in figure 3, suppose that the Warehouse is to replace the defective part by a new one, but due to a malfunction of the stacking system that occurs from time to time, the parts stored in the upper shelves of the Warehouse cannot be retrieved, so it is not possible to provide the new part. In that case, the model should take into account the precondition that the stacking system must be working properly (completeness) and there must be some way to check if that precondition is true or not (observability). The qualification problem may therefore be considered as a combination of the observability and completeness issues.

6.5 Ramification Problem

[Russell and Norvig 1995] describe the ramification problem in the "Wumpus world" as the dust on the gold brick: if one picks up the brick, then naturally the dust will also be picked up. As another example taken from the process shown in figure 3, suppose that Production is to repair the defective part. The part may be repaired and then delivered to the customer or, alternatively, the customer may have been already provided with a new part, and once the defective part is repaired it will be available for delivery to another customer. This means that a defective part that is turned into a good one becomes a new part, just like any other newly produced part. Therefore, repairing a defective part results in a new part, which, in turn, increases the stock level. This fact is a ramification and must be explicitly modelled.

As with any problem-solving exercise, it is necessary to enumerate all the axioms that represent the mechanics of the real world no matter how obvious they are. This is a difficult endeavour especially if the person that is providing the rules and facts of a specific domain is not well versed with inner workings of the inference engines. Initial research has pointed to the use of planning as an experiment to test the learned knowledge [Wang 1996, Tae et al 1999]. A failed plan can potentially be used to identify possible errors in planning operators and prompt the user for more information.

7 Conclusions

Much work has been done concerning unstructured processes. From these efforts we can see that researchers have approached this problem from different points of view and with various goals in mind. In an attempt to address this problem systematically, we have identified a set of fundamental issues to be studied so that unstructured processes can be successfully supported. This allowed us to decompose the problem and identify a set of candidate techniques to reach that objective.

Fully automated support of unstructured processes is a difficult problem to solve. The central issue is that of model representation, which cannot be automated for the reasons we have pointed out in section 6. As such, we have conveniently delegated this responsibility to an "expert modeller". The more attainable goal is to reduce this modelling effort and allow the system to adapt according to users' needs, thus providing a semi-automated management of unstructured process.

The key to supporting unstructured processes is a combination of learning and planning. By observing the users' actions and prompting them for information, we aim at successfully learning planning operators. These planning operators will then be used to find the sequence of steps (the process model) required to reach a given goal. The plans can also be used as means to disseminate domain-specific knowledge throughout the user community on how to reach specific goals. The system should evolve to a point were all process models may be automatically generated without intervention of the process modeller. At that point, we will have achieved a fully automated life-cycle for business process management.

8 Future Work

Learning is a critical feature in order to support unstructured processes, because it provides the acquisition of domain knowledge used in automated model construction (planning). In other words, it is an essential part of automating (or reducing the effort of) modelling, which is our main goal. Much of our effort will concentrate on this topic.

Previous work suggests that ILP is the most appropriate learning method. It is a combination of logic programming and learning which can produce a set of predicates. These predicates can be used to express domain knowledge and planning operators in a convenient way. We are currently applying ILP algorithms to learn planning operators. Once this objective has been reached, we will move onto integrating learning and planning.

References

1. W. van der Aalst, "The Application of Petri Nets to Workflow Management", Journal of Circuits, Systems and Computers, vol.8, no.1, pp.21-66, 1998

2. W. van der Aalst, B. van Dongen, J. Herbst, L. Maruster, G. Schimm, A. Weijters "Workflow Mining: A survey of Issues and Approaches", Data and Knowledge Engineering, vol.47, no.2, pp.237-267, 2003

3. W. van der Aalst, A. ter Hofstede, B. Kiepuszewski, A. Barros, “Workflow Patterns”, BETA Working Paper Series, WP 47, Eindhoven University of Technology, Eindhoven, Netherlands, 2000

4. W. van der Aalst, T. Weijters, L. Maruster, “Workflow Mining: Discovering Process Models from Event Logs”, IEEE Transactions on Knowledge and Data Engineering, vol.16, no.9, pp.1128-1142, 2004

5. M. Adams, D. Edmond, A. ter Hofstede, “The Application of Activity Theory to Dynamic Workflow Adaptation Issues”, 7th Pacific Asia Conference on Information Systems, 2003

6. M. Ai-Chang, J. Bresina, L. Charest, A. Chase, J. Hsu, A. Jonsson, B. Kanefsky, P. Morris, K. Rajan, J. Yglesias, B. Chafin, W. Dias, P. Maldague, “MAPGEN: Mixed-Initiative Planning and Scheduling for the Mars Exploration Rover Mission”, IEEE Intelligent Systems, vol.19, no.1, pp.8-12, 2004

7. G. Alonso, D. Agrawal, A. Abbadi and C. Mohan, “Functionality and Limitations of Current Workflow Managment Systems”, IEEE Expert, vol.12, no.5, 1997

8. G. Alonso, C. Hagen, D. Agrawal, A. Abbadi, C. Mohan, “Enhancing the Fault Tolerance of Workflow Management Systems”, IEEE Concurrency, vol.8, no.3, 2000

9. G. Alonso, C. Mohan, “Workflow management systems: The next generation of distributed procesing tools” in Jajodia and L. Kerschberg (eds.) “Advanced Transaction Models and Architectures”, pp.35-62, Kluwer Academic Publishers, 1997

10. M. Bain, “Learning Logical Exceptions in Chess”, PhD thesis, University of Strathclyde, Glasgow, 1994

11. C. Baral, J. Lobo, “Formalizing Workflows as Cooperative Agents”, Dynamics '97, Workshop in ILPS, 1997

12. R. Barták, “Theory and Practice of Constraint Propagation”, J. Figwer (editor) Proceedings of the 3rd Workshop on Constraint Programming in Decision and Control, Poland, June 2001

13. J. Blythe, E. Deelman, Y. Gil, “Automatically Composed Workflows for Grid Environments”, IEEE Intelligent Systems, vol.19, no.4, pp.16-23, 2004

14. J. Chen-Burger, “Knowledge Sharing and Inconsistency Checking on Multiple Enterprise Models”, Artificial Intelligence Applications Institute for Representation and Reasoning, University of Edinburgh, Research Report EDI-INF-RR-0037, 2001

15. Y.-H. Chen-Burger, “Sharing and Checking Organization Knowledge”, Knowledge Management and Organizational Memories, Kluwer Academic Publishers, pp.183-192, 2002

16. M. Cox, “Planning As Mixed-initiative Goal Manipulation”, Proceedings of the Workshop on Mixed-Initiative Intelligent Systems, 18th International Conference on Artificial Intelligence, 2003

17. S. Decker, M. Erdmann, R. Studer, “A Unifying View of Process Modeling and Knowledge

Engineering", Proceedings of the 10th Knowledge Acquisition for Knowledge-Based Systems Workshop, vol. 34, pp.1-16, 1996

18. T. Estlin, "Using Multi-Strategy Learning to Improve Planning Efficiency and Quality", PhD thesis, Department of Computer Sciences, University of Texas at Austin, 1998

19. M. Fox, D. Long, "PDDL2.1: An Extension to PDDL for Expressing Temporal Planning Domains", Journal of Artificial Intelligence Research, vol. 20, pp.61-124, 2003

20. Y. Gil, E. Deelman, J. Blythe, C. Kesselman, H. Tangmunarunkit "Artificial Intelligence and Grids: Workflow Planning and Beyond", IEEE Intelligent Systems, vol. 19, no.1, pp.26-33, 2004

21. K. Gopal, "An Adaptive Planner Based on Learning of Planning Performance", Master's thesis, Texas A&M University, 2000

22. J. Herbst, "A Machine Learning Approach to Workflow Management", ECML 2000: 11th European Conference on Machine Learning, Lecture Notes in Computer Science, vol.1810, pp.183-194, Springer-Verlag, 2000

23. A. Howe, "Improving the Reliability of Artificial Intelligence Planning Systems by Analyzing their Failure Recovery", IEEE Trans. Knowl. Data Eng., vol.7, no.1, pp.14-25, 1995

24. M. Huhns, M. Singh, "Workflow Agents", IEEE Internet Computing, vol. 2, no.4, pp.94-96, July/August 1998

25. IFIP–IFAC Task Force on Architectures for Enterprise Integration, "GERAM: Generalised Enterprise Reference Architecture and Methodology", Version 1.6.3, 1999

26. P. Jarvis, J. Moore, J. Stader, A. Macintosh, A. Casson-du-Mont, P. Chung, "Exploiting AI Technologies to Realize Adaptive Workflow Systems", Agent-Based Systems in the Business Context, workshop proceedings, AAAI Press, 1999

27. N. R. Jennings, P. Faratin, T. J. Norman, P. O'Brien, B. Odgers, "Autonomous Agents for Business Process Management", Intl. Journal of Applied Artificial Intelligence, vol.14, no.2, pp.145-189, 2000

28. P. Kearney, D. Borrajo, A. Cesta, N. Matino, N. Mehandjiev (Contributors), "PLANET Workflow Management R&D RoadMap", http://planet.dfki.de/, 2003

29. R. Khardon, "Learning to Take Actions", Machine Learning, vol.35, no.1, pp.57-90, 1999

30. J. Klingemann, J. Wäsch, K. Aberer, "Deriving Service Models in Cross-Organizational Workflows", Proceedings of RIDE - Information Technology for Virtual Enterprises (RIDE-VE '99), Sydney, Australia, 1999

31. T. Könik, J. Laird, "Learning Goal Hierarchies from Structured Observations and Expert Annotations", Proceedings of 14th International Conference on Inductive Logic Programming, pp.198-215, Springer, 2004

32. T. Kumar, “Developments in AI Planning”, Technical Report, Knowledge Systems Laboratory, Stanford University, 2002

33. N. Lavrač and S. Džeroski “Inductive Logic Programming: Techniques and Applications”, Ellis Horwood, New York, 1994

34. M. van Lent, J. Laird, “Learning Procedural Knowledge through Observation”, Proceedings of the International Conference on Knowledge Capture, Victoria, British Columbia, Canada, 2000

35. F.-R. Lin, M.-C. Yang, Y.-H. Pai, “A Generic Structure for Business Process Modeling”, Business Process Management Journal, vol.8, no.1, pp.19-41, 2002

36. T. Madhusudan, J. Zao, B. Marshall, “A case-based reasoning framework for workflow model management”, Data Knowledge Engineering, vol.50, no.1, pp.87-115, 2004

37. J. Marques-Silva, “An Overview of Backtrack Search Satisfiability Algorithms”, Fifth International Symposium on Artificial Intelligence and Mathematics, January 1998.

38. R. Medina-Mora, H. Wong, P. Flores, “ActionWorkflow as the Enterprise Integration Technology”, Data Engineering, IEEE Computer Society Press, vol.16, no.2, pp. 49-52, 1993

39. M. Moreno, P. Kearny, D. Meziat, “A Case Study: Using workflow and AI Planners”, PLANSIG 2000: 19th Workshop of the UK Planning and Scheduling, 2000

40. S. Muggleton, L. de Raedt, “Inductive Logic Programming: Theory and Methods”, Journal of Logic Programming, vol.19, no.20, pp.629-679, 1994

41. K. Myers, P. Berry, “At the Boundary or Workflow and AI”, Agent-Based Systems in the Business Context, workshop proceedings, AAAI Press, 1999

42. U. Nilsson and J. Maluszyński “Logic, Programming and Prolog”, 2nd Edition, John Wiley and Sons Ltd., 1995

43. G. Nutt, “The evolution towards flexible workflow systems”, Distributed Systems Engineering 3:276-294, 1996

44. C. Page, A. Srinivasan, “ILP: A Short Look Back and a Longer Look Forward.” Journal of Machine Learning Research, vol.4, no.4, pp.415-430, 2003

45. D. Pearson, “Learning Procedural Planning Knowledge in Complex Environments”, PhD thesis, Computer Science and Engineering, University of Michigan, 1996

46. M. Penker, H.-E. Eriksson “Business Modeling With UML: Business Patterns at Work”, Wiley, 2000

47. S. Russell and P. Norvig “Artificial Intelligence: A Modern Approach”, Prentice Hall, 1995

48. A. Sheth, W. van der Aalst, L. Arpinar, “Processes Driving the Networked Economy”, IEEE Concurrency, pp. 18-31, July-September 1999

49. S. Smith, D. Hildrum and M. Becker "Workflow Management form a Scheduling Perspective", Agent-Based Systems in the Business Context, workshop proceedings, AAAI Press, 1999

50. K. Tae, D. Cook, L. Holder, "Experimentation-Driven Knowledge Acquisition for Planning", Journal of Computational Intelligence, 1999

51. J. Tang and S.-Y. Hwang, "A Scheme to Specify and Implement Ad-hoc Recovery in Workflow Systems", 6th International Conference on Extending Database Technology, Lecture Notes in Computer Science, vol.1377, Springer-Verlag, 1998

52. A. Tate, J. Dalton, J. Levine, "O-Plan: a web-based AI Planning Agent", Proceedings of the National Conference of the American Association of Artificial Intelligence (AAAI-2000), 2000

53. A. Tate et al, "I-X: Technology for Intelligent Systems" (website), http://i-x.info/, 2004

54. M. Veloso, J. Carbonell, A. Pérez, D. Borrajo, E. Fink, J. Blythe "Integrating Planning and Learning: The Prodigy Architecture", Journal of Experimental and Theoretic Artificial Intelligence, vol.7, no.1, 1995

55. M. Veloso, M. Pollack, M. Cox, "Rationale-Based Monitoring for Planning in Dynamic Environments", Proceedings of the Fourth International Conference on Artificial Intelligence Planning Systems, pp. 171–179, 1998

56. M. Voorhoeve, W. van der Aalst, "Ad-hoc Workflow: Problems and Solutions", Proc. Workshop Databases and Expert Systems, pp.36-41, 1997

57. X. Wang, "Learning Planning Operators by Observation and Practice", PhD thesis, Computer Science Department, Carnegie Mellon University, USA, 1996

58. WfMC, "Workflow Process Definition Interface – XML Process Definition Language (XPDL)", Document Number WFMC-TC-1025", 2002

59. T. Zimmerman, S. Kambhampati, "What next for learning in AI Planning?", 2001 International Conference on Artificial Intelligence (IC-AI'2001), Las Vegas, Nevada, USA, 2001

60. T. Zimmerman, S. Kambhampati, "Learning-Assisted Automated Planning: Looking back, Taking Stock, Going Forward", AI Magazine, Summer 2003