

# AN EVALUATION OF A COOPERATIVE PROCESS SUPPORT ENVIRONMENT

Alf Inge Wang

Dept. of Computer and Information Science,  
Norwegian University of Science and Technology, N-7491 Trondheim, Norway. Email: alfw@idi.ntnu.no

## ABSTRACT

This paper describes an evaluation of process modelling where the same distributed conference organizing process was modelled in three different process centred environments (PCEs) - Endeavors, ProcessWeb, and our own CAGIS PCE. Endeavors is an activity based, flexible workflow system, ProcessWeb is a role-based workflow system with a web-interface, whereas the CAGIS PCE combines an activity based workflow system with a software agent system. The goal of the evaluation was to investigate if a combination of a traditional workflow system and a software agent system is more suitable for modelling, changing and configuring distributed cooperative processes than stand-alone workflow systems. As far as we know, there exist no similar studies for process centred environments.

**Keywords:** Workflow, Computer Supported Cooperative Work (CSCW), Software Agents, Comparison, Evaluation, and Software Engineering.

## 1 INTRODUCTION

Over the last 15 years, there has been a strong research interest in improving software process support resulting in various process centred environments (PCEs). However, there are very few studies of actual usage of these environments. The lack of studies on the practical benefits of such environments has driven many companies to use informal in-house process modelling languages (PMLs) to describe their processes in text or by using general-purpose drawing tools. There is a substantial need for validating process centred environments, to see if they can improve software engineering processes in projects.

Real validation of PCEs is hard to conduct. This paper describes a small study of how suited three different PCEs are in modelling and configuring and managing changes of a process. In this paper, *process modelling* means the activity of describing a process as activities, roles, artefacts, and relationships using a graphical or textual process modelling language. *Configuring* a process model is the activity of making a process model executable (instantiating the process model). The suitability of the PCEs is measured by measuring process model coverage and how much effort spent on modelling and changing the process. Our work does not include analysis or measurement of enactment of the process, but the process models have been run in all three environments to ensure that they are executable.

Originally we wanted a realistic cooperative software engineering example from the software industry to evaluate

our CAGIS<sup>1</sup> PCE. We first tried to search in the research literature and among international colleagues for such reference example, but we could not find any. Our second attempt was to contact companies to try to find a cooperative software engineering example we could use. In the end, we had to admit that it was impossible to find a real example that was described in sufficient detail. The chosen example is a well-known *conference organizing process*. This scenario has previously been described and analysed ([5, 1]), and contain both individual activities and more dynamic, cooperative activities.

## 2 EXPERIMENT DEFINITION

The evaluation described in this paper uses the *assertion* validation methods according to Zelkowitz and Wallace classification of validation methods [10]. Assertion is an ad-hoc, observational validation method where developers execute their own experiment to see if one proposed technology is better than other alternatives. A problem by using this validation method is that to that the result can have limited external validity. This problem is further discussed in section 7.

### 2.1 EXPERIMENTAL PROCESS

The process of the experiment started by selecting the goals to achieve and the process example to be used. Then the process example was analysed and modelled and configured (also includes modelling of process changes) in the three PCEs. During the modelling and configuration, the effort and completeness of models were measured. The empirical data (quantitative research) was then used as a starting point for qualitative discussions.

### 2.2 GOAL AND RESEARCH QUESTIONS

The goal of our experiment was to compare three different PCEs to evaluate their suitability to model, change and configure the same conference organizing process. More precisely, we wanted to address the following research questions:

*Is a combination of a traditional workflow system and a software agent system better than a stand-alone workflow system to:*

- Q1** *Model and configure processes, containing both dynamic cooperative activities, as well as structured, individual activities?*
- Q2** *Adapt to process changes in both these activities?*

<sup>1</sup>CAGIS is a Norwegian research project and is an abbreviation for Cooperative Agents in Global Information Space

## 2.3 CONTEXT OF THE EXPERIMENT

The evaluation was run by Alf Inge Wang at the Dept. of Computer and Information Science, at the Norwegian University of Science and Technology (NTNU) in Trondheim, Norway. Mark Greenwood, in Informatics Process Group at the University of Manchester modelled the cooperative activities in ProcessWeb, and Professor Richard N. Taylor, Dept. of Information and Computer Science at the University of California Irvine, suggested a solution to model cooperative activities in Endeavors.

The experiment was run at NTNU. This means that the result of the experiment is not automatically valid in a real setting, but the result should give an indication of what to expect. The experiment addresses a real problem, i.e., how to give sufficient process support to both structured, individual activities as well as cooperative dynamic activities. The full description of the experiment can be found in [7].

## 2.4 RESEARCH QUESTION EVALUATION

The evaluation of the two research questions listed above will be answered in a combination of quantitative and qualitative methods. By quantitative means that, **Q1** will be measured by looking at how much of (coverage) the conference organizing process is modelled and configured by the process model in each PCE. The *coverage* will be measured in number of activities that can be modelled and enacted. The scale 0-5 will be used to express how much of an activity is modelled. If every aspect of an activity can be modelled it will be weighted by 5 and 0 if none. The conference process consists of 25 activities, and the percentage coverage of the total model will be measured by:  $(\text{score} / 25 \times 5) * 100\%$ . In addition, the effort spent on creating process models in the three process environments will be measured (in person hour or days).

Further **Q2** will be measured by measuring *adaptability*. Adaptability is measured, by assessing the effort spent on implementing a specific process change. A scale 1-5 will be used to indicate how much effort has been used. The value 5 indicates *Very little effort* (5 minutes or less), while 1 indicates *Very strong effort* (5 hours or more).

The empirical data collected in this experiment is a bit limited since only one scenario is modelled. Qualitative discussions will therefore be used to supplement the discussions.

## 3 THREE PROCESS CENTRED ENVIRONMENTS

This section describes the three process centred environments Endeavors, ProcessWeb and CAGIS PCE, we used in the evaluation. Endeavors and ProcessWeb were selected because they are very flexible activity-based and role-based workflow tools respectively.

### 3.1 ENDEAVORS

Endeavors is an open, distributed, extensible process execution environment developed at University of California Irvine, and has been licensed by Endeavors Technology

Incorporated. It is designed to improve coordination and managerial control of development teams by allowing flexible definition, modelling, and execution of typical workflow applications. There are five main characteristics for Endeavors: Distribution, integration, incremental adoption, customisation and reuse, and dynamic change.

The Process Modelling Language (PML) used in Endeavors is a graphical object-oriented language, based on the Teamware process modelling language [9], and consists of five major modelling *categories* [3]: Activity, artefact, resource, network and arc. Each of these categories has their own set of messages, definition of fields, and default handlers, but each definition can be dynamically extended, altered, assigned, removed, or re-specified.

Processes can be modelled in Endeavors using the graphical Network Artist tool enabling the user to draw processes using a palette of activities and control-flow mechanisms. If the standard available activities are not sufficient, user-defined activity types can be created. When an activity is activated, a handler is executed. A handler is a program using Endeavors API to communicate with the system. Handlers can be used to wrap commercial tools, or the user can create his/her own tools (questionnaire, simple user-interfaces etc.). Handlers can be implemented using one of the supported programming languages Java, Python, Ada95, or Tcl.

### 3.2 PROCESSWEB

ProcessWeb [8] is a web-interface based workflow system based on the ProcessWise[6] Integrator (produced by ICL) implemented by Information Process Group, University of Manchester. The web-interface is provided through the ProcessWise Integrator application interface. ProcessWise Integrator creates an environment enabling the activities of people in an enterprise to be coordinated and integrated with the organization's computing facilities. A process management system built using the ProcessWise Integrator has a client/server structure and consist of four main components: User Interface, Process Control Manager, Process description in PML, and an Application Interface.

ProcessWise Integrator's PML is also object-oriented, and uses roles as the main concept for modelling processes. Objects are called *roles*, and *interactions* are used to provide communication channels between roles. The role concept helps address the complexity of models by providing a quasi-intuitive structuring of activities (human processes are messy and might not be amenable to hierarchical approach). A role is defined by its *actions* (methods) and its *resources* (attributes). Preconditions called '*when guards*', are used to select action for enaction, and they are expressed as if statements. Interactions are unidirectional, asynchronous typed communication channels provided through a takeport and a giveport. A *takeport* received data or control flow from another role, and a *giveport* sends data or control flow to another role. The giveport and takeport represents two ends of a single interaction. No

global variables can be defined in the PML.

### 3.3 CAGIS PCE

The CAGIS PCE consists of three interacting main components [7]:

- **CAGIS SimpleProcess** is a workflow tool with a web-interface, used to model and guide individual activities. An activity can typically be presented to the user as a work description, links to relevant documents or tools, HTML-forms, and/or Java applets. A process in CAGIS SimpleProcess can consist of several autonomous process fragments that can be distributed over several workspaces. To specify the process model XML or a web-based *process modeller tool* can be used. Hierarchical workspaces are used to model roles in the organization, and CAGIS SimpleProcess has support for moving process fragments between workspaces (roles).
- **CAGIS Distributed Intelligent Agent System (DIAS)** provides support for cooperative activities between roles (workspaces). The software agents in CAGIS DIAS can be used to coordinate artefacts and resources, negotiate about artefacts and resources, monitor the working environment for changes or events, provide infrastructure for brainstorming, electronic meetings, trading services etc. CAGIS DIAS provides the infrastructure for creating co-operating agents, and consists of four main components: Agents, agent meeting places (AMPs), workspaces, and repositories.
- **CAGIS GlueServer** The CAGIS GlueServer is a middleware that uses a so-called GlueModel, where relations between process fragments (CAGIS SimpleProcess) and software agents (CAGIS DIAS) are defined. The GlueModel can be used to define rules for interaction between roles, and to specify changes of the process model. By using a *separate model* to describe the interconnection between agents and process fragments, it is possible to use other workflow tools as well as other agent systems. This makes the CAGIS PCE open-ended to other systems.

The process model in the CAGIS PCE consists of three parts:

- The **CAGIS SimpleProcess PML** that describes process as a activity-network. Each activity is represented by a web page that has a state and relationships to other activities.
- The **DIAS Agent Language** is a high-level agent API written for Java for modelling cooperative activities. KQML is used to define how agents should communicate, and the content of a speech-act (e.g. tell-agent, ask-agent etc.) is specified in XML. It is possible to implement both stationary and mobile agents. Examples of agent methods available are: migrate agent, communicate to other agent, tell other agent, announce service, request service, etc.
- The **Glue Modelling Language** uses XML to define the relationship between a process fragment and a software agent, and the reactions to take depending on the results of the agent execution.

## 4 THE STUDIED EXAMPLE

In this section, we describe the conference organizing process used as a case for the evaluation. In Olle et al. [5], a short textual description of the conference organizing process is presented, and a more detailed description stands in [1]. We describe the scenario using a simple graphical notation showing activities and their sequence, and some additional textual description to clarify the scenario.

The high-level process of the conference organizing process consists of seven high-level activities A1-A7. A decomposition of the conference process, including the high-level activities, is shown in figure 1. The roles involved in this process are PC Chair, PC Member (several instances), and Session Chair (several instances). An assumption for the scenario is that the conference organizing process is executed distributed without any face-to-face interaction.

In the next subsection (4.1), the main activities of the process will be described. Section 4.2 describes the negotiation process which is a more detailed description of the activities A3.2, A.6.2.3 and A6.2.5. Finally, section 4.3 will describe changes applied to the process described in figure 1.

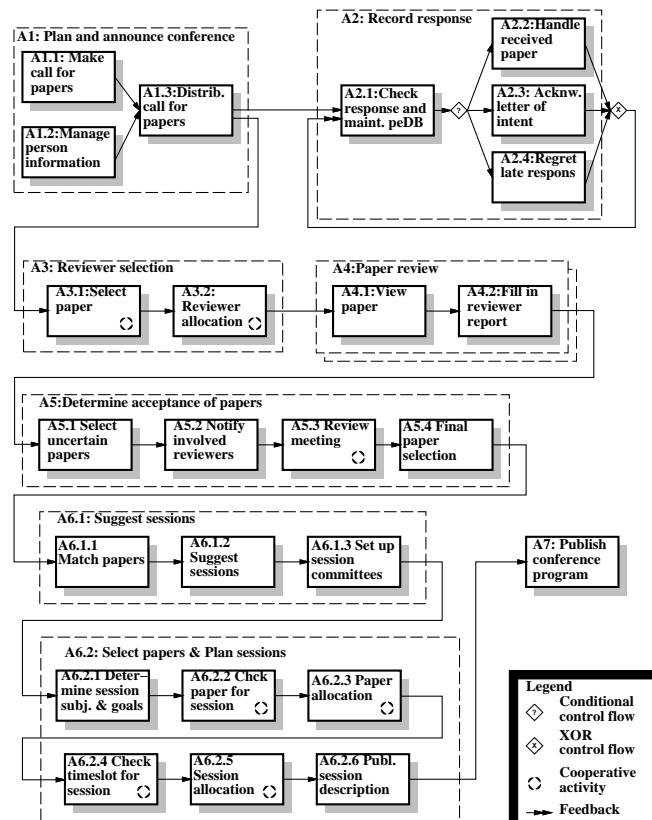


Figure 1. Decomposition of the conference process.

### 4.1 A DESCRIPTION OF THE ACTIVITIES

To get an overview of the process, here is a short review of the full description of the activities [7]:

A1 **Plan and announce conference** consists of three sub-activities that make and distribute call for papers and manage available person information (mailing-lists etc.). *PC Chair* is responsible for this activity.

A2 **Record response** consists of four sub-activities that manage responses to the conference (questions, papers, etc.). *PC Chair* is responsible for this activity, but sub-activities can be delegated to other *PC Members*.

A3 **Reviewer selection** consists of two cooperative activities where PC Members can indicate what papers they want to review by selecting papers from a list. Reviewers will be selected according to a negotiation process as described in section 4.2. All *PC Members* are responsible for this activity.

A4 **Paper review** consist of two sub-activities where allocated reviewers review papers and fill in reviewer reports. All *PC Members* are responsible for this activity.

A5 **Determine acceptance of papers** consists of four sub-activities where the papers to be accepted are determined. In cases where the reviews are pointing in different directions, the involved actors hold cooperative review meetings. *PC Chair* is responsible for this activity, but *PC Members* might be involved.

A6.1 **Suggest sessions** consists of three sub-activities where the *PC Chair* matches papers for topics and make an initial session plan.

A6.2 **Select papers and Plan sessions** consists of six sub-activities where papers are allocated for sessions and timeslots are allocated for sessions. Note that the paper allocation and timeslot allocation are cooperative activities, where *Session Chairs* select papers and timeslots from a list, and a negotiation between involved parties is initiated if there are collisions.

A7 **Publish conference program** is the last activity where the *PC Chair* distribute the final conference program.

## 4.2 N1: NEGOTIATION PROCESS

If more than three PC members are interested in reviewing the same paper (see A3.2 in figure 1), or more than one Session Chair is interested in the same object (a paper in A6.2.3 or a timeslot in A6.2.5 in figure 1), the following steps will be used to solve the conflict of selecting objects for the involved actors:

N1.1 The actor with *most objects marked* will be forced to "unmark" the conflicting object.

N1.2 All involved actors will be asked at one at a time to "unmark" the conflicting objects.

N1.3 One or more involved actors will be picked out randomly to "unmark" the conflicting object. This time the actor will be forced to "unmark" object.

The process will start with N1.1. If N1.1 does not solve the conflict, N1.2 will be initiated. If still there is a conflict, N1.3 will be iterated until the conflict is solved.

We have developed the negotiation strategy described above, and it only shows *one approach* to solve the conflict in a simple and clean way. Other negotiation strategies could be used, but are not described here.

## 4.3 PROCESS CHANGES

This section describes the process changes that were modelled and configured in the three PCEs after the initial process has been modelled and enacted. Note that these process changes occur during enactment. The following process changes should be applied to the scenario:

C1 **Change activity sequence.** The activities A6.2.4 and A6.2.5 are exchanged with A6.2.2 and A6.2.3. The new sequence for A6.2 will then be A6.2.1, A6.2.4, A6.2.5, A6.2.2, A6.2.3, and A6.2.6 (see figure 1).

C2 **Assign activity to another role when one role is not available.** The activity A2: Record response should be re-assigned to PC Member #1.

C3 **Change negotiation strategy in N1.** The new negotiation strategy *N2*, involves three steps:

N2.1 All actors that have selected conflicting objects, will be asked to "unselect" these objects at the same time. The main difference with this step and N1.2 in N1 is that this step is executed in parallel while in N1 is executed sequentially.

N2.2 For the remaining conflicting objects, a communication channel is opened between actors that have selected the same conflicting object. By using this communication channel (e.g. talk, IRC or similar), the actors can give arguments for who should "unselect" this object. For paper reviewing, expertise within a certain research field can be used as an argument.

N2.3 For the remaining conflicting objects, actors will pick out at random to "unselect" these object until the conflict is solved.

C4 **Change reviewer selection.** In a new version of the activity A3, this activity is monitored. If all reviewers are not allocated within a certain time, PC Chair will make a list of reviewers without any intervention from the PC Members.

## 5 EXPERIMENTAL RESULTS

To make it easier to compare the different PCEs and to reduce the modelling time, we chose to represent all individual activities (opposed to cooperative activities) as web pages describing what to do including HTML-forms, and links to relevant tools and documents. All three PCEs accessed the same HTML-files for presenting the activities to the users.

### 5.1 MODELLING IN ENDEAVORS

The modelling in Endeavors consisted of two main parts: The implementation of the handlers, and the graphical modelling of the process. A handler is executed when an activity in Endeavors is executed. We chose to implement

the handlers in Java, since also the Endeavors framework is implemented in Java. The following activity handlers were implemented:

- H1 **Show HTML handler** was used to show HTML for individual web pages. In addition, we used this handler to provide the cooperative activities Select paper (A3.1), Check paper for session (A6.2.2), and Check timeslot for session (A6.2.4). A3.1, A6.2.2 and A6.2.4 accessed a database through HTML-forms using CGI.
- H2 **Allocation handler** is responsible for detecting conflicting objects for current actor by accessing a database (Paper-list or Session-schedule) through a CGI. In addition, this handler will initiate the negotiation process (N1) as described in section 4.2.

## 5.2 MODELLING IN PROCESSWEB

In ProcessWeb, the conference organizing process was modelled in two parts.

The *individual activities* was modelled as three roles: PC Chair, PC Member, and Session Chair. Each role has several states representing the activities that this particular role is responsible for. For each state, HTML was used to provide a user interface representing each activity. The role changes state when the user clicks on a button in an HTML-form, notifying ProcessWeb to change state. Interaction channels were implemented between roles, making it possible for e.g., notify the PC Member when PC Chair had finished an activity.

Only the *cooperative activities* A6.2.4 and A6.2.5 were implemented in ProcessWeb. The reason for this was, that the other cooperative activities were similar and did not give any new challenges. Mark Greenwood carried out the implementation of the cooperative activities. This was done to get an expert to implement the more complicated aspects of the scenario. These activities have been modelled through two roles: SessionChair and TimeTable. In the SessionChair role, the user can select what timeslots (s)he is interested in. The TimeTable role is the role responsible for the allocation of timeslots for different sessions. From the TimeTable role, the allocation process is initiated, including the negotiation process N1 described in 4.2.

## 5.3 MODELLING IN CAGIS PCE

In CAGIS PCE, the scenario was modelled using CAGIS SimpleProcess PML, cooperative agents and the GlueModel.

The CAGIS SimpleProcess PML was used to model the individual activities in the conference organizing scenario. All activities were described as *< activity >* elements in an XML-document.

The GlueModel was used to model the relationships between the individual activities (previous paragraph) and the cooperative agents (next paragraph). The GlueModel contains six process fragment - agent pairs for the activities A3.1, A3.2, A6.2.2, A6.2.3, A6.2.4, and A6.2.5. For the

selection activities (A3.1, A6.2.2, and A6.2.4), the GlueModel specifies that *selection agents* are used for selecting papers or timeslots. If the selection agent fails, current activity will be re-executed. The GlueModel further specifies that for the allocation activities (A3.2, A6.2.3, and A6.2.5), *negotiation agents* are initiated. If the allocation (of papers or timeslots) performed by the negotiation agents goes well, the next activity in SimpleProcess will be executed. If the negotiation process discovers that the user has to select more papers or timeslots, an activity for selecting papers or timeslots will be executed.

Cooperating agents were used in this scenario for selecting papers and timeslots, and for allocating the same resources. A selection agent provides the user-interface to select a paper to review. Depending on the steps N1.1-N1.3 in the negotiation process N1 (see section 4.2), the negotiation agent initiate actions according to this description. Cooperative agents are a separate research field, and we have done little on detailing cooperative/negotiation agents, and providing a rich infrastructure for such agents.

## 6 EVALUATION OF THE THREE PCEs TO MODEL THE EXAMPLE

In this section we will compare Endeavors, ProcessWeb and CAGIS PCE.

### 6.1 COVERAGE OF THE SCENARIO (Q1)

Table 1 shows the measures of model coverage and modelling time for the three PCEs. Note that additional time was used to create HTML for the individual activities used in all PCEs, and is not shown in the table.

Modelling	Enda.	ProcWeb	CAGIS
Indv. activities	100 %	100 %	100 %
Coop. activities	80 %	100 %	100 %
Modl. time indiv. acti.	6 hours	10 hours	3 hours
Modl. time coop. acti.	10 days	2 days	5 days

Table 1. Coverage and effort of scenario modelled

Our first research question (Q1), was how complete the three different PCEs could model the conference organizing scenario. For individual activities, all PCEs could model all activities completely. For cooperative activities (A3.1, A3.2, A6.2.2, A6.2.3, A6.2.4, and A6.2.5 <sup>2</sup>), we had some problems in modelling the allocation of papers and timeslots in Endeavors. The problem was how to represent negotiation process between different roles in an activity-network. We chose to solve this problem by invoking an allocation handler for each role at a time. After all roles had activated their allocation handler, the databases were checked to see if there were remaining paper or timeslots conflicts. If there were more conflicts, the same allocation handlers for all roles involved were executed again. This loop continued until all conflicts were solved. Figure

<sup>2</sup>IRC was used to implement the cooperative activity A5.3

2 shows graphical process model of A3: *Reviewer selection*, where a loop with A3.2 for all roles involved (here five) models the allocation process.

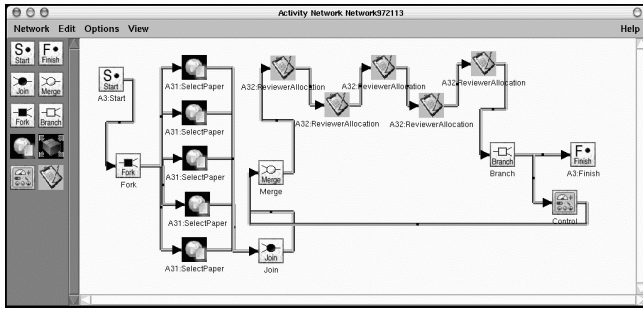


Figure 2. Activity A3: Reviewer selection modelled in Endeavors

The modelling time in table 1 shows that for individual activities most time was used in ProcessWeb and least in CAGIS PCE, where Endeavors comes in between. One explanation could be that the modeller was most familiar with CAGIS PCE. However, modelling the individual activities in Endeavors and ProcessWeb was very straightforward, and we were not delayed by any time-consuming problems. The difference in modelling time can be explained as follows:

- **ProcessWeb:** The PML in ProcessWeb is close to a textual object-oriented programming language. This means that the modeller needs to “program” the process as roles with different states representing the activities the role is responsible for. In addition, the modeller has implement the infrastructure for interaction between roles, assignment of roles, and role configuration.
- **Endeavors:** More time was spent on modelling in Endeavors compared to CAGIS PCE, because in Endeavors we had to implement an activity handler that initiated an HTML-document in a Web-browser. If we don’t consider the time spent on implementing the activity handler, Endeavors was more efficient for modelling the individual activities than CAGIS PCE.

According to table 1, ProcessWeb is the most efficient PCE for modelling cooperative activities. The reason is that the PML used in ProcessWeb is designed to efficiently implement role interaction. CAGIS PCE used an agent-API to implement the cooperative activities. The current version of our CAGIS agent-API is too low-level for efficient cooperative activity modelling. It should also be noted, that the user interface used in the CAGIS PCE was more advanced compared to simple web-interface in ProcessWeb. In Endeavors, the cooperative activities were implemented in Java from scratch, making this the least efficient approach. Richard N. Taylor at UCI<sup>3</sup> has suggested (through email correspondence) to use globally visible blackboard

<sup>3</sup>University of California, Irvine

for papers and timeslots in Endeavors, accessed by various roles. This was not implemented because the documentation for implementing such blackboards was not available.

## 6.2 PROCESS CHANGE ADAPTABILITY (Q2)

This subsection describes how easily the three PCE can adapt to the process changes described in section 4.3. In all three PCEs, the process model can be changed during enactment. In Endeavors, the process model can graphically be manipulated during enactment, and activity handlers can dynamically be changed as well. In ProcessWeb, a process change is provided by extracting data and state from a current role, a new role definition replaces the current role, and the extracted data and state is restored in the new role definition. The CAGIS PCE, the activity network can be changed by moving process fragments to another workspace or another place in the workspace.

Table 2 shows the effort to implement the process changes C1-C4 in the three PCEs scaled 1-5:

Process change	Endea.	ProcWeb	CAGIS
C1. Change act. seq.	5 (2min)	3 (30min)	5 (2min)
C2. Assign act. to role	5 (2min)	2 (2h)	5 (2min)
C3. Change neg. strategy	1 (5h)	3 (30min)	5 (2min)
C4. Change reviewer sel.	1 (5h)	3 (30min)	5 (2min)

Table 2. PCE adaptability to process model changes

Here are comments on results for each process change (C1-C4) shown in table 2:

### C1 Change the activity sequence:

In Endeavors, to do this process change, we simply had to remove two control flow arcs and add two new ones. In ProcessWeb this particular process change depends on how the activities are represented in the roles. If the activities are represented as states in the role, some if-sentences must be changes to change the activity sequence. The changed role definition must in addition be compiled into the system, and the role must be modified to get the new behaviour. In CAGIS PCE, to change an activity sequence, we simply moved the activities (the process fragment) to the wanted place in the activity network.

### C2 Assign an activity to another role:

In Endeavors, editing the *AssignedTo* attribute for an activity through the Endeavors graphical user interface can change role assignment. In ProcessWeb, assigning an activity to another role is more complicated, because the role is the unit of change. First, the PML code describing the activity must be transferred from one role to another. Second, the PML code for both roles (source and target roles) must be changed to cope with the removal of activity in the source role and adding the activity in the target role. Third, PML code for both roles must be compiled, and both roles must be modified to their new

behaviour. This problem can be avoided by modelling all activities as separate roles. In CAGIS PCE, hierarchical workspaces are used to represent roles. To re-assign an activity to another role, this activity is simply moved to another workspace.

### C3 Change negotiation strategy:

We have not in detail considered the effort of implementing the new negotiation strategy, but rather how to integrate an already implemented negotiation strategy into the process. In Endeavors, it is rather hard to implement the N2 negotiation strategy (described in section 4.3) because it requires communication across different roles, which is hard to model in an activity network. ProcessWeb is probably the best environment for implementing the N2 negotiation strategy. Yet, to incorporate this strategy with existing roles can be hard. N2 demands extensive changes of role definitions and interaction channels between roles. If this process change had been known in advance, the process model could incorporate a meta-process to handle such changes. Here, a process architecture is used to provide support for a generic change making migration easier. Essentially making one change is hard, but with an explicit meta-process the cost could be spread over many changes over the lifetime of a system. In CAGIS PCE, it is very easy to change negotiation strategy by editing the GlueModel for the process fragments involving negotiation agents. Implementing N2 in the agent system is probably more time consuming than using ProcessWeb, but it is really easy to change negotiation strategy if a matching implementation (cooperative pattern) is already available.

### C4 Change reviewer selection:

This process change demands that the activity A3: *Reviewer selection* is monitored. If three reviewers are not allocated to all papers within a certain time, the PC Chair will do the selection of reviewers himself. In Endeavors this process change can be implemented by adding timeout functionality for the activity handlers in A3. When the timeout event occur, a control activity will route the workflow to PC Chair to do the reviewer selection her/himself. This process change demands a lot of changes in both the activity handlers and the activity network. In ProcessWeb this process change is easier to implement, but will anyhow demand some effort. The role assigned originally for A3 (PC Member) need to have a timeout function for selecting and allocating reviewer to papers. When this timeout function is triggered, a message is sent to the PC Chair role that (s)he must do the reviewer to paper allocation. This means that the PC Chair and the PC Member role definitions must be changed, and the communication channel between them must be (re-)configured. In CAGIS PCE, the GlueModel can be used to implement this process change. A monitor agent will monitor the reviewer allocation agent. If they are not finished within a certain time, the monitor agent will notify the GlueServer. The GlueServer will then initiate an execution of a Reviewer selection process fragment for PC Chair. The definition of the Reviewer selection process fragment for

PC Chair must be defined. Since monitor agents are a part of the CAGIS DIAS, changes must be made in the GlueModel, and an additional process fragment for reviewer selection for PC Chair must be defined.

## 7 DISCUSSION

Looking back on the research questions in section 2.2, we can from the results in the two above sections say that a combination of a traditional workflow system and a software agent system is better compared to a stand-alone workflow system to model and enact cooperative processes and adapt to process changes. Some objections can however be raised to this statement:

- O1 **The chosen conference scenario is non-representative.** The reason for selecting the conference organizing scenario was in order to use a scenario that was already described in the literature (external validity). One important question is whether the scenario is representative for cooperative software engineering processes. To answer this question, we have to look at how the conference scenario has been modelled. We have distinguished between individual and cooperative activities. Individual activities are activities where one role is assigned for performing this activity. In cooperative activities, several roles are involved in performing the activity. We believe that as long as other scenarios are modelled by distinguishing between individual and cooperate activities, the evaluation result will be valid. If another approach is used to model the process, the evaluation results are not necessary valid.
- O2 **The selection of process changes is biased.** We have tried to pick out process changes that are likely to occur in the conference scenario. It is possible that our selection is too limited, and more changes should be considered. Other process changes could have indicated a more nuanced score for the three PCEs. However, we believe that the selected process changes represent a wide spectrum of possible changes making the result believable.
- O3 **More experience with your own environment.** A problem with our experiment is that the modeller is more experienced with our own environment than others. This means that the modelling time can be unreliable. To address this problem, we spent about one week to exercise process modelling in Endeavors and about one month with modelling processes in ProcessWeb. The reason more time was spent on ProcessWeb, was that the author had used ProcessWeb for modelling before the experiment was planned. During the experiment, we had no time-consuming problems when modelling the individual activities in Endeavors and ProcessWeb. For cooperative activities a modelling expert in ProcessWeb was used, but not for Endeavors. This means that the modelling time for cooperative activities in Endeavors is likely to be less if an expert had been used. However, since modelling interaction between roles in Endeavors is not a part of the

PML, it is likely that the evaluation result will point in the same direction.

**O4 Statistically invalid data.** Using only one target process (conference scenario) we cannot statistically validate our two research questions. For a statistically valid experiment, data should be collected from several scenarios modelled in the three PCEs. The collected data has only been used as an indication of the evaluation result, and qualitative discussions have enlighten the results from data collection.

It should be noted that Endeavors and ProcessWeb are much more mature environments compared to CAGIS PCE, and they offer richer semantics for expressing the process. In CAGIS PCE, we have used agents to provide the functionality that we lose through having only a simple workflow model. Since the agent-API is still to low-level, more advanced workflow can be time consuming to implement in CAGIS PCE.

## 8 RELATED WORK

As far as we know, there exist no similar evaluations of PCEs as the one described in this paper. In 1990, the “*ISPW-6 software process example*” [4] was developed as a standard software engineering scenario for modelling software processes. This process example was used as a benchmark to show that PCEs e.g., Merlin, EPOS, ICL Process-wise Integrator, and Valmont, could model and execute the given process. The ISPW-6 example was later extended to incorporate team-work and process change (ISPW-7). The main problem with the ISPW-6 and 7 process examples are that they reflect very general software processes. Even if ISPW-6 and 7 have been modelled in many PCEs, there are no papers describe an evaluation of different PCEs modelling the same process example in regards to modelling completeness and modelling efficiency. In [2], a survey is presented where thirteen PMLs are classified according to characteristics like process elements and support for different meta-process phases. This survey does not include any actual modelling using the different PMLs.

## 9 CONCLUSION

This evaluation was mainly performed by the author. For similar future evaluations, we would like to avoid this approach to ensure the validity of the results. One approach could be to use students that are not familiar with any of the PCEs, give them a proper introduction to the PCEs (the same amount), and let them model some processes. By using this approach, we would be able to collect more data, and could make some statistical analysis. A problem would be to find and pick descriptions of neutral scenarios that not favour one of the PCEs. Another approach would be to use experts on each PCE to model processes. A problem with this approach is to deal with interpretations of the scenarios. Also if the experiment was not controlled, it could be possible to fake the measurements. We can anyhow conclude that carrying out such experiments are very hard and time-consuming.

The main contribution of our work is the CAGIS GlueServer, facilitating couplings between activity-based workflow and software agents. With regards to modelling efficiently, ProcessWeb was the best environment for cooperative activities. When the activity handlers had been implemented in Endeavors, Endeavors was most efficient for modelling individual activities. Future work should therefore investigate how a combination of Endeavors, ProcessWeb and CAGIS GlueServer would work as a federation of workflow systems. This combination features solid and rich modelling support for both individual and cooperative activities, where the CAGIS GlueServer serves as a middleware. Further, the GlueServer can be used to combine more than two workflow systems, allowing loosely coupled, autonomous organizations to choose their own workflow tool. In this way, the CAGIS PCE provides cooperative support for a heterogeneous environment.

## Acknowledgements

We would like to thank Mark Greenwood for modelling parts of the conference management process in ProcessWeb, and for giving valuable feedback during the experiment. We would also like to thank Ian Robertson, Richard N. Taylor, and Letizia Jaccheri for useful comments.

## REFERENCES

- [1] S. Carlsen. *Conceptual Modeling and Composition of Flexible Workflow Models*. PhD thesis, Dept. of Computer and Information Science, Norwegian University of Science and Technology, Trondheim, Norway, December 15 1997.
- [2] J.-C. Derniame, B. A. Baba, and D. Wastell. *Software Process: Principles, Methodology, and Technology*. Springer Verlag LNCS 1500, Berlin, Germany, 1998.
- [3] Gregory Alan Bolcer. *Flexible and Customizable Workflow Execution on the WWW*. PhD thesis, University of California, Irvine, 1998.
- [4] M. I. Kellner, P. H. Feiler, A. Finkelstein, T. Katayama, L. Osterweil, M. Penedo, and H. D. Rombach. Software Process Modeling Problem (for ISPW6), Aug. 1990.
- [5] T. W. Olle, H. Sol, and A. A. Verrijn-Stuart. *Information Systems Design Methodologies: A Comparative Review*. North-Holland, 1982.
- [6] I. E. Process Management Centre, Enterprise Technology. *ProcessWise Integrator, PML Reference*. Staffordshire, UK, first edition, April 1996.
- [7] A. I. Wang. *Using a Mobile, Agent-based Environment to support Cooperative Software Processes*. PhD thesis, Norwegian University of Science and Technology, Dept. of Computer and Information Science, NTNU, Trondheim, Norway, February 5th 2001.
- [8] B. Yeomans. *Enhancing the World Wide Web*. Technical report, Computer Science Dept., University of Manchester, 1996. Supervisor: Prof. Brian Warboys.
- [9] P. S. Young. *The Teamware Language Reference Manual*. Teamware, March 1994.
- [10] M. V. Zelkowitz and D. R. Wallace. Experimental Models for Validating Technology. *IEEE Computer*, 31(5):23–31, May 1998.