

# Supporting Interactive Workflow Systems through Graphical Web Interfaces and Interactive Simulators

A. Bruno, F. Paternò, C. Santoro  
ISTI-CNR  
Via G. Moruzzi, 1  
56100 Pisa (Italy)  
+39 050 3153066

fabio.paterno@isti.cnr.it

## ABSTRACT

The design of workflow systems originated as an attempt to support coordinated data access. The improvement of interaction technology has created an opportunity for more flexible and interactive activities. Tools for modelling tasks in cooperative applications have started to appear. In this paper, we show how to extend one such tool for supporting workflow control in distributed environments through interactive graphical interfaces. In particular, we show how we have created an environment that exploits a workflow server containing a simulator of cooperative task models. This enables the possibility of allowing users with different roles to access the system through interactive graphical Web interfaces obtained using SVG. Users can access from any location where a Web access is available and obtain information regarding the state of a specific workflow instance, their enabled tasks according to the current state, and the history of tasks accomplished with details regarding their performance. Whenever a task is performed, the user can inform the system regarding this, and the simulator will update the state of the procedure and accordingly enable or disable each user's tasks for all the roles involved.

## Categories and Subject Descriptors

H.5 INFORMATION INTERFACES AND PRESENTATION

## General Terms

Management, Documentation, Design, Human Factors

## Keywords

Workflow Systems, Task Models, Interactive Simulators, Cooperative applications.

## 1. INTRODUCTION

In recent years workflow systems have become an important area, especially within companies and organisations, where they are expected to provide effective solutions to capture and formalise knowledge about current and desired work practices and monitor how related processes, such as document management and multi-

user cooperation, are carried out at the organisational level. The Workflow Management Coalition (WFMC) defines workflow as the automation of a business process, wholly or partly, during which documents, information, or tasks are passed from one participant to another for action, according to a set of procedural rules [9]. Thus, the strong motivation of workflow systems is to provide automatic support to organisation processes by incorporating the rules that define how activities should be carried out, how they should communicate/coordinate/synchronise each other, and how to manage the involved information objects while harmonising their solutions with existing operating systems, databases, software applications and hardware infrastructures. The expected outcome is that the activities that are unnecessary or too expensive/long or that might slow down the overall process are identified and possibly modified or even substituted with other more efficient ones.

It is worth pointing out that task modelling deals with similar concepts, as its objective is to identify the activities users are supposed to carry out in order to achieve their goals while using an interactive system. The ultimate aim is to use the information contained in task models to generate effective user interfaces accordingly.

Moreover, on the one hand workflow systems share a lot with task modelling, especially when methods based on task analysis are applied to multi-user applications, since both of them cope with identifying roles, specifying constraints between the activities carried out by the various roles for accomplishing their work, the information objects manipulated, and so on. On the other hand, two basic differences can be identified between them. The first one is the granularity of the considered activities: in workflow systems the focus is generally on entire processes at the organisational level, which might even have a long duration, whereas task analysis focuses more on users' interactions with applications, which in some cases can be almost instantaneous. The other difference is in the area from which they have developed: workflow systems originated in the area of databases, where their original aim was allowing different users to have access to the same information in a consistent manner, while task analysis is basically connected with human-computer interaction.

In our research we focus precisely on the possible synergies that can be found between these two approaches, by investigating how the concepts and tools developed from the perspective of one approach can be exploited for the purposes of the other. In particular, we present AGATA, an environment for controlling and monitoring workflow processes, which extends a well-known

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Conference '04, Month 1-2, 2004, City, State, Country.  
Copyright 2004 ACM 1-58113-000-0/00/0004...\$5.00.

tool for task analysis and modelling (CTTE) also developed in our group.

In this paper after giving some background information about CTTE, we provide a discussion of the related work and introduce the motivations for the tool. We describe its architecture and capabilities and then show an example of use of our system in a real organisational situation and conclude with some remarks and an overview of future research directions and development.

## 2. CTTE

### 2.1 Modelling Cooperative Tasks with CTTE

The ConcurTaskTrees Environment (CTTE, [3,4,5]) allows the designer to model cooperative applications by specifying cooperative task models, which contain one model for each role involved and another model for describing the co-operations among users belonging to different roles. Therefore, there is a task model for each role involved (single-user task model), whereas the cooperative activities occurring between the different roles are modeled by means of another separate model (the cooperative one), in which the global picture of how the activities performed by different roles have to interact/synchronize with each other is described.

So, the main purpose of the cooperative part is to indicate the relations among tasks performed by different users. Within the task model of each single role it is possible to identify the activities that, while carried out by the considered role, also affect the successful completion of a cooperative activity. Such activities, from the viewpoint of the single role, are identified as *connection* tasks. All the connection tasks appearing in each single-user task model are the basic tasks in the cooperative task model, where they are combined through appropriate temporal relations. Then, the cooperative task model provides an overview of how the multi-user activities coordinate each other, regardless of how complex each basic activity actually is, when it comes to be refined and actually carried out from the perspective of the single user. Thus, the cooperative part is specified by using the same notation as in the single-user part, the only addition is the possibility of specifying cooperative tasks, which are tasks that are performed in part by one user in part by another one. Thus, in the task decomposition of the cooperative tasks at some point we will reach the connection tasks associated with two or more roles.

### 2.2 The CTTE Simulator

One key feature of AGATA is the exploitation of the CTTE simulator, quite a successful feature of the tool that allows designers to better understand the dynamic behavior of the model specified, something which can be demanding especially with large specifications. The basic idea is that, at any time, the simulator shows the list of enabled tasks according to the temporal relations specified in the task model. In the list of enabled tasks, only basic tasks (those that are not further decomposed in the model) are considered, in order to provide the user with the most detailed view of the activities that can be performed at any time. Before starting the simulation, the tool automatically checks that the task model is complete and consistent. The designer can interactively select one task from the list and then the simulator shows what the next enabled tasks are after the performance of the selected task. In AGATA, this tool is

located in the server which is accessed by all the users involved in the workflow through Web interfaces in order to know what tasks they are enabled to do or to communicate what tasks they have accomplished.

## 3. RELATED WORK

In this section we are going to analyze a number of works that we judge relevant for the issues involved by workflow systems and related techniques.

A classical approach for workflow systems is Action Workflow [2], which is based on “speech acts”: the focus is on the coordination among persons rather than on the execution of tasks themselves. Indeed tasks are defined by a network of requests and commitments between the client and the executor, who interact each other according to a number of interaction modalities defined by the model. The basic unit of this model is composed of a loop with four elements: i) *proposal*: client and executor propose that the executor carry out the work. ii) *agreement*: client and executor reach an agreement iii) *execution*: the executor executes the work; iv) *satisfaction*: the client evaluates the work and the level of satisfaction. On the one hand, as far as the advantages of this model are concerned, AW model is good at highlighting the network of commitments and allows for easy identification of uncompleted workflow, together with the possibility to identify slowing down within specific steps. On the other hand, as AW model is stuck into networks of loops, it does not seem to provide the flexibility necessary to model real activities and, in addition, the tasks cannot be explicitly described with this model.

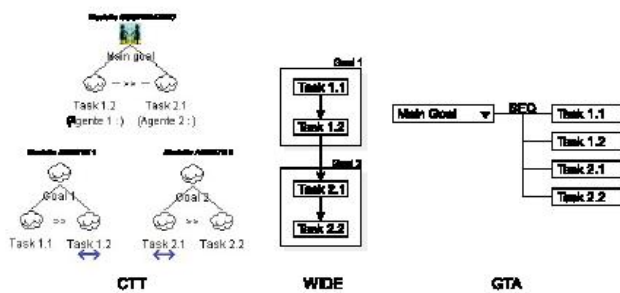
WIDE [10] is another approach for modelling workflow activities. It is based on three models: the Organisation Model, which includes the concepts of role, agent, team, etc., the Information Model, which refers to variables, form, documents, folders, the Process Model, which includes (different types of) *tasks*, the basic units of the work, and *connectors* (e.g.: fork, join, cycle, ..), which specify the order in which the tasks have to be carried out. With respect to Action Workflow, WIDE is good at putting more emphasis on the activities that should be performed by the roles rather than on the communications among them, and also includes additional, more sophisticated concepts like business transaction and multitask, while making provision also for mechanisms for exception handling (ECA: Event, Condition, Action). Moreover, WIDE models are well-structured but they do not allow users to represent a satisfactory number of temporal relationships that need to be expressed in realistic workflow situations. Instead, a special focus on representing realistic use of groupware systems is presented in the work of Pinelle and others [6], in which a modelling technique called Collaboration Usability Analysis is proposed to represent the range of ways that a group task can be carried out. Indeed, CUA allows variable paths through the execution of a task, and allows alternate paths and optional tasks to be modelled. CUA’s main contribution is to provide evaluators with a framework in which they can simulate the realistic use of a groupware system and identify usability problems that are caused by the groupware interface.

In the area of modelling groupware mechanisms Groupware Task Analysis [8] should be mentioned, as it is a method that

emphasizes on studying groupware systems and their activities rather than studying single users. It considers three different aspects: *agents*, which often indicate people, either individuals or groups, but may also refer to systems; *work*, which can be split into different tasks, with different levels of complexity, from the lowest task level people refer to their work, the 'unit task' (which is often role-related), and complex tasks, which may be split up between agents or roles; *situation*: the current situation for the performance of a certain task, which involves describing the environment (physical, conceptual, and social) and the objects in such environment.

Figure 1 shows how CTT, WIDE e GTA, even if they have different background, can express a similar model. Two agents, *Agent 1* and *Agent 2*, perform two tasks each, sequentially: *Agent 1* performs first "Task 1.1" and then "Task 1.2", while *Agent 2* performs first "Task 2.1" and then "Task 2.2". It is also important the order in which the tasks are performed by the two agents. In particular, the first task of *Agent 2* ("Task 2.1") should be performed after the second task of *Agent 1* ("Task 1.2").

By analysing the models it is possible to note some differences.



**Figure 1: Modelling tasks with CTT, WIDE and GTA**

The CTT model explicitly indicates on tasks, their performance order and the associated agents. In order to provide such information, three representations are provided, one associated with the cooperative part and one for each of the two agents involved. WIDE and GTA show all tasks in the same representation but they not indicate explicitly to what agent they are associated. This information is provided separately at the level of the single task description. CTT and GTA by representing the model in a tree-like description focus on the structure of the tasks. The representation adopted in WIDE allows designers to better highlight the execution flow in the model. In general, we note that in WIDE it is possible to express only a subset of the relations that can be in GTA or CTT.

In the need to pay attention on how the different activities are concurrently carried out we can find the motivations to analyse possible relationships between workflow and task models. For instance, in the work of Trøttestad [7] workflow models are identified as useful for group or organization interaction, while task models are judged basically focused on individual users. Then, he motivates integration of workflow and task modelling and present and compare workflow concepts with those used in task modelling, with suggestions for integrating them. In particular, he shows how Marshak's four dimensions[1] for describing workflow, i.e. action structure, actors, tools and information, have natural correspondences in the task modelling

domain and, in particular, he shows how a workflow model can be used as a starting point for a task model for a workflow participant.

Along the same direction our work can be positioned. We tried to integrate the two approaches and, in order to evaluate its feasibility we developed a prototype by extending a tool for task modelling so as to provide in it features that are properly geared to workflow purposes.

## 4. AGATA

### 4.1 Motivations

The growing spread of Internet has promoted the development of applications in which clients/users can cooperate in order to collaboratively carry out multi-user activities by exploiting the widespread and well-known Web-based protocol. The motivations underlying the development of AGATA can be traced to this: extending the CTT Environment by empowering it with workflow-related capabilities. This is obtained by exploiting the Web, since the inherently distributed nature of workflow systems (by definition they are expected to be carried out by different users from different locations) are highly suitable for support from web-based applications. In addition, the graphical user interface provided by AGATA has mainly been developed by using SVG, with the aim of being as independent as possible from the server and at the same time keeping the time for client/server communications at minimum. Indeed, an SVG object is a bare XML fragment which is rendered on the client machine: this mechanism allows both accelerating the loading process and relying on the capabilities of the client machine used for rendering the SVG graphic. This also enables the user to interact with the graphical Web representation of the model, in order to zoom in and out and perform other manipulations.

Through this technological support we have designed and implemented an environment in which it is possible to use task models of cooperative applications as specifications of the possible workflows, load many of them in the AGATA server, and allow the creation of multiple instances of each of them. When a model is activated it means that users can access it from any device that can connect to the server through Web browsers. In this way, they can be informed whether they have some task enabled and they can communicate when they have accomplished any task. The information of task accomplished is transmitted to the server, which calculates, with the support of the CTTE interactive simulator, the next set of tasks enabled, so that when users access it from any node they can have the updated list of enabled tasks.

### 4.2 The Architecture of AGATA

The architecture of AGATA (see Figure 2) is basically a distributed client-server, which provides network-based support for extending the original stand-alone CTT Environment. The architecture is centrally based on a HTTP server devoted to handling the activities that are supposed to be carried out by the different roles. More specifically, the core part of AGATA manages the communications with clients and also supports the exchange of information and data with the CTTE simulator, which works as the system engine. Indeed, depending on the current state of ongoing activities performed by the clients, such

engine is able to calculate the next activities to be performed by each role and update the various client views accordingly.

It is worth pointing out that it may occur that the activity of one role has to wait for another activity to be executed by a different role. In this case, the client gets an empty pool of tasks, and, as soon as the concerned activity is carried out by the other role, the server updates the current state and shows an updated list of tasks to be done. The client side requires a suitable browser able to interpret SVG scripts, so as to properly render the output provided by the server.

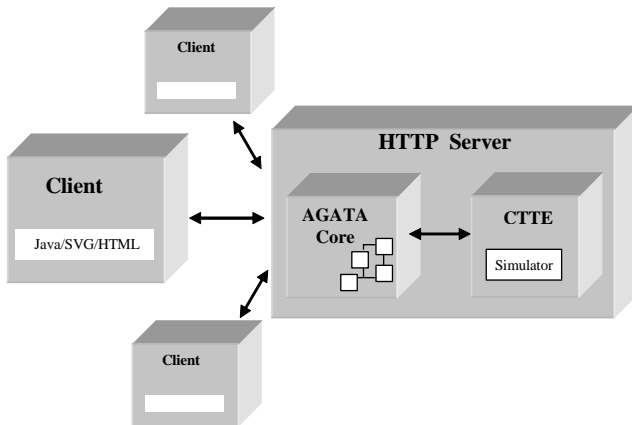


Figure 2: The Architecture of AGATA

In the next section we will provide the reader with an overview of the features AGATA offers to the clients/users for supporting effective interactions in the workflow process.

### 4.3 The AGATA User Interface

AGATA's user interface is composed of two main modules: the Model Manager and the Workflow Manager. The AGATA panel presenting both of them is visualized in Figure 3.

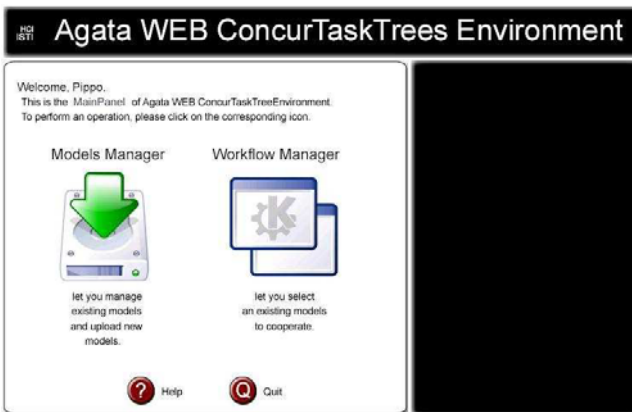


Figure 3: The Main Panel of AGATA

#### 4.3.1 The Model Manager

The Model Manager is a file manager that allows the user to handle the models. The user is expected to interact with this module first, in order to select the models of interest. One of the main responsibilities of this module is to check the available

models according to the requirements of AGATA and possibly discard models that do not satisfy them. Such constraints are, for instance, that the client requesting to participate in the cooperation is authenticated, the session is valid, the IP of the client is trusted, the syntax is corrected, etc. If there are requests that are not valid they are deleted, while the other ones are encapsulated in an appropriate data structure associated and then passed on to the specific handler, which is also in charge of identifying and possibly formatting the output to be delivered to the user.

#### 4.3.2 The Workflow Manager

The Workflow Manager is in charge of handling the cooperations, managing the actions performed by the different client/users and updating the workflow accordingly. The core part of the manager can be reached by means of a Wizard, which asks the users to provide the information necessary to define the terms of their participation within the cooperation, for instance, which instance of a model the user would like to join and in what role. This does not mean that AGATA limits a given user to only one cooperation, since it is possible that the same user cooperate and participate in different workflow models at the same time. When the user handles the workflow manager, s/he is able to execute a task, visualize a graphical representation of the task model itself, get information about tasks, visit the task model, etc.

## 5. AN AGATA SESSION

At the beginning of the interactive session, in the Model Manager a login screen is presented to the user. This panel allows users to select a cooperative model they are interested in, and which is associated with a specific workflow. The user can either directly select a cooperative task model from the list presented, or select another model of interest from the local disk. Then, in any case, the user is ready to start a cooperation by closing the Model Manager (see related button in the window). The user will thus be redirected to the beginning of the Workflow Manager, so as to begin interacting with the system.

The cooperation can begin when the starter role (namely the first role with enabled tasks) performs the necessary actions to activate the other roles. In order to guide the user within the cooperation, AGATA provides a Wizard within the Workflow Manager, which visualizes a set of cooperative models (previously loaded from interacting with the Model Manager) and requires the user to select the model to participate in (see Figure 4).

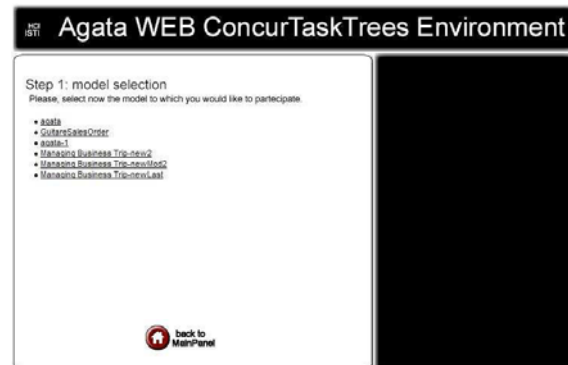


Figure 4: Selecting a Model through the Wizard

Then, since for each cooperative model it is possible to have more than one instance active at any time, it is required from the users to select to which *instance* of the model they would like to participate (indeed, it is possible to have more than one instance active at the same time). If no instance exists, it is possible to create a new one. As soon as a new instance is created, it is available to the other roles that have selected the same model. Each instance is identified by means of the corresponding model, the creation time and the role that created such an instance (see Figure 5). At the next step, the user is required to specify the role that the actor wants to play, since each role has to be univocally assigned.

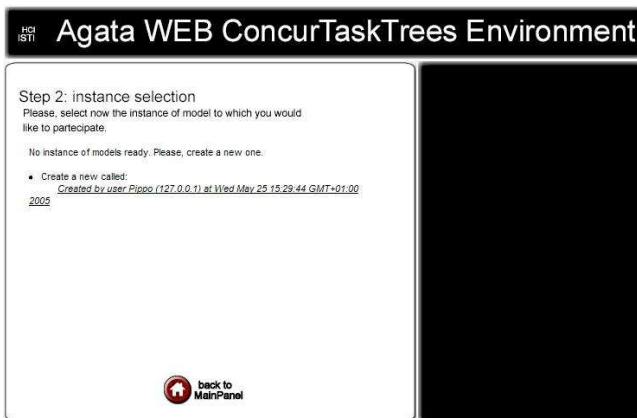


Figure 5: Selecting a Model Instance through the Wizard

Lastly, the Wizard proposes a summary of the currently selected options that the user can still change, otherwise the cooperation can start through the Workflow Manager. The window of the Workflow Manager is split into different panels, containing various information. The panel on the top-left part (see Figure 6) contains information about the user (id and name of the user, information about the model instance and the role played by the user). From this screen onward, each graphical object is realized with SVG, so it is necessary that the user browser has a dedicated plug-in installed in order to handle SVG scripts. During the cooperation a graphical representation of the current task, namely the most recent task that has been executed by the user, and then it provides information about the current state of the cooperation.

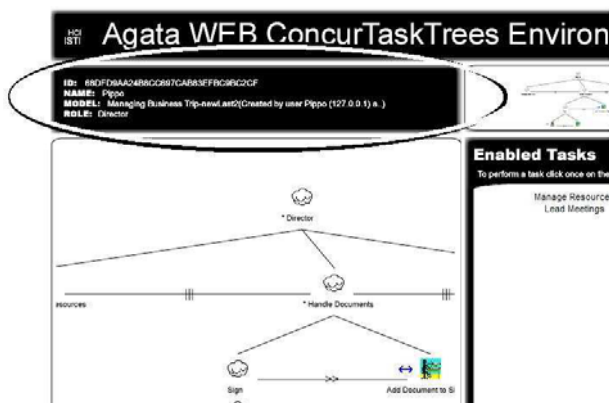


Figure 6: Role-related Information in the Workflow Manager

In the top-right part there is an overview panel, in which a zoomed-in visualization of the model is provided, so as to allow

the user to have immediately an (interactive) overview of the model (see Figure 7). The overview panel has been provided with interactive capabilities as well since, especially with large models, it can make easier moving around the different areas of the models.

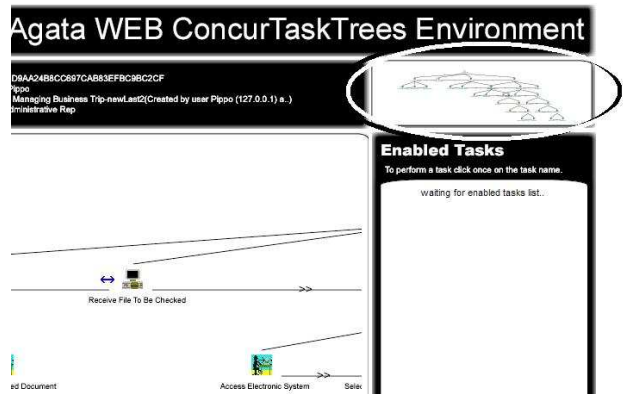


Figure 7: Visualising an Overview of the Task Model

Centrally to the workflow manager there is the visualization of (a portion of) the task model in which the user is currently interested (see Figure 6 and 7). The user is able to move around the model with zooming capabilities by interacting with the user interface, so as to highlight the current area of interest.

Finally, on the bottom-right part of the window there is the panel with the list of tasks that are currently enabled, which means the set of tasks that can be executed by the specific role in the current state of the cooperation (see Figure 6 and 7). Then, from the viewpoint of a specific role, it is possible to know the task model describing the activity that s/he is supposed to perform, together with the list of tasks that s/he is supposed to carry out at a specific moment of the cooperation: such tasks may be locally dependent (depend on the tasks that s/he has already completed), but may be also dependent on the tasks that the other roles have carried out (globally dependent), according to the information provided within the cooperative task model. On the bottom of the Workflow Manager there is a panel gathering the list of all the tasks that all the cooperating users have carried out up that moment. The visualization is provided in such a way to easily spot the actions that have been carried out by the specific role (a different character style is used).

It is worth pointing out that, in order to speed up client/server communications, the requests that the Workflow Manager receives are handled in an asynchronous way. It means that as soon as a request from the client is received (which means, in workflow-based terms, that a task has been executed by a client) it is queued (according to a FIFO strategy) by the server, while the resources needed are immediately released. Only afterwards the actual answer calculated by the engine (namely, the new set of enabled tasks) will be provided to the user.

The calculation of the roles that are available at any time to the user is done dynamically at runtime by the system, in order not to offer roles that have been already selected and guarantee that the selection will be univocally carried out. In addition, it is possible to save the current situation of actions performed by the different users (see the "Save" button).

## 6. AN EXAMPLE OF APPLICATION

We used as case study an example of how business trips are handled in our research organization. We identified four basic roles that play part in this cooperation. The main role is played by the Researcher, who is expected to fulfill a number of requests for issuing the order, both in an electronic form and in a paper-based manner, before leaving for the trip and after coming back as well. First of all, in order for the requests to be accepted by the director of the organization, the researchers need to get the request approved by their lab head, who is expected to supervise on the availability of the budget to cover the travel expenses. Then the request reaches the administration representative, who is in charge of formally submitting the request to the director (so as to have it officially signed) and, at the same time, to check if the electronic request previously submitted by the researcher and approved by the lab head is consistent and compatible with the view they have of the current financial situation of the lab.

It is worth pointing out that it is needed that the lab head approves the request for making the administration rep be able to check the request, as the paper-based request is only destined to the director. Then, until the lab head approves the form, the administrative rep cannot physically see the request, as the researchers' requests become available to the administration only after having been electronically approved by the corresponding lab head.

As soon as the paper-based and the electronic approvals have been received, the researcher is formally authorized to leave. After the trip, the researcher fill in a form for requiring reimbursement of the travel expenses and sends it to the administrative rep who is in charge of checking if all the expenses are allowed. After approval has been got they proceed with the reimbursement accordingly. For sake of brevity, Figure 8 shows only the cooperative part of the task model of the considered application.

AGATA is able to manage this workflow model obtained with the CTT notation and editor. The model involves four roles and where the people associated with each of them can be located anywhere as long as a Web access is available. In Figure we have shown the task models built for modelling the activities of the considered case study: as you can see, it includes four single user task models and a cooperative one for specifying how the different roles coordinate.

Then, in order to start using AGATA, the first step is identifying the role characteristics, which will be performed through a multi-step Wizard available in the tool. In Figure 9 the panel of the Wizards showing the possible roles of the cooperative model of the case study is presented. As you can see, the user can select the role that s/he would like to play, amongst those available and automatically calculated by the engine. If the user is the first one to select to participate in a certain instance of the model, s/he has the complete choice of roles. As soon as a role is selected, it will be made unavailable to the other roles.

Assuming that the role selected is the Lab Head, at the beginning of the lab head session (see Figure 10) the user can perform three tasks in an unspecified order, leading meetings in the laboratory (*Lead Lab Meetings*), access the electronic administrative tool (*Access GECO*, from the name of the tool) and managing the resources in the Lab (*Manage Lab Resources*). It is worth pointing out that, until no actual electronic request from the personnel is submitted through GECO, the lab head can only monitor the current situation and quit the system. Only after a researcher in the lab submits a request, a new task will be enabled in the list of the lab head, requiring from him/her appropriate check on the related information.

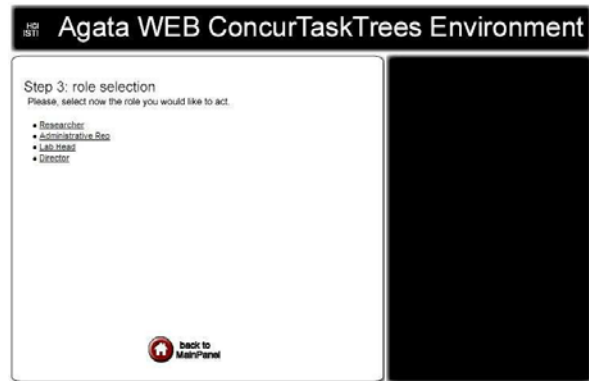


Figure 9: Role selection.

On the one hand, it is possible to note that, although the researcher has already filled in some fields of the electronic form (see the scenario that has been produced till the current moment and visualized in the bottom part of Figure 10), the lab head still

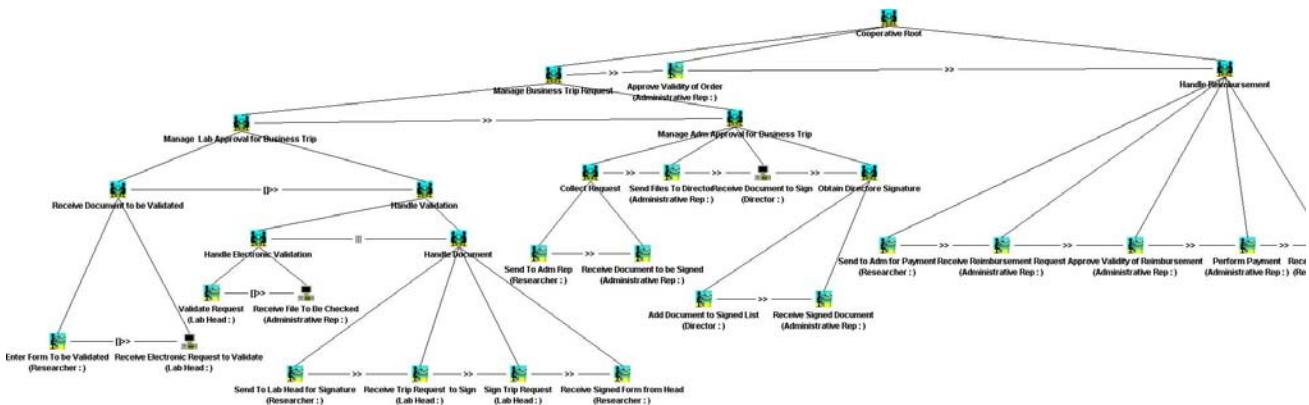


Figure 8: The Cooperative Part of the Case Study

keeps an unmodified set of enabled tasks, because it is only the completion (and related electronic submission) of the form by the researcher that actually enables the management of the form in the enabled task set of the lab head. On the other hand, the lab head is aware of the actions that have been already performed by the other roles on the model.

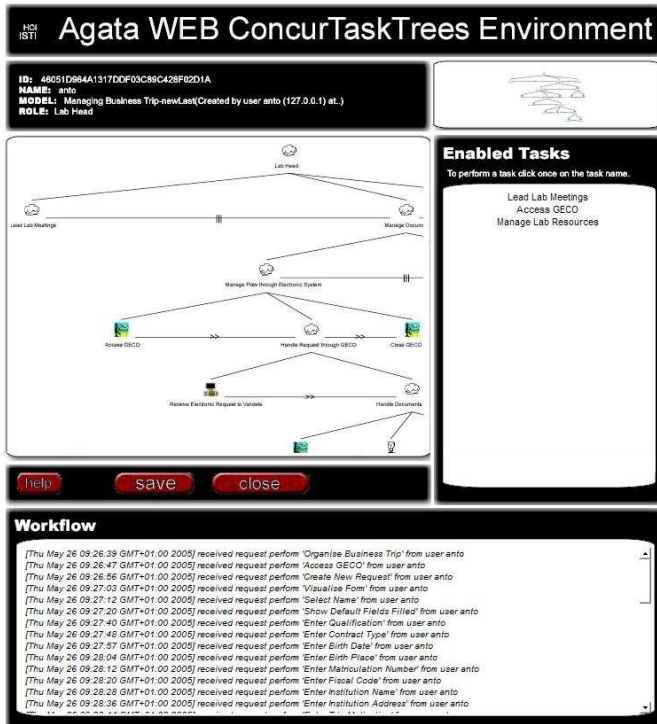


Figure 10: Interacting with the Workflow Manager

## 7. CONCLUSIONS AND FUTURE WORK

In this paper a system for supporting workflow through interactive Web graphical interfaces is described. The system has been developed by extending a previous tool for specifying and analysing task models (CTTE), so as to provide support for graphical rendering of the process managed by a workflow system. Web-oriented technologies and languages (like JSP and SVG) have been used for providing a graphical user interface to the users.

The system has been developed in a highly modular way. In addition, the use of SVG for the graphical user interface allows a great scalability for the layout, as well as the almost total independence of the platform because it is only needed a suitable plug-in for the rendering of the SVG snippets on the client machine. In addition, the peculiarity of this XML-based language ensures a fast download of the SVG fragments of code (much smaller than loading gif/jpg images), which allows the creation of quite complex layout without compromising the time requested for the loading.

Further research should be focused on various types of flexibilities that may be desired in workflow management applications. For instance, dynamic (re)allocation of activities, as well as advanced feature to spot the areas of the workflow that need improvement from an efficiency point of view.

## 8. REFERENCES

- [1] Marshak, R.T., Workflow: Applying Automation to Group Processes. In: Coleman, D. (ed.): Groupware - Collaborative Strategies for Corporate LANs and Intranets. Prentice Hall PTR (1997) 143-181
- [2] Medina-Mora, R., Winograd, T., Flores, R., Flores, F., The action workflow approach to workflow management technology, Proceedings of the 1992 ACM conference on Computer-supported cooperative work, CSCW 92, Toronto, Ontario, Canada.
- [3] Mori G., Paternò F., Santoro C., "CTTE: Support for Developing and Analysing Task Models for Interactive System Design", IEEE Transactions on Software Engineering, IEEE Press, Agosto 2002, Vol. 28, No. 8, pp. 797-813.
- [4] Paternò, F., Model-Based Design and Evaluation of Interactive Application. Springer Verlag, ISBN 1-85233-155-0, 1999.
- [5] Paternò, F. Santoro C., Tahmassebi, S. "Formal Models for Cooperative Tasks: Concepts and an Application for En-Route Air Traffic Control", Proceedings DSV-IS'98, pp.71-86, June'98, Abingdon, Springer Verlag.
- [6] Pinelle, D., Gutwin, C., Greenberg, S. Task analysis for groupware usability evaluation: Modeling shared-workspace tasks with the mechanics of collaboration Pages: 281 – 311, ACM Transactions on Computer-Human Interaction, December 2003
- [7] Trætterberg, H.: Modeling work: Workflow and Task modeling. In: Vanderdonckt, J., Puerta, A.R. (eds.): Proc. of 3rd Int. Conf. on Computer-Aided Design of User Interfaces CADUI'99 (Louvain-la-Neuve, 21-23 October 1999). Kluwer Academics, Dordrecht (1999) 275–280.
- [8] van der Veer, G., Lenting, B., Bergevoet, B., "GTA: Groupware Task Analysis - Modelling Complexity", Acta Psychologica, 91, pp. 297-322, 1996.
- [9] WfMC. Workflow Management Coalition Terminology & Glossary, WfMC-TC-1011, Document Status- Issue 2.0,
- [10] The WIDE Workflow Model and Language. WIDE Esprit Project 20280, Deliverable 5.1.2