December 2002

INTELLIGENT AGENT-BASED FLEXIBLE WORKFLOW MANAGEMENT FOR CRISIS RESOLVING

Dongming Xu
City University of Hong Kong

Minglu Li
Shanghai Jiao Tong University

Huaiqing Wang
City University of Hong Kong

INTELLIGENT AGENT-BASED FLEXIBLE WORKFLOW MANAGEMENT FOR CRISIS RESOLVING

Dongming Xu
Department of Information Systems
City University of Hong Kong
isxu@is.cityu.edu.hk

Minglu Li
Department of Computer Science & Engineering
Shanghai Jiao Tong University
li-ml@cs.sjtu.edu.cn

Huaiqing Wang
Department of Information Systems
City University of Hong Kong
iswang@is.cityu.edu.hk

Abstract

The unpredictability of business processes require workflow systems to support crisis resolving in the dynamical environment. Traditional approaches to handle this problem have fallen short, providing little support for changes, particularly when the execution of a process has started. Further more, crises vary widely in their characteristics and significance, challenging the application of any single approach to resolving them. In this paper, we briefly discuss the classification of such crises, highlighting differing impacts on the workflow model. Based on this discussion, we define our main goals to address in the development of flexible workflow support, including strategies for avoiding crisis, detecting them when they occur, and resolving them at various levels of impact. We then proposed a novel intelligent agent-based architecture to supporting these goals within the design of a workflow system infrastructure, by utilizing crisis manger agents, crisis monitor agents, crisis backup agents and crisis mender agents, and integration with an intelligent agent-assisted decision support system.

Keywords: Multiagent systems, software agents, flexible workflow management, workflow management systems, emerging applications

Introduction

The occurrence of exceptions is a fundamental part of organizational processes (Suchman , 1983). In order for workflow systems to support these exceptions handling processes, they must be able to support the handling of these inconsistencies and adapt to changes over time (Grudin, 1994). Workflow process exceptions can be frequent and extremely disruptive (Saastamoinen, 1995). Traditional approaches have utilized inflexible, formal control policies that make reactive control and graceful exception handling difficult, if not impossible, tasks (Kammer et al, 2000; Alonso et al, 2000).

Crises may appear in virtually all parts of government and business. They range from large scale crises, e.g., natural/economic disasters, military tensions and contentions, and epidemic outbreaks, to highly localized crises, like simple accidents. The principal characteristic of a crisis is that it occurs unexpectedly and that its exact course is unknown and unpredictable (Georgakopoulos et al, 2000). Crises are defined as situations with a high degree of threat to important values, and a high degree of time pressure (Nunamaker et al, 1988). A crisis management system is a specific decision support system (DSS) with emphasis on reducing the time required to make decisions, while maintaining decision quality (Smith and Hayne, 1991).

Crises impact a workflow model at varying levels of significance. Some crises cause only minor perturbations to the work process. Others may affect only the current running instance of the workflow. The most significant require the process model itself to evolve to accommodate changes that have occurred in the environment. These different classes of crisis require different approaches to support their resolving and recovery as well as evolution within the workflow system.

To support these various goals, the research presented in this paper is an architecture framework for intelligent agent-based flexible workflow management systems (WfMS) (Wang, 2001). Our architecture is modeled in the spirit of an intelligent agent-assisted decision support system (IADSS) (Wang, 1997). The proposed architecture contains a set of crisis monitors gathering information to determine the cause of the crisis, a set of crisis backups substituting the run-down parts to keep the productivity of the WfMS, the crisis menders that deals with the spreading of the crisis, and the crisis managers coordinates appropriate containment with the current reports from the crisis monitors, the crisis backups and the crisis menders, support by the crisis decision system based on the IADSS. Furthermore, we discuss a case study of the typical power crisis resolving equipments – uninterruptible power supplies (UPS) under the proposed architecture. A prototype of the proposed architecture has been built using Java and XML technologies for simulations of the UPS to demonstrate the technical feasibility of our architecture and serve as the basis for future discussion and evaluation.

General Goals for Flexible Workflow

Detecting Crises

The discussion of handling crises tends to proceed with the assumption that the anomaly has already been detected and fully described. A workflow system needs to support the discovery of crises in a timely and useful manner. While some crises will be obvious to users, others will be more subtle. The initial error may not be the one that eventually derails the workflow. Data constraints or general rules about workflow progress may be used to provide forewarning of problems occuring, giving clearer indications of the source of the crisis. Active data stores and event driven architectures may provide alerts when constraints are violated or inconsistencies arise. Multiple appropriate views of workflow execution can provide visualizations of where problems may be occurring. Reflexive process agents can provide analysis and feedback on workflow execution as it progresses.

Avoiding Crises

A workflow support system, integrated into an existing work environment, may itself be the cause of critical conditions. If a system is rigid in its use, involves significant adoption cost, or integrates poorly with existing tools and approaches, participants are likely to operate outside the system. Ownership of work procedures may be divided among subunits (or even across organizations). Open systems, support for incremental adoption, flexible execution approaches, reusable process components, and integrated support for communication between participants, may all help avoid crises.

Customized agents and event monitoring infrastructure are useful for gauging the effectiveness of these workflow components. Further, when a new process is introduced to a work environment and culture, there is often some pushback to its adoption. It is important for the process to be able to adapt in response to this pushback. In addition, process discovery tools are useful for comparing and validating the workflow model with the actual work being accomplished. Wide divergences can indicate technology mismatches or inapplicability and thus the need for evolution and optimization of the process.

Resolving Crises

Tolerating Minor Deviations

Noisy crisis in workflow execution describes minor deviations from the normal process that are not significant enough to require changes in the execution process model. Noisy crisis may be either accommodated within the tolerance of the model, or may produce consequences which can be ignored without unacceptable harm to the continued execution of the process.

An overly rigid workflow description or one that requires over-specification of work activities results in workflow specifications that may be fragile, not accommodating reasonable minor deviations from the "ideal" process. Even requiring complete specification of process and resource dependencies prior to execution may contribute to this problem. While, the exact character of deviations from the main process may not be known, it may be possible to determine locations where added flexibility is required and permissible. Appropriate support for abstraction, flexibility in degree of specification, and a flexible execution model can provide some tolerance for minor deviations.

Handling Changes to the Process Instance

Buffering crises are particular to a workflow instance or collection of instances. For example, if a regular participant is unavailable (e.g. sick or on vacation) then changes might be required to accommodate the absence. These would not be reflected in the model process, however, because the individual would resume the same role upon returning. This sort of change requires the ability to modify individual instances of the execution process without altering the overarching process type. A dynamic instance model of the workflow, created or modified "on-the-fly" at run-time, helps accommodate this sort of change.

An alternative approach to addressing this issue is not to specify a model with precise ordering of activities, but rather to provide constraints and present the user with multiple available tasks to perform.

Evolution and Optimization of the Process Model

Evolution of process objects and workflows occurs over time as a result of changing tasks, priorities, responsibilities, and even people. Optimization occurs when the improvement of a previous work model results in a better way of doing things by adding, removing, or redefining process activities and their constraints. As a workflow process is repeatedly executed, small changes may be made each time through. Eventually, a process may converge into a common practice and become institutionalized.

Unfortunately, a common practice and a best practice may not be the same thing. In order to determine this, metrics must be kept to evaluate one execution from another. This evaluation is subjective because the criteria underlying the metrics changes the same way the workflow does. Successful workflows, like successful software, will be applied in new situations that may have been unanticipated by the original creator. Unsuccessful workflows will be abandoned or changed. It is important that the workflow infrastructure allow change to occur both before and after deployment of the workflow, by both technical and non-technical participants. Some optimizations may even be performed by the system itself through agents, validation tools, or optimizers.

Intelligent Agent-Based Architecture

As we have pointed out in our introduction, there exist numerous obstacles that remain to be overcome in today's WfMSs to fully achieve the vision of intelligent agent-based flexible WfMS. The integration of intelligent agents with WfMSs will be able to address most, if not all, of the articulated issues.

The Proposed Architecture

To detect, avoid, and resolve crises for flexible workflow, we propose an intelligent agent-based architecture, which is shown in Figure 1.

A multi-agent-based flexible workflow system is composed of two collaborative parts: a WfMS for general purposes and an extended intelligent crisis solution system.

General purpose WfMS: To support atop developing flexibility, we consider that the commercial workflow products are our first selection. There are more than 100 kinds of workflow products in the market, such as IBM's FlowMark/MQSeries Workflow, Action Tech's Action Metro, Oracle's Oracle Workflow, HP's Changengine, InConcert's InConcert, FileNET's Visual WorkFlo, Forte's Conductor, and JetForm's JetForm et al. While Georgakopoulos et al developed the CMI system supporting workflows in crisis mitigation situations, they deployed IBM FlowMark version 2.3 and MCC's Composite Event Detection and Monitoring System (CEDMOS) event processing system (Georgakopoulos et al, 2000). Casati and Pozzi (1999) exemplify the introduced concepts by showing how expected exceptions can be mapped on top of Changengine, Hewlett-Packard's WfMS.

Intelligent Crisis Solution System (ICSS): ICSS is composed of four classes of agents: Crisis Monitors, Crisis Backups, Crisis Menders and Crisis Managers, and an intelligent agent-assisted DSS supporting crisis decision making.

Intelligent Agent-assisted DSS (IADSS): IADSS is our proposed novel architecture to support the co-operate decision process by utilizing event-driven and task-driven data mining agents, along with user assistant agents and a knowledge manager agent (Wang, 1997). In this paper, IADSS aids the crisis managers to making decisions in resolving crises.

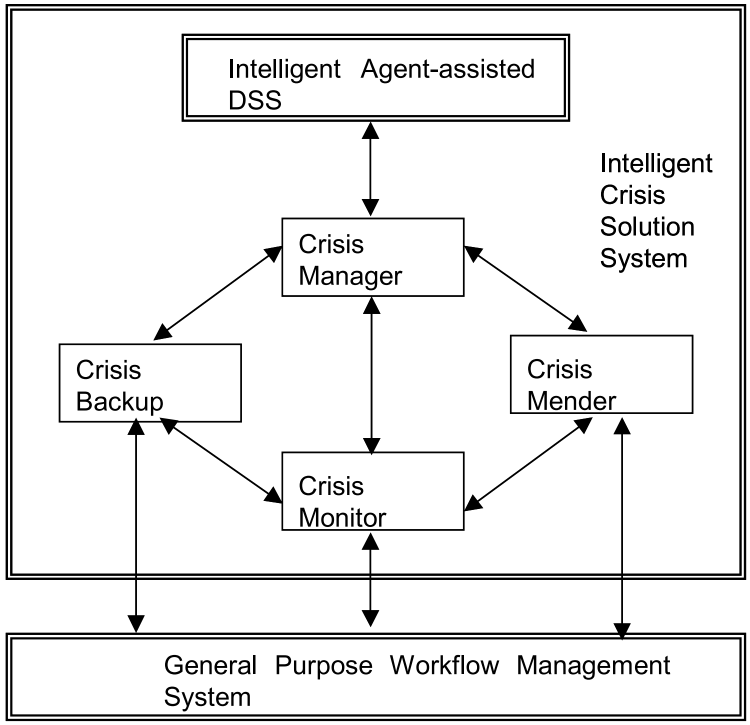

Figure 1. A Multi-Agent-Based Flexible WfMS

Crisis Monitors

The role of crisis monitors in the proposed architecture is monitoring the current running states of the underlying WfMS. The architecture of a crisis monitor is shown in Figure 2.

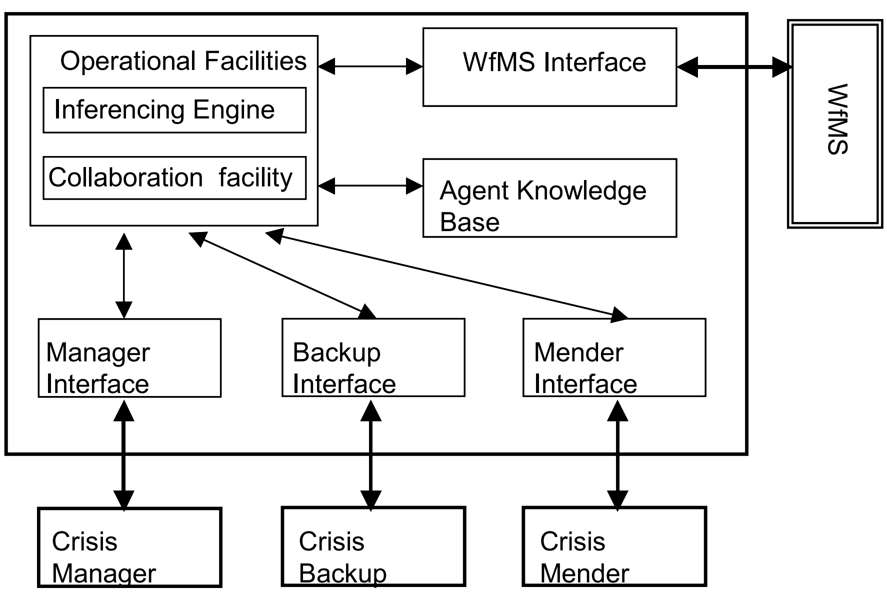

Figure 2. The Architecture of a Crisis Monitor

Crisis Backups

Crisis backups are backups supporting those very important agents in the WfMS. Their life cycle is a sleep-work-sleep-work loop. Once a very important agent crashes in the WfMS, the crisis monitor or the crisis manager will start a crisis backup to replace the work of the bad agent. On the other hand, the running crisis agent will stop if the bad agent is recovered. Figure 3 shows the architecture of a crisis backup, which is composed of five components.

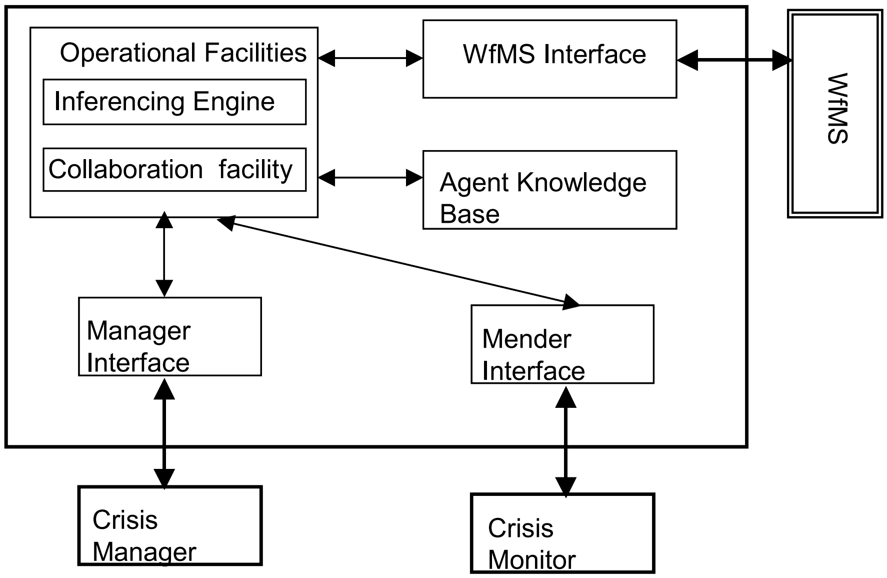

Figure 3. The Architecture of a Crisis Backup

Crisis Menders

For expected crises, such as stop power or water supply, the crisis mender may finish the treatment as needed. However, for unexpected crises, such as earthquake, epidemic and tornadoes, the degree of recovery that the crisis menders are able to achieve may be limited. Meanwhile, the crisis menders evolve the workflows in the WfMS.

As Figure 4 showed, a crisis mender contains five components. Informally, the functionality of the crisis menders is similar as firemen. They will act immediately if any fire, and withdraw quickly if no fire. On the other hand, the crisis menders repair the bad things like repairmen, or treat the patients like doctors, and so on. In the proposed architecture, the crisis monitors or the crisis managers can start the crisis menders in crisis resolving situations. During the treatment of the crisis menders, they report the recent evolutions to the crisis managers. According to the current reports, the crisis managers can order new crisis menders joining into the mending process, or require the support of the higher-up crisis managers in the inter-organizational workflows. The crisis managers or the crisis monitors decide when to stop the work of the crisis menders.

Crisis Managers

The crisis managers provide management and coordination control functions over all the agents in the proposed architecture. The internal component-wide architecture of the crisis manager is shown in Figure 5, which contains six components: The IADSS interface, the operational facilities, the monitor interface, the backup interface, the mender interface and the agent knowledge base that provides support for localized reasoning. From the functional standpoint, the crisis managers provide the following functionality in the proposed architecture:

- Record the reports from the crisis monitors, the crisis backups and the crisis menders. Then, transfer them to the IADSS for evolution of the repository (Wang, 1997).
- According the reports, make decisions to start the crisis monitors, the crisis backups and the crisis menders in relevant moment.
- Make decisions whether requirement of the support of the higher-up crisis managers in the multi-organizational workflow environment (Georgakopoulos et al, 2000).

- Make decisions whether delegation of the support of the lower level crisis managers in the multi-organizational workflow environment (Georgakopoulos et al, 2000).
- Make decisions when to recover the workflow, and to stop the crisis monitors, the crisis backups and the crisis menders.
- Applying the advanced features of the IADSS, update the domain knowledge of the crisis monitors, the crisis backups and the crisis menders (Wang, 1997).

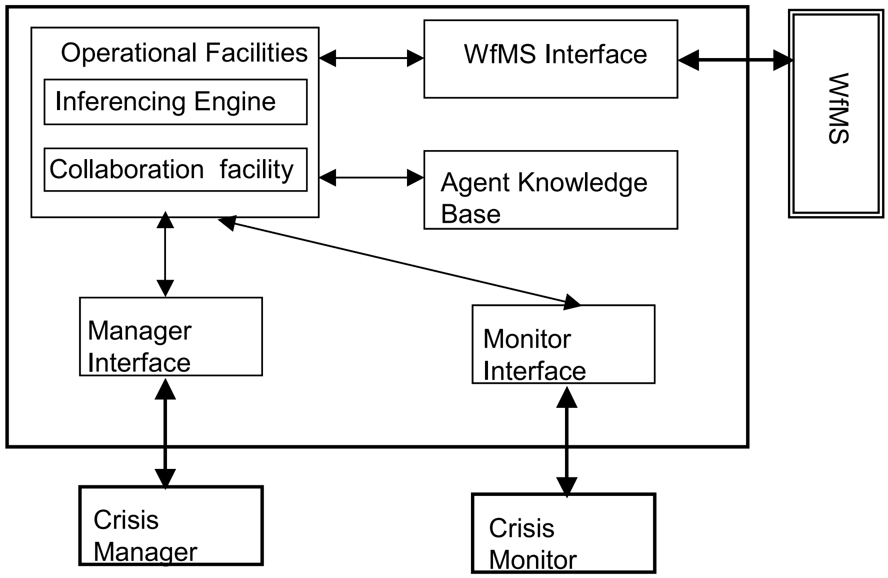

Figure 4. The Architecture of a Crisis Mender

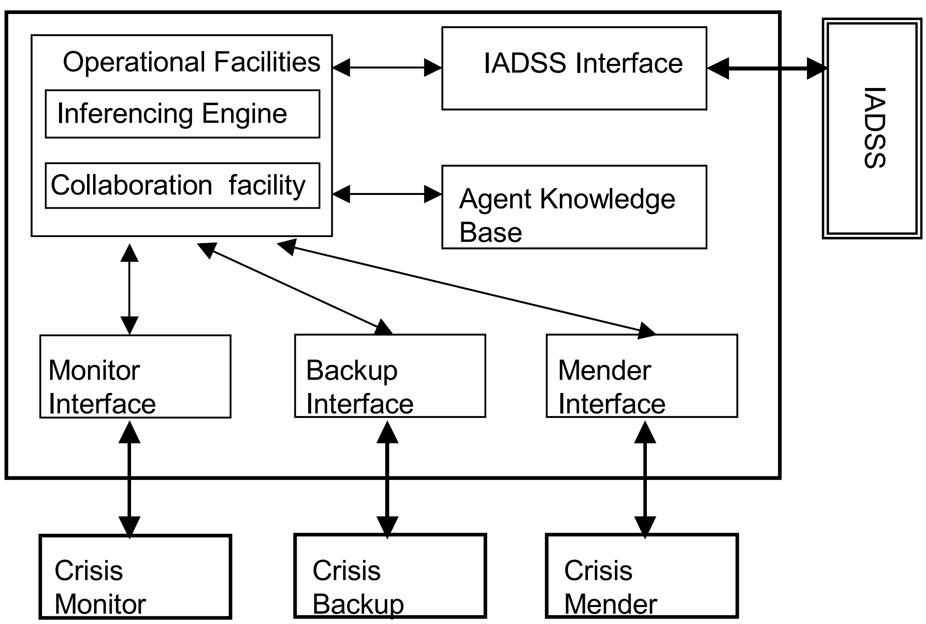

Figure 5. The Architecture of a Crisis Manager

Conclusion

Workflow systems are often called upon to adapt to a changing environment. Crises arising unpredictably often drive the need to adapt workflows. Alternatively, the need to evolve or optimize the process may drive the changes in the process model. Traditional approaches to handling dynamism in workflow have generally fallen short. Some attempts have proven too rigid, not easily tolerating dynamic change once a process starts executing. Others lack the structure to provide a coherent process model, identify crises, guide responses, or manage evolution of processes over time.

In this paper, we indicate that crises vary in their impact on the workflow model. This variance suggests the need for different approaches to support different ways of resolving crises. Overall flexibility, integrated support for dynamic change, appropriate information, and decision making all contribute to the ability to resolve unexpected crises as they develop. Systematic support for evolution can reduce unexpected crises over time. It is our belief that this direction of flexible workflow research, although in its infancy, holds much promise. Our research results will serve as the first step, however crude and minute.

Acknowledgements

The authors would like to recognize the hard work and effort in the design and implementation of the prototype by Dr. Deming Ni.

References

Alonso, G., Hagen, C., Agrawal, D., El Abbadi, A. and Mohan, C. (2000), Enhancing the fault tolerance of workflow management systems, *IEEE Concurrency*, 8(3): 74-81

Casati, F., Castano, S., Fugini, M., Mirbel, I.and Pernici, B. (2000), Using patterns to design rules in workflows, IEEE Transactions on Software Engineering, 26(8): 760-785

Georgakopoulos, D., Schuster, H., Baker, D. and Cichocki, A. (2000), Managing escalation of collaboration processes in crisis mitigation situations, in *Proceedings of 16th International Conference on Data Engineering*, pp. 45-56

Grudin, J. (1994), Groupware and Social Dynamics: Eight Challenges for Developers, *Communications of the ACM*, 37(1): 93-05.

Kammer, P. J., Bolcer, G. A., Taylor, R. N., Hitomi, A. S. and Bergman, M. (2000.), Techniques for Supporting Dynamic and Adaptive Workflow, *Computer Supported Cooperative Work*, 9: 269-92

Kazibwe, Wilson E. and Sendaula, Musoke H. (1993), *Electric Power Quality Control Techniques*, New York: Van Nostrand Reinhold

Nunamaker, J.F., Jr., Weber, E.S., Smith, C.A.P. and Chen, M. (1988), Crisis planning systems: tools for intelligent action, *Proceedings of the Twenty-First Annual Hawaii International Conference on System Sciences, Vol. III. Decision Support and Knowledge Based Systems Track*, pp. 25-34

Saastamoinen, H.T. (1995), Case Study on Exceptions, *Information Technology and People*, 8(4): 48.

Smith, C.A.P. and Hayne, S.C. (1991), A distributed system for crisis management, *Proceedings of the Twenty-Fourth Annual Hawaii International Conference on System Sciences*, vol.3, pp. 72-81

Suchman, L.A. (1983), Office Procedures as Practical Action: Models of Work and System Design, *ACM Transactions on Office Information Systems*, 1(4): 320-28.

Wang, Huaiqing (1997), ntelligent Agent Assisted Decision Support Systems: Integration of Knowledge Discovery, Knowledge Analysis, and Group Decision Support *Expert Systems with Applications*, Vol. 12, No. 3, pp.323-335.

Wang, Huaiqing and Xu, Dongming (2001), ollaborative Multi-agents for Workflow Management *34th Hawaii International Conference on System Science*, January 2001, Hawaii, USA.