A collaborative multi-agent based workflow system

Bastin Tony Roy Savarimuthu and Maryam Purvis

Department of Information Science, University of Otago,
P O Box 56, Dunedin, New Zealand
{tonyr,tehrany}@infoscience.otago.ac.nz

Abstract. Agent based workflow systems are being discussed by many researchers for a long time. In this paper we present a new collaborative approach for workflow systems. Agents collaborate by forming social network such as societies. Various kinds of societies can exist and that facilitates co-operation and a communal effort towards group oriented problem solving. This framework would help us study the social behavior of real life resources represented by the agents in an agent-based workflow system. The societal framework for the workflow management system would be built upon our existing multi-agent based workflow management system.

1 Introduction

Most of the commercially available workflow management systems do not offer sufficient flexibility for distributed organizations that participate in the global market. These systems have rigid, centralized architectures that do not operate across multiple platforms ([2],[6],[11]). Employing a distributed network of autonomous software agents that can adapt to changing circumstances would result in an improved workflow management system. In the past, WfMS were used in well-defined activities, such as manufacturing, where the processes tend to be more established and stable. But in the current climate WfMS may be used for more fluid business processes, such as e-commerce, or in processes involving human interactions, such as the software development process. In such situations, it is not always possible to predict in advance all the parameters that may be important for the overall processes. This gives rise to the need of adaptive systems. Our previous works ([1], [7], [10]) describe the advantages of our agent-based framework JBees, such as distribution, flexibility and ability to dynamically incorporate a new process model.

In this paper we propose an architecture by which these agents form societies and abide social norms. They work towards their mutual benefit and towards the group goal at the same time. The paper is organized as follows. A brief description of our agent-based framework is given in Section 2. Section 3 describes our architecture for the collaborative agent based workflow society. In the Section 4 we explain our framework with an example. The concluding remarks are presented in Section 5.

2 Background

In this section we explain the background of our work. We explain the architecture of the existing multi-agent based workflow system.

2.1 Existing architecture

Our research is focused on developing an agent-based WfMS, where the work associated with running a WfMS has been partitioned among various collaborating agents that are interacting with each other by following standard agent communication protocols [4]. JBees is based on Opal [5] and uses the CPN execution tool JFern [3]. The processes are modeled using coloured Petri nets ([8],[9]). A first description of JBees can be found in the previously published papers [1] and [10]. Our enhanced system consists of seven Opal agents, which provide the functionality to control the workflow. Figure 1 shows these seven agents and their collaboration.

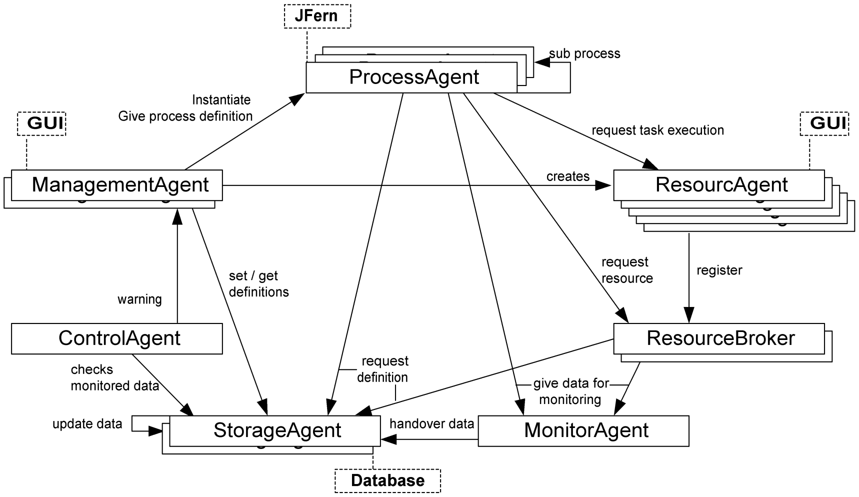

Figure 1: The architecture of JBees

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 monitor agent collects all the process specific data and sends them to the storage agent. The control agent continuously looks for anomalies to the criteria specified by the human manager and reports the violations to these criteria to the manager agent. The manager agent provides information to the human manager, which can be used for a feedback mechanism.

3 Architecture of the collaborative agent based workflow system

In order to achieve the goals of a collaborative environment in a workflow system, the agents in the system form societies. The inspiration for this work has been drawn from [12], [13], [14] and [15]. The proposed architecture of the system is shown in Figure 2. The new architecture would include the libraries that specify the norms of the society. Each society will comprise of its own norms and rules. There would be a repository for ontologies specific to the workflow system as well as repositories for domain specific application (software development, medical information system etc). Each resource in the society has certain capabilities. The star that encompasses certain resources indicates that these resources possess similar capabilities within a given society. Assume that resource brokers allocate resources depending upon the manager's request, the resource brokers negotiate with the resources to identify and allocate the best possible resource that suits the task to be performed.

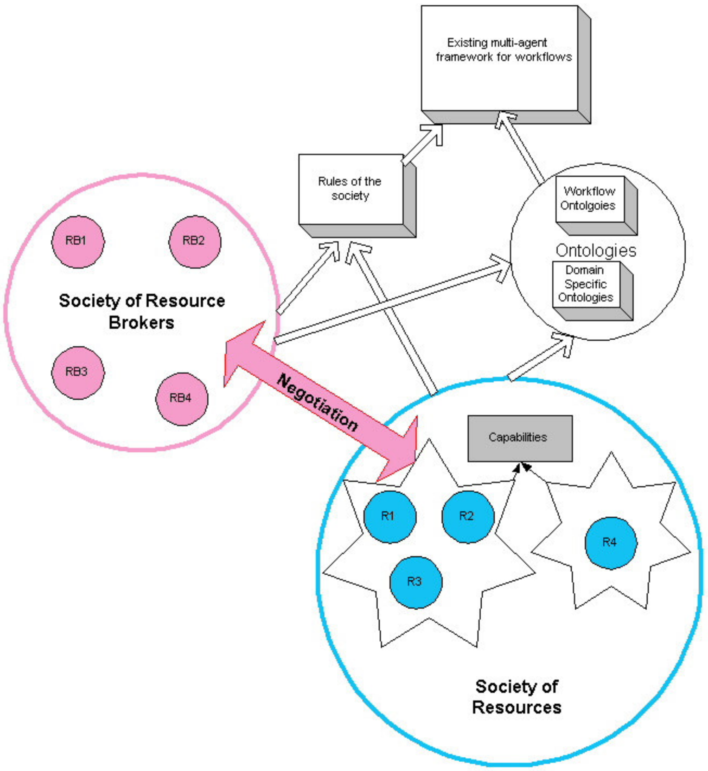

Figure 2: The architecture of the collaborative multi-agent based workflow system

3.1 Workflow Society

Workflow society comprises of resources to perform various tasks. The resources are designated to certain roles. Each role includes certain attributes. For example, a programmer may have the attributes such as A) Qualification (Bachelors/Masters) B) Years of experience (2,3 etc) C) Quality of work done (low, medium, high) D) Team Work (Cooperative, Individualistic) E) Efficiency (Lines of code per day)

These attributes are of two types, generic and specific. The first four of the attributes defined above are of generic type and the efficiency can vary depending upon the role. Specific attributes for a resource which is required to drive a car could include how safe the his/her driving is, the average number of kilometers the driver drives per day, while a programmer can have specific attributes such as his/her problem solving ability and lines of code (LOC) developed per day.

3.2 Capabilities

The workflow manager can specify the minimum level of skills requirement for a given task at hand. This is specified through a combination of attributes. Assuming that there are two programming agents. One of them can be a reliable, experienced and a team player and the other agent could be a less reliable, inexperienced but can be innovative. Depending upon what the situation demands, the workflow manager may want to choose the person with the appropriate capability. The current system provides a mechanism that identifies these groupings, which an agent can be a member of.

3.3 Ontology

There are different levels of ontologies, which are possible. In our case, we could have the workflow specific ontology and application specific ontology. Workflow specific ontology includes resource, process, tasks, etc. and the application specific ontology for software development domain, which include testing, debugging, maintenance etc.

3.4 Negotiation

During the allocation of resources in the context of workflow, we have the option of choosing a specific resource depending upon certain desired capabilities. Based on the negotiation mechanism, it is determined what is the best available resource given certain factors such as time, cost and availability constraints.

3.5 Co-operation models

Normally, the interaction between various agents in a society may follow different models of cooperation such as Market, Network and Hierarchical, as mentioned in [15]. Workflow systems fit in the Hierarchical model where agents are usually cooperative and oriented towards common global goal. For dynamic workflow processes there is a need for more networked agents, which negotiate with each other in order to achieve both group goals as well as personal goals. The Network model encompasses the self-interested nature of the Market model as well as the cooperative nature of the Hierarchical-model. In our system we are moving more towards the network model, which involves the interaction between the resource brokers and the individual agents. The resource agents are guided with their self-interest while the resource brokers are looking after the overall goal of the process model.

3.6 Inter-societal cooperation

In larger enterprises there exist different kinds of societies such as managers society, workers society etc. Interactions between these agent societies are often desirable for known advantages such as mutual benefits. There may also be some agents, which belong to both societies. These agents can be a 'linking-pin' between both societies as mentioned in [16] thereby forming a larger society. If the domains of these societies are different the "linkers" can act as "translators" for both of these societies.

4 An example – Software development environment

As an example to demonstrate the various aspects of our system, we use the well-known software engineering development environment as an example. A software firm consists of many levels of resources such as project managers, domain specialists, architects, programmers, testers etc. To simplify the scenario, we consider managers, programmers and testers for this example. A society of mangers is formed by the "role" played by the manager. There is an entry-level check for a manager to join this society. The manager has to meet the requirements in order to join the society. When the manager agent joins the society, it is expected to obey the rules laid down by the society. It is similar to the social obligations that a person must meet in the real world. Similarly the programmers and testers form their own respective societies.

Figure 3 shows the overview of the societies in a software development environment. The circles represent a particular type of societies such as testers, programmers and managers. The agents that adhere to the rules laid down by the society form this society. The agents can have different capabilities such as reliability, having obtained a level of competency such as number of defects/LOC, experience, skill sets, etc. The agents found within these stars have similar capabilities. In the given scenario, the agents have two different capabilities, the reliable agents (shown inside solid star shapes) and the non-reliable agents (shown inside stars that have dotted borders). The manager agent negotiates with the society to obtain the best possible programmer/tester depending upon the task at hand. This negotiation can be as simple as specifying certain attributes and obtaining the appropriate performer.

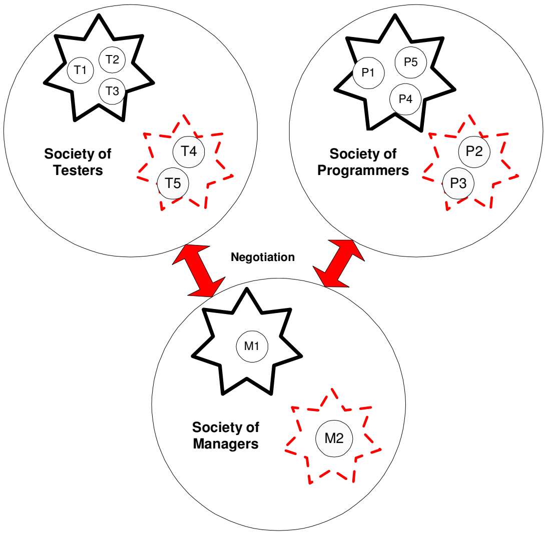

Figure 3: The interaction between the societies

Currently the characteristics are specified in terms of XML attributes for every resource. There exists a match making mechanism, which uses a simple search algorithm to identify the list of resources that possess certain capabilities. The resource brokers are the matchmakers, which identify the list of suitable resource agents. We are planning to maintain the history of the work done by each resource so that a more sophisticated resource allocation can be achieved based on the resource's experience and how well the resource has performed previously.

5 Conclusion

We have presented the architecture of our agent-based society, which is built upon our multi-agent workflow system, JBees. This enables us to define society specific agent capabilities and negotiation mechanism for resource allocation. Our framework allows for the building of domain specific ontology. We have also presented through an example how the architecture will be suitable for a particular domain such as software development environment. We are currently implementing this agent based societal architecture on top of our existing system.

References

1. Martin Fleurke, Lars Ehrler, and Maryam Purvis, 'JBees - an adaptive and distributed framework for workflow systems', in Workshop on Collaboration Agents: Autonomous Agents for Collaborative Environments (COLA), Halifax, Canada, eds., Ali Ghorbani and Stephen Marsh, pp. 69–76, http://www.cs.unb.ca/˜ ghorbani/cola/proceedings/NRC-46519.pdf, (2003). National Research Council Canada, Institute for Information Technology.
2. S. Meilin, Y. Guangxin, X. Yong, and W. Shangguang,' Workflow Management Systems: A Survey. ', in Proceedings of IEEE International Conference on Communication Technology, (1998).
3. Mariusz Nowostawski. JFern – Java based Petri Net framework , 2003.
4. FIPA, FIPA Communicative Act Library - Specification. 2002. http://www.fipa.org/specs/fipa00037
5. Martin K. Purvis, Stephen Cranefield, Mariusz Nowostawski, and Dan Carter, 'Opal: A multi-level infrastructure for agent-oriented software development', The information science discussion paper series no 2002/01, Department of Information Science, University of Otago, Dunedin, New Zealand, (2002).
6. J.W. Shepherdson, S.G. Thompson, and B. Odgers, 'Cross Organisational Workflow Coordinated by Software Agents', in CEUR Workshop Proceedings No 17. Cross Organisational Workflow Management and Coordination, San Francisco, USA, (1998)
7. Department of information science University of Otago. JBees. http://jbees.sourceforge.net, 2004.
8. W.M.P van der Aalst and K. van Hee, Workflow Management: Models, Methods, and Systems , MIT Press, 2002.
9. Jensen, K., Coloured Petri Nets - Basic Concepts, Analysis Methods and Practical Use, Vol. 1: Basic Concepts. EATCS Monographs on Theoretical Computer Science. 1992, Heidelberg, Berlin: Springer Verlag GmbH. 1-234.
10. Savarimuthu, B.T.R., Purvis, M. and Fleurke, M. (2004). Monitoring and Controlling of a Multi-agent Based Workflow System. In Proc. Australasian Workshop on Data Mining and Web Intelligence (DMWI2004), Dunedin, New Zealand. CRPIT, **32**. Purvis, M., Ed. ACS. 127-132.
11. J.W. Shepherdson, S.G. Thompson, and B. Odgers, 'Cross Organisational Workflow Coordinated by Software Agents', in CEUR Workshop Proceedings No 17. Cross Organisational Workflow Management and Coordination, San Francisco, USA, (1998)
12. Dastani, M.M., Dignum, M.V., & Dignum, F.P.M. (2003). Role-Assignment in Open Agent Societies.. In (Ed.), *Proceedings of the Second International Conference on Autonomous Agents and Multiagent Systems (AAMAS'03)* Melbourne: ACM Press.
13. Mario Verdicchio, Marco Colombetti: A Logical Model of Social Commitment for Agent Communication. Workshop on Agent Communication Languages 2003: 128-145
14. Marco Colombetti, Nicoletta Fornara and Mario Verdicchio.(2002) The Role of Institutions in Multiagent Systems *Ottavo Convegno Associazione Italiana per l'Intelligenza Artificiale AI*IA*, Siena, Italy
15. Dignum, V, Weigand, H 'Toward an Organization-Oriented Design Methodology for Agent Societies' Intelligent Agent Software Engineering. 2003, London,Idea publishing group. 191-212.
16. Huang, P. and Sycara, K, 'A computational model for online agent negotiation ', in Proceedings of 35th Hawaii International Conference on System Sciences, (2002).