AN ADAPTIVE P2P WORKFLOW MANAGEMENT SYSTEM
Flexibility and Exception Handling Support in P2P Based Workflow

A. Aldeeb, K. Crockett and M. J. Stanton
Department of Computing and Mathmatics,
Manchester Metropolitan University, Chester Street, Manchester, M1 5GD, UK

Keywords: Workflow management system, instance level adaptation, workflow flexibility, P2P, exception handling.

Abstract: Workflow processes are moving from long-lasting, well-defined, centralised business processes to dynamically changing, distributed business processes with many variants. Existing research concentrates on decentralisation and on adaptability but there is more to be done on adaptability in decentralised workflow systems. The aim of this research is to overcome the limitation of current workflow management systems by moving from a centralised workflow to a flexible decentralised peer-to-peer (P2P) workflow system. A P2P workflow management architecture is proposed which offers flexibility, exception handling and dynamic changes to both the workflow process definition and process instance level by applying a range of Artificial Intelligence (AI) techniques. An Exception Handling Peer (EHP) captures exceptions, from the workflow peers, characterises the exceptions and applies a recovery policy. Initial prototyping of the system has been carried out using JBoss jBPM whilst the P2P network environment of this prototype is based on Sun MicroSystem's JXTA.

1 INTRODUCTION

Workflow is the automation of a business process, in which documents, information or tasks are passed from one participant to another for action, according to a set of procedural rules (Fischer, 2002). A workflow management system (WFMS) is a software tool for defining, instantiating, executing and monitoring workflows.

In the literature, most workflow systems are based on centralised client/server architecture (Bauer, Reichert and Dadam, 2003). This requires a centralised database to store the workflow process definition and a centralised workflow engine to manage activities such as coordination and monitoring process execution (Yan, 2004). The main disadvantages of any such architecture are the potential bottleneck that can arise during process execution, and that the central database can become single point of failure. Workflow processes are moving from long-lasting, well-defined, centralised business processes to dynamically changing, distributed business processes with many variants (Aalst and Basten, 2002; Sadiq and Orlowska, 2005; Bauer, Reichert and Dadam, 2003). Workflow applications are inherently distributed. They involve people, resources and tools that may be distributed over a wide geographic area (Aalst and Basten, 2002; Aalst and Hee, 2002). In a peer-to-peer (P2P) architecture each workstation has equivalent capabilities and responsibilities. Recent research has shown an increased interest in P2P based workflow (Fakas and Karakostas, 2004; Yan, 2004). However, research on implementing workflow in a peer-to-peer environment is still at the conceptual stage. In addition, the research in P2P has tended to focus on tasks coordination rather than on flexibility and exception handling in workflow systems.

The aim of this research is to overcome the limitation of the current workflow management systems by moving from a centralised workflow to a flexible decentralised P2P workflow system. It proposes a P2P workflow management system which offers flexibility, exception handling and dynamic changes to both the workflow process definition and process instance level by applying a range of AI techniques. The rest of the paper is organised as follows. A brief description of the literature and related work in decentralised workflow and flexibility concepts is presented in Section 2. In Section 3, an adaptive P2P workflow is introduced with high level architecture diagrams

Aldeeb A., Crockett K. and J. Stanton M. (2007).
AN ADAPTIVE P2P WORKFLOW MANAGEMENT SYSTEM - Flexibility and Exception Handling Support in P2P Based Workflow.
In *Proceedings of the Ninth International Conference on Enterprise Information Systems - ISAS*, pages 428-433
Copyright © SciTePress

describing its different components. In Section 4, an example is presented to help illustrate the build-time and run-time functions of the system and the current state of implementation. Finally, conclusion and future work are discussed in Section 5.

2 LITERATURE REVIEW

2.1 The Workflow Reference Model

A workflow reference model was released by the Workflow Management Coalition in October 1994 (http://www.wfmc.org/standards/referencemodel.htm). This reference model consists of central workflow enactment services and five components: process definition tools, administration and monitoring, workflow client applications, invoked applications and other workflow enactment services. In addition, the reference model defines five interfaces connecting these components with the central enactment services (Plesums, 2005). The majority of deployable workflow systems are based on, and compatible with, this reference model (Yan, 2004; Buhler and Vidal, 2005).

2.2 Decentralised Workflow Systems

Exotica/FMQM (Alonso et al 1995) is a novel, distributed workflow architecture, which does not build on a centralized database. Instead, persistent messages are used to store and exchange the information between autonomous nodes in order to enact a workflow process. Each node works independently, and interacts with other nodes through the persistent messages notifying them that a step of the process has been completed.

Once a process has been defined, its definition is compiled to determine the information relevant to each node, binding activities to the nodes where they will be performed. Since there is no centralized server, the problems of centralisation are removed. Node failures will stop the execution of process instances that use that node, but will not prevent other process instances from being executed at other nodes.

This was one of the earliest decentralised workflow systems, which concentrated on data management rather than exception handling or dynamic changes in workflows.

Another decentralised workflow architecture, SwinDew (Yan, 2004), applies P2P concepts to workflow scenarios and decentralises both data and control. In this approach, each system node acts as a client application and performs both data repository and workflow engine functions. SwinDew focus's on workflow administration and management issues without detailed investigation of exceptions handling or of dynamic changes to process instances or schema.

Fakas and Karakostas (2004) designed a P2P workflow management system facilitating the distributed definition, execution and management of workflows. The architecture of this system is based on the concepts of the Web Workflow Peer (WWP). A WWP is a proposed processing entity that facilitates users to participate, administrate and manage workflow process. A peer user may participate as an administrator or participating Peer in a particular workflow instance. P2P and the centralised approaches have been compared in terms of message exchanges and sizes. It was found that, whilst there is a requirement for additional notification messages in this architecture, the P2P has a significant advantage when activities are associated with large messages.

2.3 Workflow Flexibility and Adaptation

Exceptions that occur during workflow execution have been divided into: basic failures, application failures, expected exceptions and unexpected exceptions (Casati, 1998). Basic failure is related to failures at system level (e.g., DBMS, operating systems, or network failure). Application failure corresponds to failures of any applications invoked by the WFMS. Expected exceptions correspond to predictable deviations from the normal behaviour of a process. Unexpected exceptions occur when there are inconsistencies between the workflow and its corresponding real world business process.

Flexibility in workflows may be achieved by either selection or by adaptation (Halliday, 2001). Flexibility by selection will only handle exceptions where they can be predicted before system development. Time, cost and knowledge limitations will prevent successful development of systems that rely on flexibility by selection alone. Flexibility by adaptation involves altering the workflow process to include one or more new execution paths and can be subdivided into evolutionary adaptation and instance adaptation. Evolutionary adaptation is a result of evolutionary change initiated by business managers to improve efficiency or responsiveness, or is forced by legislature or changing market demands (Aalst and Basten, 2002). Evolutionary adaptation affects new instances but it may also influence old instances.

Instance adaptation is related to ad-hoc change which may occur at only a specific instance as a result of an error, a rare event, or special demands of the user. This type of change will create a variant of workflow process. Instance adaptation involves altering one or more running instances of a given schema (Halliday, 2001). Inheritance concepts can offer some support for ad-hoc change (Aalst and Basten, 2002) where the predefined workflow process definition is the parent class and the modified workflow process definition (variant) resulting from an ad-hoc change would be a child class.

Current workflow management systems lack support for runtime adaptability (Rinderle, Reichert and Dadam, 2004a; Muller and Greiner and Rahm, 2004; Kammar, 2000; Chung, 2003; Divitini and Simne, 2000; Rinderle, Reichert and Dadam, 2004b). Flexibility and exception handling approaches are based on the centralised WFMS. Much work has been done on decentralisation, and on flexibility and adaptability but these need to be considered for decentralised workflow systems.

The research presented in this paper applies the flexibility, adaptability and exception handling concepts to a decentralised P2P based WFMS. An adaptive decentralised P2P WFMS architecture is proposed, which will allow for flexibility by selection at design time, and provide both evolutionary and instance adaptation at run-time. Additionally, due to its modular structure, the architecture will support experimentation and analysis of various techniques for the detection of exceptions at instance level. It will then be possible to use information about past exceptions to determine whether there is a need for the process to evolve.

3 AN ADAPTIVE P2P WFMS

An adaptive P2P WFMS that can handle exceptions at both schema and instance level in a dynamic environment is described in Figure 1. The proposed Adaptive P2P WFMS is inspired by current research on the administration and management of P2P based workflow systems (Yan, 2004; Fakas, 2004; Coon, 2002). Peers join “virtual communities” according to their capabilities and discover each other using the services provided by an open P2P network. The coordination is performed by notification messages exchanged between peers. In addition to this, in the architecture proposed here, exceptions are handled by a dedicated exception handling peer.

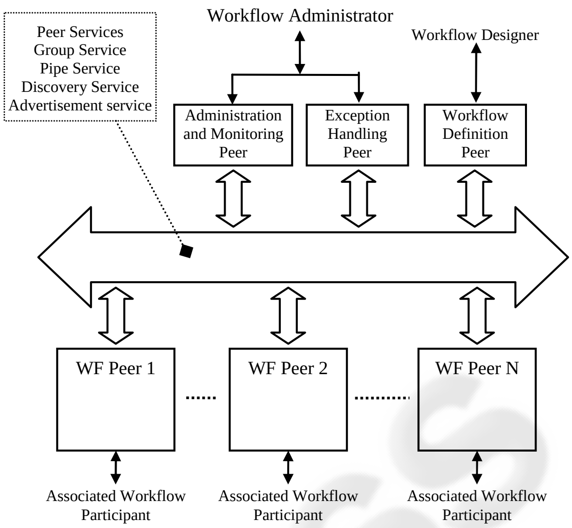

Figure 1: The adaptive P2P workflow system.

Two functions are conducted by this adaptive P2P WFMS: A Build-time function, which includes workflow process modelling, storing process definitions and distributing the process to workflow peers, and a Run-time function, which includes process instance creation, task coordination and exception handling procedures.

As shown in figure 1, the P2P network provides services that include advertisement services, group services, peer services, pipe services, and discovery services. The main components of the architecture are described in the following sections.

3.1 Workflow Peer (WFP)

The WFP can reside on any machine on the P2P network enabling direct communication with other workflow peers to enact the workflow process. The internal structure of the WFP used here is the same as that described in (Yan, 2004), consisting of user, task, and flow components. It also maintains four data repositories – a peer repository, a resource and tools repository, a task repository and a process repository. Each WFP is associated with a workflow participant and each performs a part of the workflow. Once the task is completed, the WFP informs its successor and the next task of the process may be executed.

Process co-ordination is achieved by the exchange of messages between peers. Each message will be one of three types: information, control, or exception. Information and control messages are used for data and control transfer between peers. The exception message is used in case of an exception appearance. Each peer is provided with expected-exception handling facilities. However, if a peer faces unexpected exception, this will require the intervention of the exception handling peer to initiate an exception handling routine.

3.2 Workflow Definition Peer (WFDP)

The function of this node is the design and the storage of the whole workflow schema at build-time. The workflow process is partitioned to separate tasks according to the workflow participants and the organisational structure. These tasks are then distributed to the corresponding workflow peers. Figure 2 shows the internal structure of the WFDP.

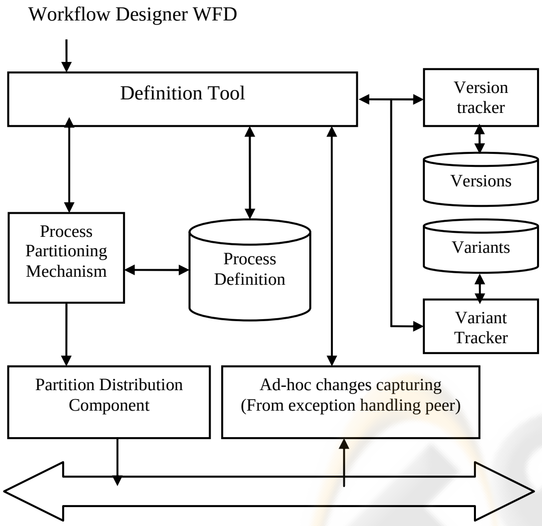

Figure 2: The workflow definition peer structure.

A workflow process definition resulting from an evolutionary change is called a "version" of the workflow process. New cases are handled according to the most recent version of a process. The current version of the process is that determined as the most likely to fit the incoming case. A workflow process definition resulting from an ad-hoc change which affects the running workflow instances is called variant of the workflow process (Aalst and Basten, 2002). Repeated similar variants may lead to evolutionary change of the workflow process. The WFDP holds all versions and variants of the workflow process. To reduce storage costs and simplify process management it may be necessary to keep the number of active versions to minimum. Evolutionary and ad-hoc changes may affect only particular tasks in the workflow process and their associated WF peers.

3.3 Exception Handling Peer (EHP)

The Exception Handling Peer captures exceptions from the workflow peers, characterises the exceptions and applies a recovery policy. The structure of the EHP is shown in figure 3. It will handle exceptions at the instance level.

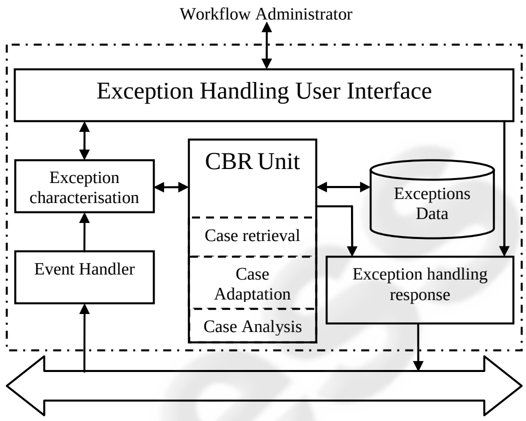

Figure 3: The workflow exception handling peer.

Once the EHP receives an exception message from a workflow peer, a direct connection between both will be established. The exception will be resolved by proper exception handler candidates, such as retry, recovery, compensation etc. At instance level, exceptions can be single task exception or multi task exception. Also, at instance level, if the exception cannot be resolved by the EHP, a new variant for the exception raising case may be created. This can be achieved by collaboration between the EHP and the workflow definition peer where the new variant will be stored.

In case of process evolution, the current running cases will be tackled either with a proceeding approach or with a transferring approach. In the proceeding approach, each case is referred to a specific version of workflow process. Newer versions do not affect old cases and the number of versions should be kept to minimum. In the transferring approach, existing cases are transferred to the new process so they can directly benefit from evolutionary changes. The EHP can acquire some knowledge from previous exceptions, which may be achieved by applying AI techniques. Initially Case Based reasoning will be applied to this problem.

The EHP is provided with a Case Based Reasoning (CBR) unit to handle exceptions which need to be managed in similar way, but may occur in different instances. This consideration suggests that previous experience in exception handling can be reused for different workflows. This can be achieved by using CBR in exceptional problem solving. CBR is an artificial intelligent technique which can be defined as the process of solving new problems based on the solutions of similar past problems.

Luo, et al. (2003) describe a workflow system which uses CBR scheme to derive patterns from exception handling. Then, the similarity is determined to identify the nearest pattern in the knowledge base to the current exception and applying the appropriate action for this case. User intervention is allowed to handle new cases or defining a new action for a specific case. CBR added great value to the exception handling in workflow as the CBR system collects more cases; the WFMS becomes more resistant because it has a large set of knowledge to handle future exceptions (Cardoso et al., 2001). Luo et al. (2003) use CBR for exception handling in cross-organisational workflow where the coordination is among different workflow management systems in different organisations. However, in P2P based workflow the coordination is among nodes within the same organisation. One goal of this research is to apply CBR in a P2P based workflow management system at instance level using an exception handling peer. The use of CBR includes: case retrieval, case adaptation, case analysis and verification and case reuse.

4 A CASE STUDY

To better illustrate how the proposed adaptive P2P workflow management system described in the previous section works, an example of a motor insurance claim process is presented. The process consists of 8 tasks as shown in Figure 4. The tasks will be distributed over the workflow peers and their workflow participants, based on the roles of the participants themselves and structure of the organisation. To examine the system, both build-time and run-time functions are implemented. Build-time function implementation includes:

(1) Modelling the process using a graphical notation using the Workflow Definition Peer (Figure 4).
(2) Partitioning the process into tasks according to the roles of the workflow participants and the organisational structure of the insurance company.
(3) Creation of Workflow Peers corresponding to the model of task distribution.
(4) Distribution and initialisation of tasks to the relevant Workflow Peers using the P2P network.

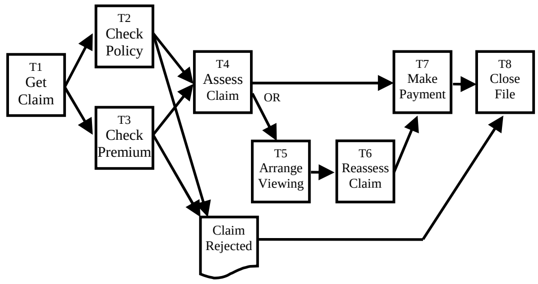

Figure 4: Example insurance claim process.

After the Build-Time functions are completed, the system can carry out its Run-time functions. These will include:

(1) Instantiating a workflow instance and coordination of tasks by message exchange between peers.
(2) Making ad-hoc changes to the running instances and examining the exception handling procedures.
(3) Handling the exception raising instance, and generating a process variant.
(4) Evolving the workflow process and establishing a versioning mechanism for the current and old instances.

Initial prototyping of the system has been carried out using JBoss jBPM. XPDL (XML Process Definition Languages) is currently being used for process definition as it offers portability between different Process Design tools. The P2P network environment of this prototype is based on Sun MicroSystem’s JXTA (http://www.jxta.org/).

5 CONCLUSIONS

This paper presented an adaptive P2P workflow management system. The main components of the architecture of this system are: workflow definition peer, workflow peers, administration peer, exception handling peer and a P2P networking infrastructure. The proposed system will, at run-time, work in two modes; Normal mode and Exception handling mode. In normal mode the process enactment is performed by the direct communication and collaboration amongst peers. However, in the exception handling mode, the exception handling peer will control and synchronise between the workflow peers affected by the exception in order to resolve the problem. The latter case makes the system work in Hybrid Centralised/P2P environment with some level of control from a single node.

This adaptive P2P approach provides more flexibility in dealing with exceptions at instance level using an intelligent exception handling peer. The exception handling peer can acquire knowledge from previous experience to deal with new exceptions using case based reasoning. This peer interacts with the workflow peers individually, so only the peer which is affected by the current exception or ad-hoc change will be suspended. Currently work has been carried out developing a single Workflow Peer, using JBoss jBPM. JXTA has been identified as a good candidate for the P2P network protocol and work will now concentrate on distributing the tasks among the Workflow Peers. Further evaluation of the architecture, consisting of both research and experimental analysis, will include: completion of the underlying P2P workflow architecture and its application to a number of workflow problems; development and evaluation of a framework for handling various exceptions at instance level; developing a CBR module for the Exception Handling Peer.

REFERENCES

Aalst, W. and Basten, T. (2002) Inheritance of Workflows: an Approach to Tackling Problems Related to Change. *Theoretical Computer Science,* 270, 125-203.

Aalst, W. and Hee, K. (2002) *Workflow Management: Models, Methods, and Systems.* London: MIT Press.

Alonso, G. et al. (1995). Exotica/FMQM: A Persistent Message-Based Architecture for Distributed Workflow Management. *In* International Federation for Information Processing (1995) Proceedings: Working Conference on Information Systems for Decentralised Organizations, Trondheim.

Bauer, T., Reichert, M. and Dadam, P. (2003) Intra-subnet Load Balancing in Distributed Workflow Management Systems. *International Journal of Cooperative Information Systems, 12(3), 295-323.*

Buhler, P. and Vidal, J. (2005) Towards Adaptive Workflow Enactment Using Multiagent Systems. *Information Technology and Management, 6, 61-87.*

Cardoso, J. et al. (2001) Survivability Architecture for Workflow Management Systems. *Technical Report,* University of Georgia, USA.

Casati, F. (1998) *Models, Semantics, and Formal Methods for the Design of Workflows and Their Exceptions.* PhD Thesis, Milan. Italy.

Chung, P. et al. (2003) Knowledge-based Process Management- an Approach to Handling Adaptive Workflow. *Knowledge-Based Systems,* 16, 149-160.

Coon, M. (2002) Peer-to-Peer Workflow collaboration: White paper, viewed: 5/11/06 http://www.proteus-technologies.com/cmm/docs/P2P_Workflow_Whitepaper.doc.

Divitini, M. and Simone, C. (2000) Supporting Different Dimensions of Adaptability in Workflow Modelling. *Computer Supported Cooperative Work,* 9, 365-397.

Fakas, G. and Karakostas, B. (2004) A peer to peer (P2P) Architecture for Dynamic Workflow Management. *Information and Software Technology,* 46, 423-431.

Fischer, L. (ed.) (2002) Workflow Handbook 2002. Florida: Future Strategies.

Halliday, J. and Shrivastava, S. and Wheate, S. (2001) *Flexible Workflow Management in the OPENflow system. In* Fifth IEEE International Enterprise Distributed Object Computing Conference, Seattle, Washington, USA.

JBoss jBPM, open source Middleware software, viewed 10/11/2006 <http://www.jboss.com/products/jbpm>.

JXTA™ technology website, viewed 10/11/2006 <http://www.jxta.org/>.

Kammar, P. et al.(2000) Techniques For Support Dynamic and Adaptive Workflow. *Computer Supported Cooperative Work,* 9, 269-292.

Luo Z. et al. (2003) Exception Handling for Conflict Resolution in Cross-Organizational Workflows. *Distributed and Parallel Databases,* 13, 271-306.

Muller, R., Greiner, U. and Rahm, E. (2004) Agent Work: a Workflow System Supporting Rule-based Workflow Adaptation. *Data & Knowledge Engineering,* 51, 233-256.

Plesums, C. (2005) *Workflow in the World of BPM, Are They the Same? In* Fischer, L. (ed.) Workflow Handbook 2005. Florida: Future Strategies, p. 20.

Rinderle, S., Reichert, M. and Dadam, P. (2004) Flexible Support of Team Processes by Adaptive Workflow Systems. *Distributed and Parallel Database,* 16, 91-116.

Rinderle, S., Reichert, M. and Dadam, P. (2004) Correctness Criteria for Dynamic Changes in Workflow Systems – a Survey. *Data & Knowledge Engineering,* 50, 9-34.

Sadiq, S. Orlowska, M. and Sadiq, W. (2005) Specification and Validation of Process Constraints for Flexible Workflows. *Information Systems,* 30, 349-378.

The Workflow Management Coalition Website, viewed 10/11/2006 <http://www.wfmc.org/>.

Yan, J. (2004). *A Framework and Coordination Technologies for Peer-to-peer based Decentralised Workflow System.* PhD Thesis, Swinburne University of Technology, Australia.