The Application of Activity Theory to Dynamic Workflow Adaptation Issues

Michael Adams, David Edmond and Arthur H. M. ter Hofstede

Centre for Information Technology Innovation, Queensland University of Technology
G.P.O Box 2434, Brisbane, Queensland, Australia, 4001
m3.adams@qut.edu.au

Abstract

Workflow Management Systems (WfMSs) are implemented to support the modelling, analysis and enactment of rigidly structured business processes. However, they typically have difficulty supporting unexpected or developmental change occurring in the work practices they model, and are unable to provide adequate support for exceptions, or deviations from the process model, even though such deviations are a common occurrence for almost all processes. These limitations mean a large subset of business practices do not easily translate to the inflexible modelling frameworks imposed by WfMSs, and so has inhibited their wider acceptance.

Workflow modelling frameworks are usually based on software programming principles and proprietary formats, which may not be the most ideal base for the support of flexible work practices. A better approach may be to develop a WfMS based on accepted ideas of how people actually work. This paper derives a set of principles for work practice from a sound theoretical base called "Activity Theory". These grounded principles are then applied to a set of criteria that a WfMS must meet if it is to provide adequate support for flexible work methods. Commercial workflow products, and some research trends, are discussed in relation to the criteria specified.

Keywords

Workflow, Workflow Management, Activity Theory.

1 Introduction

Organizations are constantly seeking efficiency improvements for their business processes. One approach aimed at supporting these objectives has been the implementation of Workflow Management Systems (WfMSs) to configure and control business processes (Dellen, Maurer & Pews 1997), by supporting their modelling, analysis and enactment (Casati 1998).

A workflow can be defined as a composite set of tasks comprised of coordinated computer-based and human activities (Lei & Singh 1997). A workflow model or schema is a formal computerised representation of work procedures that controls the sequence of performed tasks (Bardram 1997b). The key benefits WfMSs seek to bring to an organisation include improved efficiency, better process control, improved customer service, higher maintainability and business process evolution (Reijers, Rigter & van der Aalst 2002, Schael 1998).

The use of WfMSs has grown through their support for the modelling of rigidly structured business processes that in turn derive well-defined workflow instances (Joeris 1999, Reichert & Dadam 1997). However, they typically fail to allow for unexpected or developmental change occurring in the business practices and processes they model (Borgida & Murata 1999). This limitation means a large subset of business processes do not easily translate to the rigid modelling frameworks imposed by WfMSs (Bardram 1997a, Barthelmess & Wainer 1995). Also, WfMSs typically are unable to provide adequate support for exceptions, or deviations from the process model (Casati 1998, Kammer, Bolcer, Taylor & Bergman 1998), even though it has been shown that such deviations are a common occurrence for almost all processes (Barthelmess et al. 1995, Hwang, Ho & Tang 1999, Kammer et al. 1998). Such issues have inhibited the wider implementation of WfMSs.

It is becoming increasingly apparent that a limiting factor in the support of more flexible work practices by workflow systems may be the rigid structure or framework imposed on business process representation and enactment (van der Aalst & Berens 2001). Rather than continue to try to force business processes into rigid frameworks (with limited success), it may be time for a juxtapositional change – first determine the underlying factors inherent in the ways people go about planning, enacting and reflecting on their work, *then* use that knowledge to design a workflow management system that supports how work is actually performed.

Workflow management systems use proprietary conceptual frameworks that are usually postulated rather than grounded in formal theory (Joosten & Brinkkemper 1995). The models that capture work practices should be based on a formal theory and be as simple as possible (Agostini & De Michelis 2000). A more flexible workflow system may be one that is based on accepted ideas of how people work rather than on programming principles.

The aim of this paper is to derive a set of independent principles for workflow support based on *Activity Theory*, a formal theory of human work activities. It is anticipated that the derived principles will prove applicative to further research in the dynamic workflow domain, and deliver systems that better reflect actual work processes. This paper is organised as follows: Section 2 discusses the issues relating to the procedural representation of work practices, including the difficulties in applying the traditional 'assembly-line' metaphor to flexible work environments and the associated problems with process evolution and exception handling. Section 3 introduces Activity Theory. In Section 4, principles derived from Activity Theory that describe work practices are defined. Section 5 describes a set of criteria, each broadly representing a major topic area of workflow research, discusses the relationship between them and the relevant derived principles, and compares each to how well commercial and prototype workflow products meet them. Finally, Section 6 concludes the paper.

2 Issues in Representing Work Practices Procedurally

Whenever a series of actions is undertaken with a view to achieving a pre-conceived result, some plan or set of principles is implemented that guide and shape those actions. In workflow terms, analysts seek to model some aspect of the real world by using a metaphor that both bears some resemblance to the real world, and represents an understanding of how computers work. Such metaphors are abstract constructions that form a common reference model which assist us in representing the external world through computers.

The computational metaphor (Stein 1999) takes a set of inputs, performs a series of functional steps in a strict sequence, and, on completion, produces some output that represents the goal of the process. Thus the computational metaphor supports a single, centralised thread of control, which very much reflects its mathematical ancestry.

Technological developments are, according to Holt (1997), "as much affected by fashion as clothing". Technology does not evolve automatically (as Marx assumed) but rather reflects prevailing human culture (Mumford 1963). New technologies are derived from perceived needs and realised, not in isolation, but through the conventions and norms of their social milieu.

Thus the traditional computational metaphor reveals the influence of pioneers such as von Neumann, and Turing, whose abstract machine proposed 'step-at-a-time' processing, and which in turn reflects the influence on thinking of the contemporaneous development of assembly-line processing (Hendriks-Jansen 1996).

As the prevailing technological advances influenced the structure of early computers, so too has the computational metaphor become a significant model system for the conceptualisation and interpretation of complex phenomena, from cognition to economics to ecology (Stein 1999). Of particular interest is the way the metaphor has been applied to the definition of organisational behaviour issues and the representation of organisational processes.

The computational metaphor remains applicable to well-defined problem domains where goal-directed, sequential, endpoint-driven planning is required (Stein 1999). Such domains were the early beneficiaries of workflow management systems. Consequently, current commercial workflow systems provide support for standardised, repetitive activities that do not vary between execution instances.

Adherence to the metaphor by WfMSs has been an important factor in their acceptance by organisations with structured work practices. Descriptions can be found throughout the workflow literature to the 'processing', 'manufacturing' and 'assembly-line' modelling metaphors that are employed by commercial workflow systems. However, while the Workflow Management Coalition claims that "even office procedures can be processed in an assembly line" (Workflow Management Coalition 2002), there are many aspects where administrative and service processes differ from manufacturing processes (van der Aalst et al. 2001).

Even for highly structured work practices (such as banking and air-traffic control), it remains difficult (if not impossible) to successfully capture *all* work activities, and in particular all of the task sequences possible, in a standard workflow model, at least not without producing some very complex models. It may be that the computational metaphor has become an inhibiting factor in the development of workflow systems able to effectively support flexible work practices.

The prescriptive and rigid nature of workflow systems do not allow for unanticipated variations (Borgida et al. 1999). But in many environments, business processes vary greatly, are dynamic with regards to their actual execution (Reichert & Dadam 1997), and may differ between individual instances. Many technologically adept systems fail because they ignore human and social factors (Schael 1998).

The process followed for successfully completing any human activity is constantly evolving. A worker, or group of workers, given responsibility for a particular task, will naturally seek to minimise the amount of work necessary for completing the task by deriving new, more efficient methods (Bardram 1997a).

Where a workflow model mediates a work practice, that model represents at best a point in time or snapshot of the particular process being modelled. That is, there is generally no flexibility allowed in the system to accommodate the natural evolution of the process, nor the evolution of organisational goals. Manual interventions into the workflow process become increasingly frequent as staff attempt to manipulate workflow inputs and outputs to match workplace changes (Strong & Miller 1995).

Such manual intrusions also mean that the corrective actions are not archived by the system, since the system knows nothing about the actions being taken. Therefore, there is no recorded 'organisational memory' (Ackerman & Halverson 1998, Larkin & Gould 2000) of the valid business process modifications, which means that the evolved knowledge will not be incorporated into future iterations of the system-based process.

In fact, it is because of the discrepancies between real-world activities and formal representations of them that workflow process instances typically experience a failure or exception during their execution. Traditionally, exceptions are events that by definition occur rarely. But exceptions in workflow enactions have a wider meaning than commonly understood by the term. Rather than being errors, they are simply events that were unaccounted for in the original model of the business process. Such events happen rather frequently in real working environments (Hwang et al. 1999).

Any unexpected exceptions that occur either require manual intervention, or result in a process abort. Since most production workflow processes are long and complex, neither intervention nor process termination are satisfactory solutions (Hagen & Alonso 1998).

Thus the application of workflow management systems has been limited, due to the lack of flexibility inherent in a system that, by definition, imposes rigidity. This inflexibility extends to the management of process evolution and exceptions, which places further limits on how accurately a workflow model can reflect the actual business process it is based on. The resulting process models are 'system-centric', meaning that work processes are *straight-jacketed* (van der Aalst et al. 2001) into the supplied framework, rather than the metaphor reflecting the way work is actually performed. As a result, users are forced to work outside of the system, and/or constantly revise the (static) process model, in order to successfully complete their activities, thereby negating the efficiency gains sought by implementing a workflow solution in the first place.

It is therefore desirable to extend the capabilities of WfMSs, so that the benefits offered to organisations employing rigidly defined, 'production-line' processes could also be enjoyed by those businesses which employ more creative or flexible processes. A new approach is needed which better incorporates the fluidity between a computer, its user and their environment. However, before such an approach can be realised, a thorough understanding of the network of interrelationships between workers, goals, tools and environment must be built.

A workflow management system that better supports flexible work environments requires a sound theoretical foundation that describes how work is conceived, carried out and reflected upon. One such theoretical base can be found in *Activity Theory*.

3 An Introduction to Activity Theory

The following is a necessarily brief synopsis of Activity Theory. More detailed treatments can be found in: (Engestrom 1987, Leontiev 1974, Nardi 1996b, Wertsch 1981).

Activity Theory is a powerful and clarifying descriptive tool focussing on understanding human activity and work practices, and incorporates notions of intentionality, history, mediation, collaboration and development (Nardi 1996a).

Activity Theory originated in the former Soviet Union in the 1920's and 1930's as part of the cultural-historical school of psychology founded by Vygotsky (1925), and appropriates Marxist ideas about how tools or instruments mediate work activities.

As opposed to the study of the individual as a separate entity, in Activity Theory the unit of analysis is the entire activity. An activity is composed of a subject, and an object, mediated by a tool. A subject is a person or a group engaged in an activity. An object (in the sense of 'objective') is held by the subject and motivates the activity, giving it a specific direction. The mediation can occur through the use of many different types of tools, material tools as well as mental tools, including culture, ways of thinking and language. Computers are considered to be special kinds of mediating tools (Kaptelinin 1996).

Transforming the object into an outcome motivates the existence of the activity. An object can be a material thing, less tangible (a plan) or totally intangible (a common idea), as long as it can be shared for manipulation and transformation by the activity participants (Kuutti 1996).

Leontiev (1974) extended Activity Theory by including the concept of collective activity, adding the ideas of *Community, Rules* and *Division of Labour* to denote the situated social context within which collective activities are carried out. Community is comprised of one or more people who share the objective with the subject. Rules regulate actions and interactions within an activity. The division of labour informs how tasks are divided horizontally between community members as well as referring to any vertical division of power and status. Just as artefacts mediate the relationship between subject and object, rules mediate the relationship between subject and community, and division of labour between community and object.

Activities are not isolated units but are influenced by other activities and environmental changes – these influences may cause imbalances. Activity Theory uses the term *contradiction* (Engestrom 1987) to indicate a misfit between elements, within them, and between different activities. Contradictions manifest themselves as problems, ruptures, breakdowns, clashes – i.e. exceptions. Tension that arises between elements of the Activity System identify areas where systems no longer match the activities they model (Collins, Shukla & Redmiles 2002). Activity Theory sees contradictions not as problems but as sources of development; activities are virtually always in the process of working through contradictions that subsequently facilitate change (Nardi 1996a).

Leontiev also identified that individuals within a collective activity may have different tasks to perform, each with its own sub-goal, that combined move the group towards the desired outcome. To delineate an individual's behaviour from the collective activity, the hierarchical distinction was added between an *Activity*, consisting of several *Actions*, each consisting of several *Operations* (Bardram 1997a). An action is a single task performed to achieve a self-contained, pre-conceived result relevant to the overall activity. Operations are the work functions or routines within each action determined by the actual conditions and contexts of the action during its performance.

4 Principles Derived from Activity Theory

The value of any theory is not whether it provides an objective representation of reality (Bardram 1998b), but rather how well it can shape an object of study, highlighting relevant issues (Barthelmess & Anderson 2002). Ten fundamental principles, representing an interpretation of the central themes of Activity Theory applicable to an understanding of organisational work practices, are summarized below. These principles will be referred to as *p1...p10* in later sections.

- *Principle 1:* *Activities are hierarchical* An activity consists of one or more actions. Each action consists of one or more operations.
- *Principle 2:* *Activities are communal* An activity almost always involves a community of participants working towards a common objective.
- *Principle 3:* *Activities are contextual* Contextual conditions and circumstances deeply affect the way the objective is achieved in any activity.
- *Principle 4:* *Activities are dynamic* Activities are never static but evolve asynchronously, and historical analysis is often needed to understand the current context of the activity.
- *Principle 5:* *Mediation of activity* An activity is mediated by tools, rules and divisions of labour.
- *Principle 6:* *Actions are chosen contextually* A repertoire of actions and operations is created, maintained and made available to any activity, which may be performed by making contextual choices from the repertoire.
- *Principle 7:* *Actions are understood contextually* The immediate goal of an action may not be identical to the objective of the activity of which the action is a component. It is enough to have an understanding of the overall objective of the activity to motivate successful execution of an action.
- *Principle 8:* *Plans guide work* A plan is not a blueprint or prescription of work to be performed, but merely a guide which is modified depending on context during the execution of the work.
- *Principle 9:* *Exceptions have value* Exceptions are merely deviations from a pre-conceived plan. Deviations will occur with every execution of the plan, and give rise to a learning experience which can then be incorporated into future executions.
- *Principle 10:* *Granularity based on perspective* A particular piece of work might be an activity or an action depending on the perspective of the viewer.

5 Functionality Criteria Based on Derived Principles

This section discusses six functionality criteria that a WfMS needs to meet to support the principles of work practice derived from Activity Theory. Each criterion broadly represents a major topic area of workflow research. Each is discussed with reference to the relevant

Activity Theory principles (see *Table 1* for a summary mapping), and how well it is supported by both leading commercial workflow products, and some research prototypes.

	Flexibility & re-use	Adaptation via reflection	Dynamic evolution	Locality of change	Comprehen-sibility of models	Exceptions as "first-class citizens"
Activities are hierarchical	■			■	■	
Activities are communal			■			
Activities are contextual	■	■				
Activities are dynamic		■		■		
Mediation of Activity	■	■	■			
Actions are chosen contextually	■					■
Actions are understood contextually			■	■		
Plans guide work		■			■	■
Exceptions have value		■	■			■
Granularity based on perspective					■	

Table 1. Summary mapping of Activity Theory principles vs. workflow functionality criteria

Generally speaking, the currently available commercial workflow systems rely on a monolithic, single-schema architecture, which makes it difficult to fully capture the business processes to be supported (Bichler, Preuner & Schrefl 1997), and has been recognised as a major limiting factor in the uptake of WfMSs (Heinl, Horn, Jablonski, Neeb, Stein & Teschke 1999). Also, the commercial WfMSs provide little support for exception handling at the process-conceptual and instance-execution layers, besides triggers for deadlines being reached (Casati & Pozzi 1999). Even then, most simply inform a system administrator of the event – it is expected that the exception will be dealt with off-system.

The functionalities of the five commercial systems reviewed (Staffware (Staffware plc 1999), COSA (Baan Company N.V. 1999), HP ChangeEngine (Hewlett-Packard Company 2001), IBM MQ-Series (International Business Machines Corporation 2001) and SAP R/3 (SAP AG 1997)) were gleaned from their relevant manuals and other literature.

Criterion 1: Flexibility and re-use Workflow systems should support the utilisation of actions contextually chosen from a repertoire, which can be employed in several different activities.

The analysis of human activity as a three-level hierarchical structure emphasizes that the same activity can be realized through different actions and the same action can be realized through different operations (Bardram 1997a). Conversely, the same action can be a component of several different activities *(p1, p5)*.

While some workplaces have strict operating procedures because of the work they do (e.g. air traffic control) many workplaces successfully complete activities by developing a set of informal routines that can be flexibly and dynamically combined to solve a large range of problems (Ackerman et al. 1998). Thus, realizing an activity in a contingent environment is aided considerably by having a repertoire of actions and operations to choose from (Bardram 1997a) *(p6)*.

This denotes a crucial factor for the representation of flexible workflows. At any point in time, there may be several possible sequences that can be followed utilising a sub-set of available actions to achieve the objective of the activity. Choices are made dependent on the actual circumstances of the activity at that point in time *(p3)*.

Thus a WfMS would need to manage a catalog of actions that at runtime could ideally be chosen programmatically, but in any case with minimal human intervention, as contextual information presents itself. Since each action has it own goal, each should be a distinct, self-contained workflow process in its own right.

The availability of a catalog of actions would also encourage reuse, since actions may be made available for use in several distinct activities, and/or used as templates for the definition of new actions.

By using a modular approach, a resultant model may range from a simple skeleton to which actions can be added at runtime (supporting dynamic change) or may be a fully developed model representing the *a-priori* complete work practice, depending on user and organisational needs.

The commercial products reviewed all provide modelling frameworks that are basically monolithic, but with various levels of support for the deconstruction of tasks. Staffware provides 're-usable process segments' that can be inserted into any process. ChangeEngine provides some support for sub-processing, but not as distinct workflows. SAP R/3 allows for the definition of 'blocks' that can be inserted into other 'blocks', thus providing some support for encapsulation and reuse. COSA supports parent-sibling processes, where data can be passed to/from a process to a sub-process.

The *ADOME* system (Chiu, Karlapalem & Li 1998, Chiu, Li & Karlapalem 1999, Chiu, Li & Karlapalem 2000) provides templates that can be used to build a workflow model, and provides some support for (manual) dynamic change. A catalog of 'skeleton' patterns that can be instantiated or specialised at design time is supported by the *WERDE* system (Casati 1998, Casati, Castano, Fugini, Mirbel & Pernici 1998).

Criterion 2: Adaptation via reflection Workflow systems should support evolutionary adaptation of processes based on the experience gained during each execution of the process.

All human activity is guided by the anticipation of fulfilling the objective of that activity. This anticipation is formed as a result of reflection on experiential memory (Bardram 1997b). Thus, a work plan is the result of reflection on prior experiences and in anticipation of a particular goal. Because of this, plans are not rigid and accurate descriptions of the execution steps but always incomplete and tentative (Kuutti 1996) *(p5, p8)*.

An instantiation of a plan is fundamentally distinct from the plan itself. A plan is a work resource (Schael 1998), detached from concrete activities, and is used to organise and divide the work amongst the participants involved. To achieve the expected result, the actions and operations contained in the plan (conceptual) have to be adjusted to the material conditions of each situation (contextual) (Symon, Long & Ellis 1996).

Depending on the context of the instantiation, some actions will mirror the plan, while other parts of the plan may be either discarded or augmented in some way. After completion, a comparison is made between the anticipated and actual outcomes, and any incongruities or deviations from the plan adds to the experience of the participants, and so gives rise to a learning situation *(p3, p4, p8)*.

Plan adaptations for future instantiations can be achieved by recording the occurrence of deviations and incorporating them as required (Sadiq & Orlowska 1999). Thus a plan is an artefact that contains historical residue of its development (Nardi 1996a) *(p5, p8)*.

Typically, workflow systems ignore these deviations, resulting in systems that remain static in the face of change. This leads to increasing instances of work being performed off-system to accommodate those changes (Strong et al. 1995).

A dynamic, flexible, evolving workflow support system must have the ability to record deviations from the plan, why the deviation occurred and how it was handled, thus handling the deviation on-system. In addition, the 'handlers' thereafter simply become part of the plan (i.e. a possible path) for future instantiations *(p9)*.

Therefore, it is important that the modelling tool allows for the ongoing evolution and dynamic modification of a model based on experience gained while operating an instantiation of it. Also, a guided choice should be made available to the designer, allowing for the implementation of an appropriate action chosen from the catalog. A designer might be guided by the system to choose certain actions over others based on system 'intelligence' using archival data and heuristic algorithms.

All the commercial products reviewed require the model to be fully defined before it can be instantiated, and changes must be incorporated by modifying the model statically. They provide little support for learning from past instantiations besides keeping basic audit logs of executed events in a database table, with a list facility available for manual perusal. ChangeEngine and Staffware also provide monitoring systems which allow administrators to observe current executions, view statistics and be informed when certain milestones are reached.

In the research field, the *Milano* system aims to provide process models as 'resources for action' rather than strict blueprints of work practices (Agostini & De Michelis 1996). The *ADEPT* system is process-based, rather than functionally-based, and is designed to support dynamic change at runtime (Dadam, Reichert & Kuhn 1997, Reichert & Dadam 1997, Reichert & Dadam 1998). The *WAM* system (Faustmann 1996) archives all events during execution, which are subsequently weighted and categorised. A 'causality tree' is then composed which provides insights into how the work was carried out. Approaches to extracting improved models from logs of previous executions are proposed in (Agrawal, Gunopulos & Leymann 1998), (van der Aalst & van Dongen 2002) and (van der Aalst, Weijters & Maruster 2002).

Criterion 3: Dynamic evolution of work practices Workflow systems should support the evolution of processes towards individual specialisations without risking the loss of motivation for the overall activity.

Almost all work practices are collective activities that involve cooperation and mutual dependence between participants *(p2)*. Even so, practices generally evolve towards minimizing mutual articulation among individuals, without jeopardizing the overall objective of the collective activity. Activity Theory recognises this as a strict division of labour, enabling participants to specialize in certain actions, and by designing certain structured artefacts that encapsulate actions in an efficient way (Bardram 1997a) *(p5)*.

However, over-specialisation of an action leads to deskilling, and therefore loss of motivation, of the workers involved (Kaasboll & Smordal 1996). This phenomenon, referred to as *context-tunnelling*, becomes an issue when a participant is no longer motivated by the objective of the activity *(p7)*. Workflows structured as an assembly line are often the cause of such situations, because they force workers to become mere processors in a product delivery chain – that is, the human contextual factors are ignored. The manifestation of the inherent need to introduce variety into work processes (Rauterberg 1999) is often incorrectly interpreted as exception producing *(p9)*.

Activity Theory recognises that there is no need for each participant to know the detail of the *entire* activity (as asserted by proponents of case handling systems (Reijers et al. 2002)), but only the context of their own action in relation to the whole activity. Since a participant is motivated by the objective of an activity, only an understanding of how the goal of their own action contributes to the overall objective, and the freedom to seek efficiencies and/or variety in achieving that goal, is required *(p7)*.

All the reviewed commercial products present to each participant only that information they require to complete their pre-defined task. An example of a case-handling system is *FLOWer* (Pallas Athena 2000), which addresses the context-tunnelling problem by capturing and presenting data pertaining to the case via a series of forms to all participants.

Criterion 4: Locality of Change Modifications should be able to be fully applied by changing a minimal number of components, and should impact minimally on associated components.

Related to support for flexibility is support for *locality of change*. Bass, Clement and Kazman (1998) suggest that increasing the adaptability of software is largely a function of how localized necessary modifications can be made. In terms of workflow process models, this idea suggests two desirable and related goals. Firstly, to ensure that a workflow process model is strongly adaptable, modifications should be able to be limited to as few components as possible. Secondly, any changes made within those components should impact on other components to the least extent possible.

Activity Theory describes the focus for change as the individual participant who conceptualises improvements in methods at the action level *(p4, p7)*. Therefore, a workflow support system can achieve adaptability by strongly providing for locality of change at that level. One approach would be to support the definition of a workflow process as a set of sub-processes, each of which is a distinct entity representing a single action *(p1)*. Also, since each sub-process will be fully encapsulated, any changes made to one will not impact other sub-processes available to the same model. This is not achievable with monolithic modelling paradigms.

All the commercial products reviewed require modifications to be applied to the whole model – that is, change is not a trivial exercise for models of any complexity. Also, no research was found that supports the idea of sub-processes being distinct workflows in their own right.

Criterion 5: <u>*Comprehensibility of process models*</u> Process models should be comprehensible to all stakeholders, and should support representation at different levels of granularity.

A major limiting factor in the uptake of workflow systems is the complexity of the models developed for all but the most trivial activities. WfMSs that require the development of monolithic models fail to take into account the complexities in adapting and evolving those models. Also, there may be stakeholders that have difficulty in interpreting models that contain many possible paths on a single plane, and/or require different levels of abstraction to be presented *(p1, p10)*.

While the modeller will have a detailed understanding of the components and structure of the model, a workplace manager will generally not have the expertise to decipher it, nor will a staff member charged with performing a particular action *(p8)*. However, the very acceptance of the workflow solution may hinge on being able to provide management with a modelled representation of a work process which is comprehensible to them (Carlsen 1996).

An approach which may offer better understanding is the representation of workflow models as a hierarchical set of linked sub-processes, since each sub-process would be fully encapsulated and therefore a (simpler) workflow model in its own right. This approach would also allow for the model to be represented at different levels of granularity for different stakeholders.

While the various abilities of the analysed commercial products to support sub-processes have been stated above, none have the ability to present models of differing granularity to stakeholders. One research prototype that does is *SWORDIES* (Endl, Knolmayer & Pfahrer 1998), which uses rules to represent work processes at different abstraction levels that are built into an 'integration layer' at runtime.

Criterion 6: <u>*The elevation of exceptions to "first-class citizens"*</u> Workflow exceptions should not be regarded as errors but as events that provide an opportunity for a learning experience and therefore are a valuable part of any work practice.

Generally, commercial WfMSs take the view that exceptions are to be considered errors. As such, they are seen to be annoyances which either should have been foreseen and therefore prevented from occurring in the first place, or perceived as impossible to predict and therefore best left handled off-system.

However, Activity Theory regards exceptions (contradictions, diversions, tensions, deviations) as a normal and valuable part of every work activity, which can be used to refine or evolve the process towards improved efficiencies. As a result, exceptions should be implemented in a positive fashion to better reflect (within the workflow model) the actuality of the work practice it supports *(p9)*.

The ability to capture exceptional events and deal with them appropriately in real time, while concomitantly using that exception as a learning experience to evolve the workflow, would be a central requirement of any workflow system based on Activity Theory.

An effective system must also provide the ability to source an appropriate exception 'handler', then incorporate it into the running instance. That is, exception handling techniques should be available during both the design and execution phases.

The principle of contextual choice from a repertoire of actions may also apply as an appropriate mechanism for exception handling *(p6, p9)*. If, based on a set of conditions, contexts, archival data and so on, a matching handling sub-process could be found for a particular event, then that process could be automatically invoked. If no such match could be found, then a list of possible matches, selected from a catalog of processes by applying heuristic techniques, might be presented to an administrator from which an appropriate choice could be made.

The extraction of a general knowledge of workflow events using knowledge discovery methods based on archival data, conditions and context reflects the 'learning system' central to Activity Theory *(p8, p9)*.

Although there has been much research in the field of workflow exceptions, few results have been incorporated by commercial workflow products (Casati & Pozzi 1999, van der Aalst et al. 2002). Currently, there is only trivial support amongst them for exception handling. Staffware provides constructs called *event nodes*, which suspend execution until the occurrence of a pre-defined exception, at which time an exception handling path or sequence is activated. COSA provides support for timeout exceptions only, at which time notification is sent to an administrator. ChangeEngine allows administrators to manually intervene in process instances to handle exceptional events by initiating messages to executing processes – 'detectors' for each exceptional event (including a manual command to abort the process) must be built as nodes into the standard model. MQ Series provides no concept of exceptions as expected or unexpected events at the conceptual or process execution layers besides simple deadline, which when triggered sends a message to an administrator. SAP R/3 provides for pre-defined branches which are built into the standard monolithic model. When the exception occurs, an administrator is presented with the set of possible branches, from which a choice can be made manually.

The *OPERA* prototype (Hagen et al.1998) allows for exceptions to be handled at the task level, or propagated up various ancestor levels throughout the running instance. It also removes the need to define the exception handler *a-priori*, although the types of exceptions handled appear to be more at the transactional level. The *eFlow* system (Casati, Ilnicki, Jin, Krishnamoorthy & Shan 2000) uses rules to define exceptions, although they cannot be defined separately to the standard model.

Conclusion

This work extends the potential applicability of Activity Theory to the implementation of more flexible and better directed workflow support and management systems. Activity Theory offers a number of interesting insights into current workflow research domains, particularly the related issues of workflow adaptability, flexibility, evolution and exception handling. It is intended that the principles derived in this paper will provide guidance for further research in these areas.

While the commercial products reviewed support few of the principles derived from Activity Theory, and research prototypes support some individually, there are no systems that meet most or all of the principles. Recent work in IT research in relation to Activity Theory has been undertaken in the areas of Human-Computer Interface design (Bodker 1991, Bodker & Greenbaum 1992, Kuutti 1996, Kaptelinin 1996, Nardi 1996a), Artificial Intelligence (Hendriks-Jansen 1996, Rauterberg & Felix 1996, Rauterberg 1999, Stein 1999), Cooperative Learning (Issroff & Scanlon 2001, Larkin et al. 2000) and Software Analysis and Design (Bardram 1997a, Bardram 1998a, Bodker & Greenbaum 1992, Collins et al. 2002, Gould et al. 2000, Mwanza 2000, Turner, Turner & Horton 1999). However, there has been very little work done on the direct relationship of Activity Theory to workflow, which this paper hopefully goes some way to addressing.

References

van der Aalst, W. M. P. and Basten, T. (2002), 'Inheritance of Workflows: An Approach to Problems Related to Change', *Theoretical Computer Science*, Vol. 270(1-2), pp. 125-203.

van der Aalst, W. M. P. and Berens, P. J. S. (2001), 'Beyond Workflow Management: Product-Driven Case Handling' in *International ACM SIGGROUP Conference on Supporting Group Work (GROUP 2001)*, New York, pp. 42-51.

van der Aalst, W. M. P. and van Dongen, B. F. (2002), *Discovering Workflow Performance Models from Timed Logs*, Technical Report, Eindhoven University of Technology.

van der Aalst, W. M. P., Weijters, A. J. M. M. and Maruster, L. (2002), 'Workflow Mining: Which Processes can be Rediscovered?', BETA Working Paper Series WP74, Eindhoven University of Technology.

Ackerman, M. S. and Halverson, C. (1998), 'Considering an Organization's Memory' in *Proceedings of the ACM 1998 Conference on Computer Supported Cooperative Work*, ACM Press, pp. 39-48

Agostini, A. and De Michelis, G. (1996), *Modeling the Document Flow within a Cooperative Process as a Resource for Action*, Technical Report CTL-DSI, University of Milano.

Agostini, A. and De Michelis, G. (2000), 'A Light Workflow Management System Using Simple Process Models', *Computer Supported Cooperative Work,* Vol. 9(3/4), pp. 335-363.

Agrawal, R., Gunopulos, D. and Leymann, F. (1998), 'Mining Process Models from Workflow Logs', *Lecture Notes in Computer Science,* Vol 1377**,** pp. 469-484.

Baan Company N.V. (1999), 3rd edn, *COSA Workflow Version 3: Business Process Designer's Guide*, Pulheim, Germany.

Bardram, J. E. (1997a), 'I Love the System - I Just Don't Use It!' in *Proceedings of the 1997 International Conference on Supporting Group Work (GROUP'97)*, Phoenix, Arizona.

Bardram, J. E. (1997b), 'Plans as Situated Action: an Activity Theory Approach to Workflow Systems' in *Proceedings of the 1997 European Conference on Computer Supported Cooperative Work (ECSCW'97)*, Lancaster U.K., pp. 17-32.

Bardram, J. E. (1998a), *Collaboration, Coordination and Computer Support*, PhD thesis, University of Aarhus.

Bardram, J. E. (1998b), 'Designing for the Dynamics of Cooperative Work Activities' in *Proceedings of the ACM 1998 Conference on Computer-Supported Cooperative Work*, Seattle, pp. 89-98

Barthelmess, P. and Anderson, K. M. (2002), 'A View of Software Development Environments Based on Activity Theory', *Computer Supported Cooperative Work,* Vol. 11(1/2), pp. 13-37.

Barthelmess, P. and Wainer, J. (1995), 'Workflow Systems: a Few Definitions and a Few Suggestions' in *ACM Conference on Organizational Computing Systems (COOCS'95)*, San Jose, California, pp. 138-147.

Bass, L., Clement, P. and Kazman, R. (1998), *Software Architecture in Practice,* Addison-Wesley, Reading.

Bichler, P., Preuner, G. and Schrefl, M. (1997), 'Workflow Transparency' in *Proceedings of the 9th International Conference for Advanced Information Systems Engineering (CAiSE'97), Barcelona, Spain*, Vol. 1250.

Bodker, S. (1991), 'Activity Theory as a Challenge to Systems Design' in *The Information Systems Research Arena of the 90s, Challenges, Perceptions and Alternative Approaches,* ed. H.-E. Nissen, Amsterdam.

Bodker, S. and Greenbaum, J. (1992), *Design of Information Systems: Things versus People,* Technical Report, Aarhus University.

Borgida, A. and Murata, T. (1999), 'Tolerating Exceptions in Workflows: a Unified Framework for Data and Processes' in *International Joint Conference on Work Activities, Coordination and Collaboration (WACC'99)*, San Francisco, CA, ACM Press, pp. 59-68.

Carlsen, S. (1996), 'Comprehensible Business Process Models for Process Improvement and Process Support' in *CAiSE'96: 3rd Doctoral Consortium on Advanced Information Systems Engineering*, Hereklion, Greece.

Casati, F. (1998), 'A Discussion on Approaches to Handling Exceptions in Workflows' in Proceeding of the *1998 Conference on Computer-Supported Cooperative Work*, Seattle.

Casati, F., Castano, S., Fugini, M., Mirbel, I. and Pernici, B. (1998), 'WERDE: a Pattern-based Tool for Exception Design in Workflows' in *CAiSE'98: 10th International Conference on Advanced Information Systems Engineering*, Pisa, Italy.

Casati, F., Ilnicki, S., Jin, L., Krishnamoorthy, V. and Shan, M.-C. (2000), 'Adaptive and Dynamic Composition in eFlow' in *12th International Conference, CAiSE 2000*, Stockholm, Sweden, pp. 13-31.

Casati, F. and Pozzi, G. (1999), 'Modelling Exceptional Behaviours in Commercial Workflow Management Systems' in *1999 IFCIS International Conference on Cooperative Information Systems*, Edinburgh, Scotland, pp. 127-138.

Chiu, D., Karlapalem, K. and Li, Q. (1998), 'Exception Handling with Workflow Evolution in ADOME-WFMS: a Taxonomy and Resolution Techniques' in *1998 Conference on Computer-Supported Cooperative Work*, Seattle.

Chiu, D., Li, Q. and Karlapalem, K. (1999), 'A Metamodeling Approach to Workflow Management Systems Supporting Exception Handling', *Information Systems,* Vol. 24(2), pp. 159-184.

Chiu, D., Li, Q. and Karlapalem, K. (2000), 'A Logical Framework for Exception Handling in ADOME Workflow Management System' in *12th International Conference CAiSE 2000*, Stockholm, Sweden, pp. 110-125.

Collins, P., Shukla, S. and Redmiles, D. (2002), 'Activity Theory and System Design: A View from the Trenches', *Computer Supported Cooperative Work,* Vol. 11(1/2), pp. 55-80.

Dadam, P., Reichert, M. and Kuhn, K. (1997), 'Clinical Workflows - The Killer Application for Process-Oriented Information Systems?' *Ulmer Informatik-Berichte*, Vol. 97(16).

Dellen, B., Maurer, F. and Pews, G. (1997), 'Knowledge-Based Techniques to Increase the Flexibility of Workflow Management', *Data & Knowledge Engineering Journal,* Vol. 23(3), pp. 269-295.

Endl, R., Knolmayer, G. and Pfahrer, M. (1998), 'Modeling Processes and Workflows by Business Rules' in *1st European Workshop on Workflow and Process Management (WPM'98)*, Swiss Federal Institute of Technology (ETH), Zurich.

Engestrom, Y. (1987), *Learning by Expanding: An Activity-Theoretical Approach to Developmental Research,* Orienta-Konsultit, Helsinki.

Faustmann, G. (1996), 'Workflow management and Causality Trees' in *Proceedings of the Second International Conference on the Design of Cooperative Systems*, France.

Gould, E., Verenikina, I. and Hasan, H. (2000), 'Activity Theory as a Basis for the Design of a Web Based System of Enquiry for World War 1 Data' in *Proceedings of IRIS 23 Laboratorium for Interaction Technology*, (eds, Svennson, L., Snis, U., Sorensen, C., Fagerlind, H., Lindroth, T., Magnusson, M. and Ostlund, C.), University of Trollhattan Uddevalla.

Hagen, C. and Alonso, G. (1998), *Flexible Exception Handling in Process Support Systems*, Technical Report No. 290, ETH, Zurich.

Heinl, P., Horn, S., Jablonski, S., Neeb, J., Stein, K. and Teschke, M. (1999), 'A Comprehensive Approach to Flexibility in Workflow Management Systems' in *Proceedings of WACC'99*, San Francisco, CA.

Hendriks-Jansen, H. (1996), *Catching Ourselves in the Act: Situated Activity, Interactive Emergence, Evolution and Human Thought*, MIT Press, Cambridge.

Hewlett-Packard Company (2001), ed. 4.2, *HP Process Manager: Process Design Guide*, Palo Alto, CA.

Holt, A. W. (1997), *Organized Activity and Its Support by Computer,* Kluwer Academic Publishers, Dordrecht.

Hwang, S.-Y., Ho, S.-F. and Tang, J. (1999), 'Mining Exception Instances to Facilitate Workflow Exception Handling' in *6th International Conference on Database Systems for Advanced Applications*, Hsinchu, Taiwan.

International Business Machines Corporation (2001), 5th ed., *IBM MQSeries Workflow: Administration Guide Version 3.3*.

Issroff, K. and Scanlon, E. (2001), 'Case Studies Revisited: What Can Activity Theory Offer?' in *European Perspectives on Computer-Supported Collaborative Learning. Proceedings of the First European Conference on Computer-Supported Collaborative Learning*, Maastricht, Holland.

Joeris, G. (1999), 'Defining Flexible Workflow Execution Behaviours' in *Enterprise-wide and Cross-enterprise Workflow Management - Concepts, Systems, Applications,* eds. P. Dadam and M Reichert, University of Ulm.

Joosten, S. and Brinkkemper, S. (1995), 'Fundamental Concepts for Workflow Automation in Practice' in *Proceedings of the International Conference on Information Systems (ICIS'95)*, Amsterdam.

Kaasboll, J. J. and Smordal, O. (1996)**,** 'Human Work as Context for Development of Object - Oriented Modelling Techniques', in *IFIP WG 8.1/8.2 Working Conference on Principles of Method Construction and Tool Support*, pp.111-125, Chapman and Hall, Atlanta.

Kammer, P. J., Bolcer, G. A., Taylor, R. N. and Bergman, M. (1998), *Techniques for Supporting Dynamic and Adaptive Workflow*, Technical Report, Department of Information and Computer Science, University of California.

Kaptelinin, V. (1996), 'Computer-Mediated Activity: Functional Organs in Social and Developmental Contexts' in *Context and Consciousness: Activity Theory and Human Computer Interaction*, ed. B. A. Nardi, MIT Press, Cambridge, pp. 45-67.

Kuutti, K. (1996), 'Activity Theory as a Potential Framework for Human-Computer Interaction Research' in *Context and Consciousness: Activity Theory and Human Computer Interaction*, ed. B. A. Nardi, MIT Press, Cambridge, pp. 17-44.

Larkin, P. A. K. and Gould, E. (2000), 'Activity Theory Applied To The Corporate Memory Loss Problem' in *Proceedings of IRIS 23 Laboratorium for Interaction Technology*, University of Trollhattan, Uddevalla.

Lei, Y. and Singh, M. P. (1997), 'A Comparison of Workflow Metamodels' in *ER-97 Workshop on Behavioral Modeling and Design Transformations: Issues and Opportunities in Conceptual Modeling*, Los Angeles.

Leontiev, A. N. (1974), ' The Problem of Activity in Psychology' *Soviet Psychology,* Vol. 13(2), pp. 4-33.

Mumford, L. (1963), *Technics and Civilization,* Harcourt Brace Jovanovich, New York.

Mwanza, D. (2000), 'Mind the Gap: Activity Theory and Design, KMi Technical Reports, KMi-TR95, Knowledge Media Institute, The Open University, Milton Keynes, UK, http://kmi.open.ac.uk/~publications/techreports.html.

Nardi, B. A. (1996a), 'Activity Theory and Human-Computer Interaction' in *Context and Consciousness: Activity Theory and Human Computer Interaction*, ed. B. A. Nardi, MIT Press, Cambridge, pp. 7-16.

Pallas Athena (2000) *Case Handling with FLOWer: Beyond Workflow,* Positioning Paper, http://www.pallas-athena.com/downloads/eng_flower/flowerup.pdf.

Rauterberg, M. (1999), 'Activity and Perception: an Action Theoretical Approach', *Systematica,* Vol. 14(1), pp. 1-6.

Rauterberg, M. and Felix, D. (1996), 'Human Errors: Disadvantages and Advantages' in *Proceedings of the 4th Pan Pacific Conference on Occupational Ergonomics*, Hsinchu, Taiwan.

Reichert, M. and Dadam, P. (1997), 'A Framework for Dynamic Changes in Workflow Management Systems' in *8th International Workshop on Database and Expert Systems Applications (DEXA 97)*, Toulouse, France, IEEE Computer Society Press, pp. 42-48.

Reichert, M. and Dadam, P. (1998), 'ADEPTflex - Supporting Dynamic changes of Workflows Without Losing Control, *Journal of Intelligent Information Systems - Special Issue on Workflow Managament,* Vol. 10(2), pp. 93-129.

Reijers, H., Rigter, J. and van der Aalst, W. (2002), *The Case Handling Case*, Technical Report, Eindhoven University of Technology.

Sadiq, S. and Orlowska, M. (1999), 'On Capturing Process Requirements of Workflow Based Business Information Systems' in *Proceedings of the 3rd International Conference on Business Information Systems (BIS99)*, Poznan, Poland.

SAP AG (1997), ed 3.1, *WF SAP Business Workflow*, Walldorf, Germany.

Schael, T. (1998), *Workflow Management Systems For Process Organisations*, Lecture Notes in Computer Science, Vol. 1096, Springer-Verlag, Berlin.

Staffware plc (1999), *Staffware GWD Procedure Definer's Guide: Issue 2*, Maidenhead, Berkshire.

Stein, L. A. (1999), 'Challenging the Computational Metaphor: Implications for How We Think', *Cybernetics and Systems,* 30(6).

Strong, D. M. and Miller, S. M. (1995), 'Exceptions and Exception Handling in Computerised Information Processes', *ACM Transactions on Information Systems,* Vol. 13(2), pp. 206-233.

Symon, G., Long, K. and Ellis, J. (1996), 'The Coordination of Work Activities, Cooperation and Conflict in a Hospital Context', *Computer Supported Cooperative Work,* Vol. 5(1), pp. 1-31.

Turner, P., Turner, S. and Horton, J. (1999), 'Towards an Activity-Based Approach to Requirements Definition', http://www.comp.lancs.ac.uk/computing/research/cseg/projects/coherence/%workshop/Turner.html.

Vygotsky, L. S. (1925), 'Consciousness as a Problem in the Psychology of Behaviour' chapter in *Collective Works: Questions of the Theory and History of Psychology,* Pedagogika, Moscow.

Wertsch, J. (ed) (1981), *The Concept of Activity in Soviet Psychology,* Sharpe, Armonk, NY.

Workflow Management Coalition (2002), 'Introduction to Workflow', http://www.wfmc.org/introduction_to_workflow.pdf.