Monitoring and controlling of a multi - agent based workflow system

Bastin Tony Roy Savarimuthu, Maryam Purvis, Martin Fleurke

Department of Information Science
University of Otago
PO Box 56, Dunedin, New Zealand

{tonyr,tehrany,mfleurke}@infoscience.otago.ac.nz

Abstract

Business processes are often likely to undergo drastic changes and hence the workflow systems that model, simulate and enact these processes should support the monitoring and controlling of processes. In extension to our previous work on the framework for an adaptive and distributed agent based workflow system JBees, this paper describes the agents that are embedded to our system, that can monitor and control the system based upon the data obtained through simulation. In the monitoring part we focus on the performance indicators such as occupation rate, throughput time and average waiting time of various processes and tasks. We also compare the efficiency of various resources for the similar tasks. The controlling agent continuously looks for the anomalies against the criteria defined by the human manager/user with the data obtained from the system and informs the management agent to initiate appropriate action.

Keywords: Agents, Workflow, Process, Adaptivity, Monitoring, Controlling

1 Introduction

Workflow management systems WfMS (Aalst and Hee 2002, Aalst, Hofstede, Kiepuszewski and Barros 2002, Meilin,Guangxin,Yong and Shangguang 1998) are widely used to manage business processes for their known benefits such as automation, co-ordination and collaboration between entities.

The earlier work described by Fleurke, Ehrler and Purvis (2003) deals with the framework of a distributed network of autonomous software agents that can adapt to the changing circumstances in a workflow management system. The business processes undergo changes in due course of time to accommodate changing environment or the new kind of input. When the processes undergo changes or even in normal scenario, the processes are to be continuously monitored to understand the effect of these changes. This need for continuous process reengineering and increased productivity are the motivation to enhance our system with the monitoring and controlling mechanisms.

JBees, our multi-agent framework uses Coloured Petri net (CPN) by Jensen (1992), as its process definition formalism. One of the advantages of using CPN is that the process model is an executable model. Once a model for a particular business process has been constructed, the model can be simulated using various "what if" scenarios. Usually, the input data that is submitted to the simulation agent is the past data that has been recorded for that particular process and organization. Through the simulation, one can examine the behaviour of the system owing to the modifications to some of the process constraints such as variation in availability of the resources or changes associated with the types of cases that are submitted to the system.

Though there has been some work (Cui, Odgers and Schroeder 1998) on process monitoring and controlling, our work is novel as our agent-enhanced framework provides flexibility to alter process models and other process parameters that are continuously monitored and controlled to achieve better system performance.

The paper is organized as follows. The second section gives an overview of the background information such as our existing multi-agent framework. In the third section we discuss about the monitoring and analysis of the data depicted by graphs, the role of the controlling agent and the merits of our system. We have the conclusions and future directions of our work in section four and the acknowledgements in section five.

2 Background

In this section we explain the background of our work which includes the coloured Petri nets which are used to design the process models, the multi-agent system on which our workflow system has been built, the description of our existing workflow system and the former approaches that were used in monitoring and controlling of workflow systems.

2.1 Coloured Petri Nets

The sound mathematical foundation behind the coloured Petri nets makes it a very useful tool for modelling distributed systems. Petri nets consist of four basic elements. The *tokens* which are typed markers with values, the *places* that are typed locations that can contain zero or more tokens, the *transitions* which represent actions whose occurrence can change the number, locations and value of tokens in one or more of the places connected to them and the *arcs* that connect places and transitions.

2.2 Multi Agent Systems for WfMS

Multi agent systems offer distributed and open platform architecture and hence can support dynamically changing systems. In JBees, the WfMS is partitioned among various interacting agents following the interaction protocols. The model associated with a business process is represented with Coloured Petri net formalism that is executed by a specially designed agent. This agent-based environment facilitates the dynamic incorporation of changed models in the system and there by assist the process re-engineering. Advantages of employing agents include the facilitation of inter and intra organizational co-operation and flexibility in process determination and resource utilization.

2.3 Existing multi-agent framework for workflow system - JBees

JBees, consists of five Opal (Purvis, Cranefield and Nowostawski 2002) agents that provide the functionality to control the workflow (Figure 1). The manager agent provides all functionality the workflow manager needs such as creation and deletion of tasks, roles and process definitions, instantiation of new process instances and creation of resource agents. The process agent executes a process instance. Each resource in the system has its own resource agent. Every resource in the system gets registered to one of the broker agents that allocate the resources to the process. The storage agent manages the persistent data that is needed. The issues that have been addressed by the framework include adaptability, reliability and distribution.

2.4 Former approaches for monitoring and controlling of a workflow system

Monitoring and controlling of workflow systems have been discussed for many years. Few researchers have discussed the issues associated with the monitoring and controlling (Cui, Odgers and Schroeder 1998, Muehlen and Rosemann 2000). When it comes to the agent based monitoring, there has been a proposed system described by Wang and Wang (2002). But the proposed system lacks controlling of the process model using agents and also does not cater to the performance monitoring which is central to any workflow system as described by Aalst and Hee (2002)

3 Extensions to our framework

Monitoring and controlling are two of the vital aspects of any workflow management system that help in continuous improvement of the process and the resource management. Our existing framework has been enhanced with the monitoring and controlling mechanisms so that the various processes and cases can be studied, analysed and the required feedback can be given to the management agent to perform the necessary changes to the process model or to amend appropriate controlling measures.

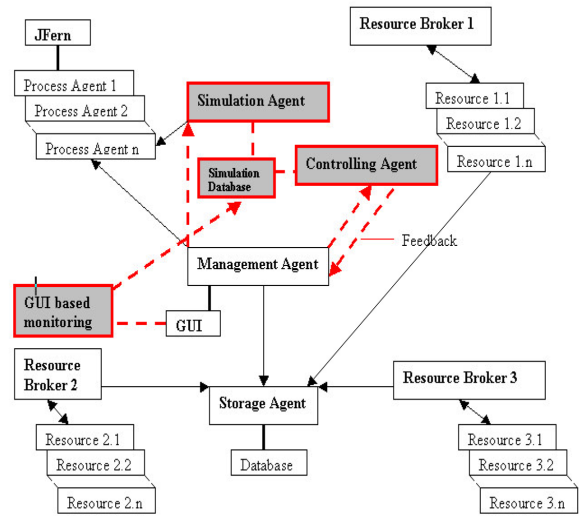


Figure 1: Architecture of the enhanced multi-agent based workflow system.

Figure 1 shows the architecture of our enhanced multi-agent based architecture that incorporates monitoring and controlling mechanisms. The enhancements to the system are represented in dotted lines and solid boxes in figure 1.

3.1 Monitoring the WfMS

Monitoring as an activity is an indispensable part of any WfMS. Every case that is executed in a process model has to be monitored for its various properties such as time taken to complete the process, the various resources employed, time taken by the resources to complete the tasks and waiting times of the jobs in the queue to be served by a resource.

The simulation agent of our WfMS records the following aspects of any process model such as the process id, case id, the various tasks that are performed and various resources that perform these tasks, the starting time of the execution, the ending time of the execution and the waiting time to get a task done if the resource is busy. It also records the timestamps of the resource initializations and the time taken by each resource to complete a particular task.

Our architecture incorporates the modules to examine, analyse and display the properties of the workflow system such as process completion time, arrival rates of cases per unit time, percentage utilization of resources, overall waiting time for any case, average waiting times of all tasks across different cases of any of the processes and the time taken by the resources to do same tasks. For each of the property mentioned above, the system produces a graph. The user of the WfMS can choose any of the properties to analyse the state of that property at any particular point of time.

With the help of the data obtained from simulation, the graphs are drawn which are used for monitoring. We have integrated the JFreeChart, an open source java graphing API with our framework to analyse the data collected by our system.

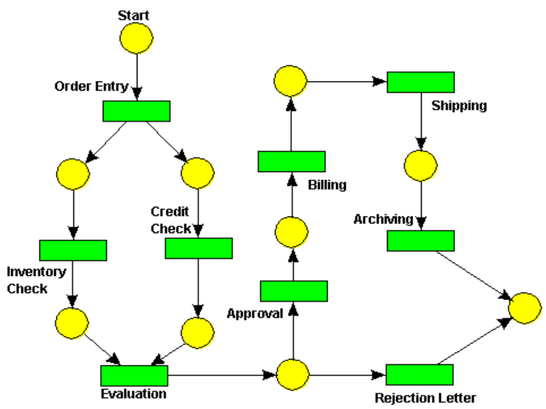


Figure 2: The Order Entry process used for simulation

We use the order entry process shown in figure 2 as the process model under monitoring. The tasks include order entry, inventory check, credit check, evaluation, approval, billing, shipping, archiving and the task associated with writing a rejection letter.

The various tasks of the process model were associated with a set of values for the process parameters such as the number of resources, their availability and their ability (time taken) to do these tasks to simulate various "real life" scenarios. Varying these process parameters we ran a sample numbers of cases and the data was recorded during simulation as shown in figure 3.

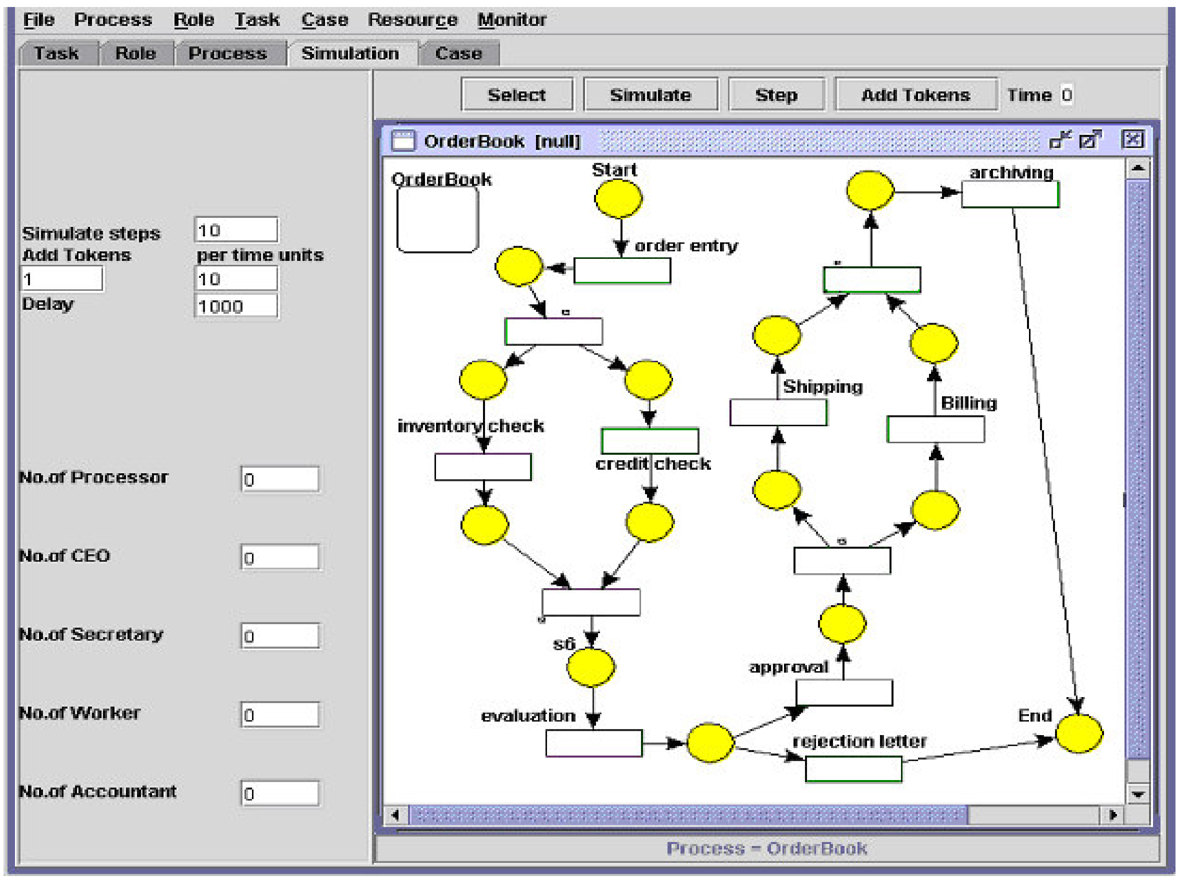


Figure 3: The snapshot of the simulation of the order entry process

3.2 Analysis

The simulation agent is responsible for the computation and storage of data generated during simulation. The simulation of the processes was carried out for the order entry process with the following initial conditions.

The waiting times for the tasks range from 0-10 time units. Also, the approximate time taken by the resources to complete the task and the resources that are capable of doing these tasks can be set in the simulation agent. A sample of these data is given in table 1. The user can choose to monitor any or all the parameters of the simulation. Given below are the analyses of the various kind of monitoring that are possible in our system.

Tasks	Resources capable of finishing the task	Approx. Time taken in (seconds)
Order entry	Processor	1
Inventory check	Worker, Processor	4
Credit check	Accountant	2
Evaluation	Processor	4
Approval	Processor, Manager	2
Billing	Accountant	3
Shipping	Worker, Processor	4
Archiving	Processor, Secretary	2
Rejection Letter	Secretary	2

Table 1: Showing the tasks, resources capable of doing those tasks and the approximate time taken by each the resources to complete the task.

1) Case arrival rate: Arrival rate of the cases for a given process is defined as the number of cases per unit time. The unit time can be set by the user of our workflow system to draw the graph shown in figure 4. Simulation agent is used to set the unit time. The arrival rate bears the direct proportion to the amount of time waiting in the queue as well as the completion time. Figure 4 shows the arrival of cases at lower and higher rates

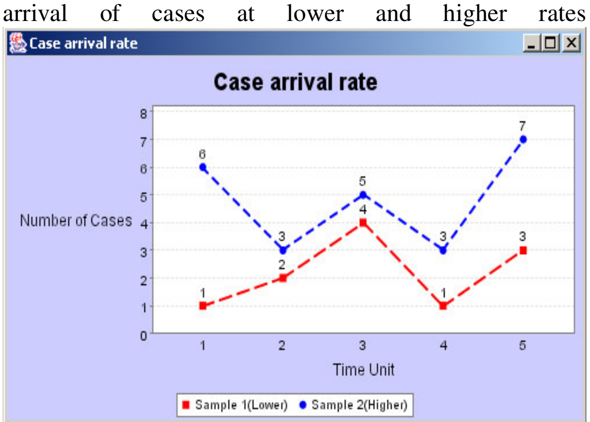


Figure 4: Graph showing the arrival rate of the cases

2) Throughput/Completion time: The figures 5 and 6 show the total completion time or the throughput of various cases during the simulation using the lower and higher rates as shown in figure 4. In each of the diagrams shown in figures 5 and 6,there are two lines indicating the system throughput corresponding to a lower number of resources available (solid line) and a higher number of resources available (dotted line).

It could be observed that the cases took more time when there were more number of tokens in the process model as the resources would be busy in their tasks while these tokens have to wait for their turn to be taken up by the resource.

It can also be observed from figures 5 and 6 that the completion times for the cases were higher when less number of resources was allocated than when more resources were allocated for doing the jobs. Our system supports the generation of the graphs during or after the simulation of the processes.

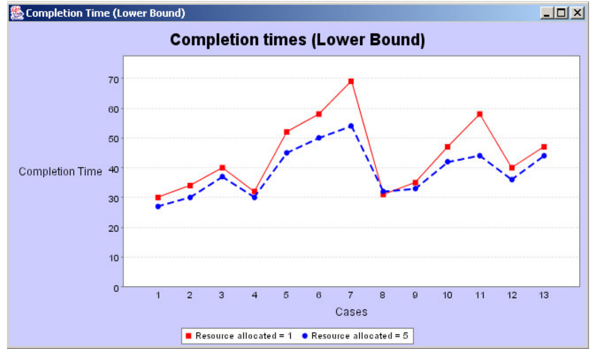


Figure 5: Graph showing the throughput time of the cases with a lower arrival rate

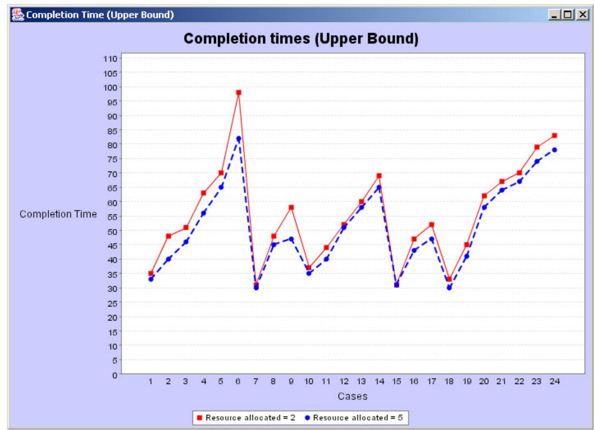


Figure 6: Graph showing the throughput time of the cases with a higher arrival rate

3) Resource utilization: Resources are assigned certain tasks that are to be completed. These resources are to be monitored for the amount of time they are busy. Figure 7 gives an overview of the percentage utilization of the all resources that were assigned with the tasks. This graph is of significance as this bears the direct relationship with the idle time of resources and also how effectively tasks are allocated to the resources.

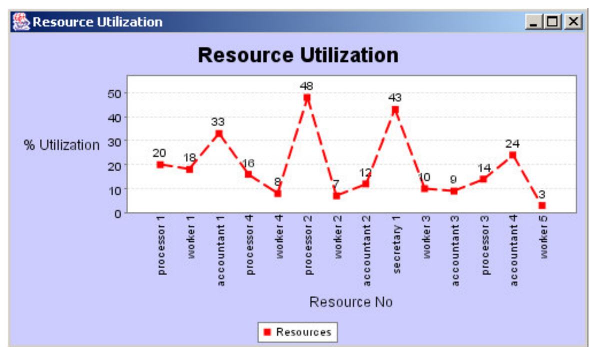


Figure 7: Graph showing the resource utilization

4) Overall waiting time: Most of the time the tasks get delayed, as the resource is busy. Figure 8 signifies the total amount of time spent waiting in the queue for getting the resources to finish the assigned task. It can be observed from that the number of jobs waiting in the queue is more when the numbers of cases added in the system were the most.

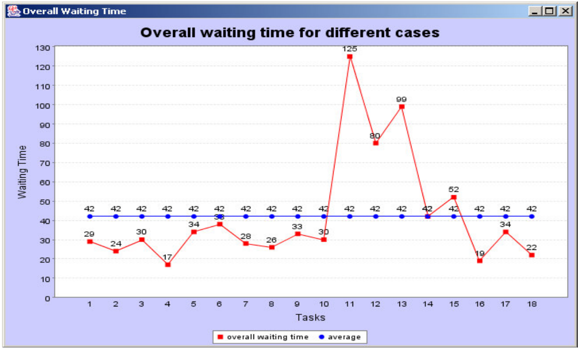


Figure 8: Graph showing the overall waiting time for different cases

5) Waiting time comparison for similar cases: Figure 9 shows the comparison of the waiting time of cases that underwent the similar path of the process model. This graph helps in the identification of those cases that have waited in the queue for a long time for most of its tasks. This graph can be taken as the basis for the inspection of those cases that have waited in the queue for most of its tasks and the reason for such behaviour can be scrutinized. One possible reason could be the status of backlog in the system at the particular time of the case submission.

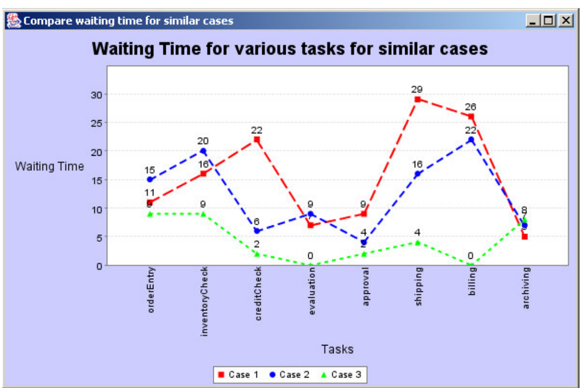


Figure 9: Graph showing waiting time for tasks for similar cases

6) Comparison of completion times varying the job tokens and number of resources: Table 2 shows the total time taken during simulation, for varying number of resources and job tokens. It can be observed from Table 2 that the total simulation time taken to complete the jobs are higher when the number of resources available were low and as the number of resources increased the completion times of simulation were less. This is attributed to easy availability of the resources to do the tasks when there were more of resources.

The total simulation times in the table are indicative of the probable number of resources that are needed to do the job in less amount of time.

Job Token / Number of each kind of resource **(R)**	**R =1**	**R = 2**	**R = 5**	**R = 10**
Job Token = 1	9	9	9	9
Job Token = 2	12	10	10	10
Job Token = 3	17	12	11	11
Job Token = 4	23	13	12	12
Job Token = 5	27	15	13	13
Job Token = 8	45	22	16	16
Job Token = 10	53	28	18	18

Table 2: Completion time of the simulation for a given number of job token and the given number of each kind of resource **(R)**

7) The efficiency of the adaptation: The ability to change the process model during run time, one of the key features of our existing framework can be verified using the monitoring system. If a process is changed (addition, deletion or modification of tasks or for example changing from parallel processing to the sequential processing of tasks), the situation can be simulated and the results can be analysed. Figure 10 shows the throughput of the system when billing and shipping are done in parallel (before change – solid line) and when shipping is done after billing (after change – dotted line). It can be observed from the graph that the parallel execution of the tasks resulted in better performance. This kind of analysis is essential for determining whether a process should under go the proposed change or not.

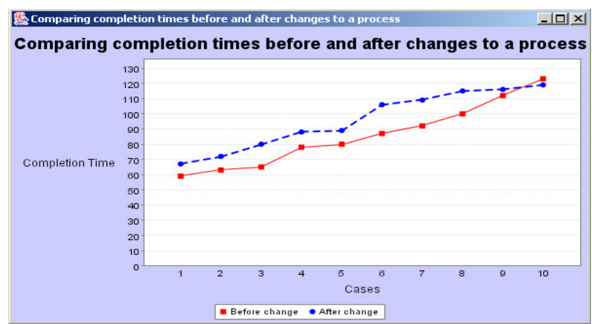


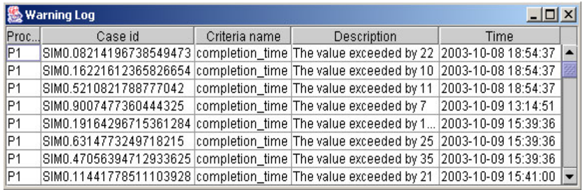
Figure 10: Graph comparing resources across tasks

3.3 Controlling the processes

When simulating various scenarios, the user/manager of the workflow system specifies certain criteria that have to be looked into continuously and should be constantly compared with the data obtained from the simulation. The user could specify if the criteria holds good for specific processes or for all processes.

When the controlling agent senses anomalies or violations to these criteria it sends the warning message to the management agent. These criteria could be the overall completion time for any given case not exceeding more than a given value or the resource utilization decreasing below a certain limit.

The controlling agent also logs the messages to the warning log so that the human manager can handle that particular problem if she is not available to look at it. The aim of this design is to capture all possible critical conditions before the enactment of the process can begin and for continuous optimization of the processes. Figure 11 is the snapshot of the warning log that is generated to indicate which criteria has been violated along with the description.



Warning Log

Proc...	Case id	Criteria name	Description	Time
P1	SIM0.08214196738549473	completion_time	The value exceeded by 22	2003-10-08 18:54:37
P1	SIM0.16221612365826654	completion_time	The value exceeded by 10	2003-10-08 18:54:37
P1	SIM0.5210821788777042	completion_time	The value exceeded by 11	2003-10-08 18:54:37
P1	SIM0.9007477360444325	completion_time	The value exceeded by 7	2003-10-09 13:14:51
P1	SIM0.19164296715361284	completion_time	The value exceeded by 1...	2003-10-09 15:39:36
P1	SIM0.6314773249718215	completion_time	The value exceeded by 25	2003-10-09 15:39:36
P1	SIM0.47056394712933625	completion_time	The value exceeded by 35	2003-10-09 15:39:36
P1	SIM0.11441778511103928	completion_time	The value exceeded by 21	2003-10-09 15:41:00

Figure 11: Warning log for the violated criteria

3.4 Advantages of our monitoring and controlling system

1. Quantitative results of the waiting times, completion times would serve in examination and analysis of any process and why such a result has been obtained.
2. Gives a clear picture of what resources are under utilized.
3. Gives the impact of possible process modifications.
4. Gives an overview of how much time the tasks have to wait in the queue, as the resources are busy. The inference could be used in altering the resource that could be used.
5. The data that is logged by the monitoring system is used by the controlling agent (described in section 3.3).

4 Conclusions

Our paper gives an overall picture about how the data that is obtained through the simulation of processes (various cases) can be used to monitor and control the overall functioning of the workflow processes in an agent based workflow system. The uniqueness of our work is in constructing an agent-enhanced workflow system that can be made to adapt an improved process model or incorporate new strategies with regard to resource usage in order to achieve better system performance through analysis of the simulation data.

The data that is obtained and logged by the simulation agent is used to draw the graphs that are used for the analysis of the performance of the various cases. The controlling agent continuously monitors the data and logs appropriate warning messages as well as displays warning message to the user of the simulation in case of any anomalies with reference to the criteria specified by the user of the system. The monitoring and controlling agents help in optimizing the workflow as well as improve the effectiveness of the system.

As a part of our future work, we would like to use the same monitoring and controlling mechanisms and modules to the enactment of the processes. The main emphasis during enactment would be on distributed monitoring and controlling using agents. We have planned to incorporate the intelligent controlling of the processes using an enhanced controlling agent. We also intend to integrate into the framework one of the Petri net analysis tools in order to examine the model for potential bottlenecks and whether certain (desirable/undesirable) states can be reached.

5 Acknowledgements

The authors wish to thank Lars Ehrler for his help in the design and implementation of the enhanced system and Prof. Martin Purvis for supporting this work.

Otago Research Grant has supported this work.

6 References

Fleurke, M., Ehrler, L. and Purvis, M. (2003): Jbees - an adaptive and distributed framework for workflow systems. *Proc. IEEE/WIC International Conference on Intelligent Agent Technology*, Halifax, Canada.

Muehlen, M.Z. and Rosemann, M. (2000): Workflow-based Process Monitoring and Controlling – Technical and Organizational Issues. *Proc. 33rd Hawaii International Conference on System Sciences.* Wailea, IEEE Press.

Wang, M. and Wang, H. (2002): Intelligent Agent Supported Workflow Monitoring System. *CAISE 2002, LNCS 2348*, 787-791.

Cui, Z., Odgers, B. and Schroeder, M. (1998): An In-Service agent monitoring and analysis system. *In Proceedings of the 11th IEEE International Conference on Tools with Artificial Intelligence*, Chicago, USA, 237-244.

Meilin, S., Guangxin, Y., Yong, X. and Shangguang, W. (1998): Workflow Management Systems: A survery. *In Proceedings of IEEE International Conferece on Communication Technology*. Beijing, China.

JFreeChart: Java Free Chart, Object Refinery Limited, http://www.jfree.org/jfreechart/index.html Accessed 01 Sep 2003.

Purvis, M.K., Cranefield, S.J.S., Nowostawski, M. and Carter, D. (2002): *Opal: A multi-Level Infrastructure for Agent-Oriented Software Development.* Department of Information Science, University of Otago: Dunedin, New Zealand.

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

Aalst, W.M.P.v.d., Hofstede, A.H.M.t., Kiepuszewski, B. and Barros, A.P. (2002) *Workflow Patterns*, Queensland University of Technology, Brisbane, Australia.

Jensen , K. (1992): Coloured Petri Nets – Basic Concepts, Analysis Methods and Practical Use, Vol. I: Basic Concepts. EATCS Monographs on Theoretical Computer Science, Heidelberg, Berlin. 1-234.