A Framework for Business Process Integration - An Agent-mediated Approach

Zheng Zhao, Virginia Dignum and Frank Dignum

Department of Information & Computing Sciences, Utrecht University, The Netherlands

Abstract. Coordination of business processes within and across organizations attracts more and more attention because of the growth of e-commerce and the implicit boundaries of organizations. Current approaches are not flexible enough to support decision making concerning Business Process Integration (BPI) solutions that take into account economic, social, and technical aspects. In this paper, we will analyse the problems that currently exist and propose an agent-based framework for mediation. We identify the requirements of this agent-mediated framework and argue the advantages of using self-adaptive agent organizations as model for mediation.

1 Introduction

Nowadays executing a typical business process may involve several different resources, databases, application systems and business rules, which can be either located within or across organizations. In order to execute a business process properly, organizations must deal with problems of heterogeneity of interfaces, ontology, communication languages, and interaction protocols. Therefore, coordination of such business processes becomes an important and non-trivial issue. Many efforts have been made to develop middleware for integrating and automating enterprise business processes over the past decade [1], however, current Business Process Integration (BPI) solutions (such as Tibco , Webmethods, Seebeyond, ebXML, IBM websphere and Cordys) are not flexible enough to support decision making that take into account both economic and technical aspects. Most systems consider only the technical aspects, such as the synchronization of workflows, resource retrieval, etc. However, economic and social aspects also play very important roles in system design. A system is attractive to users (organizations) only when the users can benefit by using it. Moreover the benefit one can get has to be shown before the user decides to join the system. Furthermore, coordinating business processes across organizations needs to consider the independent nature of each organization. In order to guarantee the cooperative outcomes, social aspects such as organizational culture and norms have to be taken into account.

Two fundamental architectural choices for process integration can be recognized. One is that parties interact directly, which is also called unmediated communication. The other is to use a third entity that mediates or supports the communication. The latter approach has several names, such as orchestration (Microsoft), choreography (SAP/Sun), brokering or mediation. We will use the term “mediation” in this paper.

Zhao Z., Dignum V. and Dignum F. (2007).
A Framework for Business Process Integration - An Agent-mediated Approach.
In *Proceedings of the 4th International Workshop on Computer Supported Activity Coordination*, pages 13-24
DOI: 10.5220/0002412100130024

One important advantage of mediation is that participants do not need to know each other or each other's standards since the central party is capable to exchange information among them. Of course, mediation also has its drawbacks, i.e. it brings overhead costs and needs a proper IT governance solution to manage it. Therefore, it is important for our research to clearly define situations where mediation is an advantage.

From the agent perspective, the term "mediator" was probably used in Computer Science for the first time in [2] for active middleware that mediates between the users' workstations and data resources. The mediation used here focused on data and was unidirectional. Nowadays, we often read about mediators or broker agents in the field of Multi-agent systems. However, it is rarely discussed in-depth whether they are really needed and under what circumstances they are needed or whether they are preferable to other solutions.

Multi-agent systems (MAS) are widely considered as suitable abstractions to model coordination and communication issues because of the characteristics they exhibit [3]:

– Are typically open and have no centralized designer;
– Contain autonomous, heterogeneous and distributed agents, with different 'personalities' (cooperative, selfish, honest, etc.);
– Provide an infrastructure to specify communication and interaction protocols.

In this paper, we analyze what types of mediators are really needed for what kind of situations and propose an agent-mediated framework. In Section 2, we give an overview of existing techniques. In Section 3, we investigate the requirements of Agent-mediated Business Coordination (ABC) method. Scenario for the ABC framework is presented in Section 4. Section 5 shows the different types of mediators. And we conclude in Section 6 with the direction of our future work.

2 An Overview of Current Techniques

In this section, we will give an overview of current existing mediation techniques based on agent technology. We don't mean to make exhaustive comparisons with these techniques from different aspects, but give an introduction to their working mechanisms.

2.1 TuCSoN

TuCSoN [4, 5] is an agent coordination infrastructure as well as a model for the (objective) coordination of Internet agents, particularly suitable to mobile information agents. The TuCSoN model is based on the notion of (logic) tuple centres, each of which is an independent interaction space that abstracts the role of the environment. Tuple centres are coordination artifacts that mediate and govern agent interaction according to coordination laws that define their reactive behavior. For example, consider a scenario with three tasks A, B, and C, which must be coordinated according to the following workflow rule: task C can start only when both task A and task B have been successfully completed. In order to provide this specific coordination service, the tuple centre is programmed by a set of reactions, i.e. agents that are responsible for executing task

A and task B provide the result of a task by inserting a tuple *task_done(...)* after the task is executed. When both *task_done(...)* tuples of task A and task B are inserted into the tuple centre, a tuple *task_todo(...)* for task C can be made observable by the tuple centre. The agent that is responsible for task C can then be aware of the task to execute by observing the *task_todo(...)* tuple and start executing it by retrieving this tuple from the tuple centre.

TuCSoN is developed in Java and directly supports the integration between heterogeneous information sources. Its tuple centres are powerful coordination abstractions that act as intelligent information mediators that map knowledge to knowledge. Because of this, participants do not need to change their model of knowledge representation and new participant can be easily integrated dynamically. This dynamic behavior specification is also the key property that allows TuCsoN to support heterogeneity at the process level. It stores coordination rules outside the interacting entities, and as such governs their interaction in a predictable way.

From the technical point of view, TuCSoN encapsulates the workflow rules as coordination laws and separates the workflow participants from the workflow engines, thus effectively achieves the dynamic coordination among participants.

2.2 JBees

JBees [6] is a prototype Workflow management systems (WfMSs) that is based on agent technology. It provides a mechanism for communication of distributed components in order to support inter-organizational WfMS. JBees is based on Opal (a Java-based agent platform) [7] and uses the Coloured Petri Net (CPN) execution tool JFern [8]. The idea of JBees is that 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. The system consists of seven Opal agents that provide the functionality to control the workflow:

– Management agent: It provides the user interface for the human workflow manager;
– Storage agent: It manages the persistent data of the workflow, such as the definitions of tasks, roles and processes, and the monitored data. It also notifies all management agents if the data has been changed;
– Process agent: It is in charge of executing one particular case. A new process agent is created for each work case and for each sub-work case. The process agent uses JFern to execute the CPN model provided by the management agent or by the "parent" process agent;
– Resource agent: It is the user interface for the human resource or the interface for tools such as printers and scanners;
– Resource broker agent: It is responsible for the resource management;
– Monitor agent: It gathers the data in the system that is necessary to analyze workflows and sends the data associated with a particular case to the storage agent for persistent storage after the case is finished;
– Control agent: It provides the feedback mechanism required for process reengineering.

By combining the CPN-formalism and the collaborating software agents, JBees provides a high level of flexibility and adaptability. Its monitoring and feedback mechanisms also make it possible to study and analyze the various processes and cases.

JBees guarantees process consistencies at each step of the process execution, thus technically provides the basic executing elements of coordinating multiple business processes.

2.3 mPower

mPower [9] is a component-based layered framework developed for easing the development of multi-agent systems. It builds on the JADE-LEAP platform that provides a homogeneous layer over diverse operating systems and hardware devices. In mPower, a business process is viewed as a linked set of conversations among participating process actor roles. The services for executing business tasks are provided by conversational components (C-COMs) via interaction with other agent roles. A C-COM consists of two main building blocks, which are an interaction protocol and the role components. The interaction protocol defines the sequence of asynchronous messages sent between the role components, and the role components perform the actions necessary according to the interaction protocol to achieve the service goal. For each C-COM, there are two generic role components – initiator (starts by sending a message) and respondent (activated when receiving a message).

The mPower framework consists of four layers – foundation, components, generic workflow and applications. The foundation layer contains the supporting functionality, such as message transportation, ontology support, language support, etc. The components layer consists of ontology components and service components (C-COMs). The ontology components map the common ontology items into the hierarchy's predefined categories and detail the attributes of the items in target domains, whereas C-COMs abstract and implement the common message-based interactions between participating agents in target domains. The generic workflow layer provides architectural patterns that can be used as templates to automate domain- or organization-specific business processes. The application layer consists of customized collection of components from the layers beneath it.

The layered mPower framework provides a flexible infrastructure for process coordination. Moreover, the generic workflow layer pre-defines workflow templates that avoid redundant development of similar business processes.

2.4 RETSINA

RETSINA [10] is an implemented open MAS infrastructure that supports communities of heterogeneous agents and has been applied in many fields, such as financial portfolio management and logistic planning. The MAS infrastructure is defined as the set of services, conventions, and knowledge that support complex social interactions among participating agents. The abstract infrastructure consists of 9 layers that provide different services to a MAS:

– Operating Environment: A layer on which a MAS relies, including physical computers, operating system, networks;

– Communication Infrastructure: This layer transfers messages between the agents as well as between the agents and the MAS infrastructure;
– ACL (Agent Communication Language) Infrastructure: This layer provides the specification of a language that can be spoken and understood by all the agents in the system community;
– Multiagent Management Services: This layer provides additional system operation services, such as logging facilities, management tools, installation services and launching services;
– Performance Measurement: This layer monitors the performance of the heterogeneous agents which differ in their ability, efficiency, reliability etc;
– Security: Since agents in an open MAS, which have been designed by different development groups, can join and leave the society dynamically, it is necessary to have a security layer that ensures agents do not misbehave;
– Mapping Names to Agent Locations: This layer maps the agent name dynamically to the agent location that provides the basis for agent mobility;
– Mapping Capabilities to Agents: This layer maps capabilities to agents by means of Middle Agents, which match agents' requests and agents' advertisements;
– Interoperation: This layer provides real-time interoperation between different MASs that may have their own architecture-specific features.

RETSINA is implemented on the basis of this infrastructure. It implements distributed infrastructural services that facilitate the relations between the agents. It is based on two types of communication channels:

1. Direct message transfer, which provides peer to peer communication between the agents;
2. Multicast based discovery process, which connects agents to the infrastructure components.

It provides an ontology based on diverse domain-specific taxonomies of concepts, as well as a protocol engine and a protocol language. RETSINA uses middle agents called Matchmakers to allow agents to search what services are available in the MAS and who provides the services. Each Matchmaker records a mapping between agents in the system and the services that they provide. One important characteristic of the RETSINA Matchmakers is that they do not stay in the middle of the interaction between the providers and the requesters. Because of this, the system becomes more robust. The RETSINA-OAA InterOperator in the RETSINA MAS infrastructure provides a bridge between the RETSINA system and the OAA (Open Agent Architecture) system. It allows any agent in one system to access any service or information provided by agents of another system.

Technically speaking, Matchmakers in RETSINA act successfully as middlemen that match the agents and the services each agent provides, which show an important basic functionality of a mediator that facilitates business process coordination.

2.5 Insufficiencies of Current Techniques

Although the techniques described in this section can solve process coordination problems to some degree, these systems are quite passive. The systems just work as middle machines that receive tasks and then execute them.

Each technique has its own characteristics:

- TuSCoN focuses on objective coordination, which is coordination outside the agents. The workflow rules are encapsulated as coordination laws and then superimposed on the agents.
- JBees focuses on the simulation, analysis, and monitoring of the process execution in order to identify potential inconsistencies and improve process performance.
- mPower considers conversations among participating process actor roles as the core of the framework.
- In RETSINA, coordination structure emerges from the relations between the agents that are facilitated by the distributed infrastructural services. RETSINA also uses Matchmakers to match between agents and the services they provide.

All these infrastructures focus on the step by step execution of business processes. None of them cares whether (at a certain step) there is another possible solution that may give more benefits. The middleware doesn't have its own goals. However, industrial solutions often require reactive and proactive mediator agents that have their own goals (such as find best partner for job, find cheapest component for solution, etc), and are able to reason about the goals and motivations of the different parties. This means that besides the technical aspect, both the economic and the social aspects of the system must be considered in a way such that business objectives, organizational culture and norms can be taken into account in the proposed interactions.

3 Requirements of the ABC Method

In section 2, we gave an overview of the current techniques. Although those techniques provide relatively good infrastructures for process coordination, they do not consider the economic and social aspects of the solutions.

3.1 Economic Requirements for Mediation

Since one main objective that organizations coordinate their business processes is to reduce costs, solutions that don't consider the economic aspects are not sufficient.

Integration is key to e-business, therefore, extensive research has been done towards theories on process integration. Johannesson and Perjons present design principles for process modeling in Enterprise Application Integration (EAI) [11]. They argue for a Process Broker architecture and propose a high-level modeling language BML (Business Model Language) for application integration. Gordijn designs the e3-value model [12] that introduces and combines several aspects that have to be taken into account when combining applications. Some researchers have also argued for a separation of coordination aspects from the application functionality, where the coordination aspects are described by contract. However, all of these are ingredients for BPI, such that the problem of how the integration should be performed hasn't yet been indicated.

In [13], it is mentioned that by reducing the costs of coordination, information technology will lead to an overall shift toward proportionately more use of markets to coordinate economic activity. With the increasing use of IT, companies may deal with customers directly (e.g. via internet), which removes the intermediaries, such as distributor, wholesaler, broker, or agent. This is also called cutting out the middleman (disintermediation). However, there are also possibilities for reintermediation, which is the reintroduction of an intermediary between consumers and producers after disintermediation has occurred. The new roles for intermediaries include trust provision, aggregation, one-stop shopping, information exchange facilitator, and information filtering brokers. Current interorganizational systems do not often succeed because the added-value for the stakeholders is hard to quantify and guarantee. The situation can be considered as a prisoner's dilemma: collaboration on the basis of system integration is profitable for all parties, but only if all parties do cooperate. Therefore, there is a need to analyze the problem and come up with workable business models, particularly for the situation where the intermediary party is independent.

Companies try to enter the market and want to gain as much benefits as possible. Consumers that use mediators can benefit by receiving more information and services. When mediators hold a big amount of customers, companies would like to join mediators as well because of the huge market mediators have. However, mediators won't work for nothing. As an independent middle party, mediators also want to benefit from the services they provide. Thus, contracts or norms are needed to enforce companies and consumers remaining in the participation.

3.2 Social Requirements for Mediation

Mediation across organizations has to deal with self-interested agents in open environment, where heterogeneous participants that have either cooperative or competitive goals can join and leave dynamically. In order to ensure the cooperative outcomes for tasks that are not doable individually, trust among participants is an important factor. One possible solution that can provide a certain level of trust is to regulate the environment to enforce appropriate types of participant's behavior.

Norms play an important role in open multi-agent systems (MAS) and allow for the development of trust and reputation mechanisms [14]. Having norms is necessary but not yet sufficient by itself, because agents will not voluntarily submit themselves to associated penalties when deviation occurs. Therefore, mechanisms such as Electronic Institution (EI) [15] are needed to enforce norm compliance. An Electronic Institution (EI) provides the necessary level of trustable environment by enforcing norms and providing specific services.

When multiple solutions are available, participants' preferences should also be taken into account in order to provide more satisfied services.

4 Scenario for the ABC Framework

We propose an Agent-mediated Business Coordination (ABC) model that focuses on the analysis of requirements of mediators, and on the development of workable mediation models that consider technical, economic and social aspects.

In this section, we use a travel example (see Figure 1) to illustrate scenario for the ABC framework. The situation is that people want to travel and they need to arrange air tickets, hotels, excursions, or car rental, etc. The objective of the arrangement is to meet the criteria mentioned by the consumer, e.g. low price, hotel should locate in the center, the trip should be on a certain date, etc. There are three possible ways to arrange the travel:

– Arrange the travel by contacting companies that provide the services directly, which means arranging everything without any intermediaries (a travel agency, a ticket agency, etc).
– Arrange the travel by consulting some agencies.
– Arrange the travel using a mediator.

When arranging the travel by contacting the companies directly, one can choose the exact air company and hotel he or she prefers. However, the person needs to arrange each step separately and finally combine them together, which can take a long time and the process is quite complicated.

When arranging the travel by consulting agencies, there are many possibilities as well. One can go to a ticket agency that provide tickets from different companies with different prices, or go to a travel agency that provides information of both tickets and hotels. Different agencies may give different promotions, but mostly they cannot provide information that cover all companies.

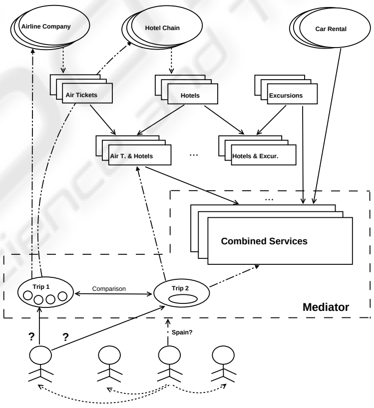

Fig. 1. Travel Example.

Another important point is that the consumer wants to travel as cheap as possible when the qualities of services meet the consumer's requirements. As a consequence, the consumer may want to compare the prices and the qualities of services they can get from different arranging methods and make decision based on the comparison. It is obvious that it is quite hard to compare all the information without any help.

We propose a way that arranges the travel using a mediator. Some possible functionalities of a mediator in this example are as follows:

– As a normal agency, the mediator is capable to aggregate information of different companies and provide it to consumers. One important aspect of mediator is to group as much information as possible.
– The mediator should be able to compare the results obtained from the two previously mentioned methods using the criteria proposed by consumers.
– Mediator should also take into account the norms of different agents. For example, maybe advertisement cannot be sent to consumers if they didn't request it, or it is not allowed to pass personal information to others.
– Another issue is to take into account the preferences of agents: likes to travel in groups or not, prefers single room or a room with cheaper price, etc.
– The mediator can be proactive, i.e. when it has some information of a potential promotion, the mediator can broadcast the information and gather people. For example, a hotel informed the mediator that there will be 10 rooms free in the coming week, if the mediator can gather 10 people, the hotel can ask for half of the price (the price of 5 rooms). In this case, the mediator can broadcast the promotion and try to gather enough people.
– Proactive can be bi-directional, i.e. when a group of people want to book hotel rooms, they can ask the mediator whether they can get any discount. The mediator can then negotiate with hotels on whether they would like to give discount to the group of people.
– Further more, when a customer wants to travel to a certain place (e.g. Spain), he/she may consult the mediator whether anybody else wants to go to Spain. The mediator can check it and reply to the customer. From the information provided by the mediator, the customer can gather people who want to go to Spain and ask the mediator for discount of the trip.

We will use this scenario to help us to generate ideas about when to use a mediator. It can also help us to build and test our future business models, and help us to identify the value of a mediator.

5 Different Types of Mediators

In different situations, it is necessary to have mediators with different power and functionalities that act in different ways. Considering the travel example in Figure 1, mediators that deal with hotels and those that deal with air tickets should act differently.

Mediators that deal with hotels can be quite powerful in a way that they can set contracts with the hotels to force them selling rooms via the mediators. They can also be proactive to give suggestions to the customers or the hotels. From the economic point of view, setting up exclusive contract can ensure the profits of the mediators. This is obvious since without a contract, hotels and customers can set up their own business relationship after getting information from the mediators. However, both advantages and disadvantages exist for hotels that decide to set up a contract with the mediator. The result depends on how many customers the mediator holds. If the mediator has a huge amount of customers, the hotels can benefit from such a big market and will enjoy the contract. On the other hand, if the mediator has just a few customers, the contract would force the hotels remain in the undesired situation.

Mediators that deal with airline companies cannot perform in the same way as dealing with hotels, because airline companies are quite powerful and have dominant positions in the market. What mediators can do is to gather as much flights (promotion) information as possible and provide it to the customers. Mediators can also be proactive in this case, but only to the customers. The reason is that mediators can not tell an airline company to arrange a flight on a specific date and time. However, they can query customers when promotions are available. For example, if a mediator knows Jane goes to Spain several times each month, and currently there are discount tickets to Spain, it can inform Jane and ask her whether she wants to book a ticket. Economically, the more valuable information a mediator can gather, the more profit it can gain, i.e. if a mediator has a dominant position among systems that provide flight information, customers would have the preference to go to the mediator.

The necessity of different types of mediators indicates that a simple model of mediator is not sufficient. What we need is to identify different business models of mediators and the factors that determine the applicability of these models to different situations.

There are several reasons of using agents to model mediators. In real-world practice of mediation, some of the most common aspects of a mediator codes of conduct may inspire the design of a mediation system [16], which are as follows:

- Mediators must adopt a neutral stance towards all parties to the mediation;
- Mediators must conduct the mediation in an impartial manner;
- Within the bounds of the legal framework, any information gained by the mediators should be treated as confidential;
- Mediators should not offer legal advice, but direct participants to appropriate sources that are useful for the participants' decision-making;
- Mediators should seek to maintain their skills by engaging in ongoing training in the mediation process;
- Mediators should practice only in the fields in which they have expertise gained by their own experience.

Intelligent agents can be considered as the most appropriate paradigm in order to handle these aspects, because of their intrinsic capabilities, such as autonomy, reactivity, pro-activeness, and sociality. Agents have goals, and can adapt to changing environment by continuous learning and updating their knowledge.

Agents can be endowed with both economic and social goals. As a consequence, interaction rules and norms should be incorporated into existing technical infrastructure in order to support the agents' coordination. Since mediators are domain-specific, one (or more) mediator(s) may be preferred to the others under certain circumstances. Therefore, coordination is needed when multiple agents (mediators) coexist.

6 Conclusions

Mediators should be able to reason about motivations and goals of the different parties, such that they can adapt to their (changing) needs. By understanding goals and motivations, the mediator is able to provide meaningful alternatives for the requests of different parties.

In this paper, we presented the initial work towards a mediation architecture for business process integration across organizations. We discussed existing solutions and their shortcomings and identified economic and social requirements for a more comprehensive solution. However, our work has just started. In the following we indicate the direction of our current research and describe the research methods we propose to use in this project.

Different research methods will be used in the research. Firstly, to come up with an interdisciplinary theory of mediation in the context of BPI, we will do literature survey on coordination mechanisms, economy and agent society. In order to be able to support the choice of an appropriate type of mediator for a particular situation, we will develop a mediator decision process. Different types of mediators and their adaptability will be tested in an agent society test bed, which will also be developed.

In section 4, we proposed the idea of using a mediator for the travel problem. We will generalize the problem to cover general business process coordination cases. First of all, it is necessary to analyze the business coordination context in which mediation is required or can give outstanding performance. According to the analyzed results, mediation models that have different functionalities can be designed. Generalization can be achieved by combining the diverse, but not conflict functionalities of different mediation models.

The final objective of our research is to develop a unified framework of mediation that takes into account technical, economic, and social aspects. In order to achieve this, we need to classify mediators according to their characteristics and functionalities. Based on the classifications, different models will be built up to represent different types of meditators. Furthermore, these models should be able to adapt to the continuous change in the environment.

Acknowledgements

This research is supported by NWO, under the project “Agent-mediated Business Coordination”.

References

1. Dayal, U., Hsu, M., Ladin, R.: Business process coordination: State of the art, trends, and open issues. In: The VLDB Journal. (2001) 3–13
2. Wiederhold, G.: Mediators in the architecture of future information systems. In Huhns, M.N., Singh, M.P., eds.: Readings in Agents. Morgan Kaufmann, San Francisco, CA, USA (1997) 185–196

3. Dignum, V.: A model for Organizational Interaction:Based on Agents, Founded in Logic. PhD thesis, Utrecht University (2004)
4. Ricci, A., Denti, E., Omicini, A.: Agent coordination infrastructures for virtual enterprises and workflow management. **Volume 2182/2001** (2001) 235–246
5. Morini, S., Ricci, A., Viroli, M.: Integrating a mas coordination infrastructure with web services. (2004)
6. Ehrler, L., Fleeurke, M., Purvis, M.: Agent-based workflow management systems (wfmss). Information Systems and E-Business Management **Volume 4** (2005) 5–23
7. Purvis, M., Cranefield, S., Nowostawski, M., Carter, D.: (Opal: A multi-level infrastructure for agent-oriented software development)
8. Nowostawski, M.: Jfern - java-based petri net framework. (2003)
9. Shepherdson, J.W., Lee, H., Mihailescu, P.: mpower: a component-based development framework for multi-agent systems to support business processes. British Telecom technology journal **Volume 21** (2003) 92–103
10. Sycara, K., Paolucci, M., van Velsen, M., Giampapa, J.: The RETSINA MAS infrastructure. Technical Report CMU-RI-TR-01-05, Robotics Institute, Carnegie Mellon (2001)
11. Johannesson, P., Perjons, E.: Design principles for process modelling in enterprise application integration. Information Systems **26** (2001) 165–184
12. Gordijn, J.: Value based requirements engineering: Exploring innovative e-commerce ideas. PhD thesis, VU Amsterdam (2002)
13. Malone, T.W., Yates, J., Benjamin, R.I.: Electronic markets and electronic hierarchies. Commun. ACM **30** (1987) 484–497
14. Cardoso, H.L., Rocha, A.P., Oliveira, E.: Virtual organization support through electronic institutions and normative multi-agent systems. (2006)
15. Dignum, V., Dignum, F.: Modelling agent societies: Co-ordination frameworks and institutions. In: Portuguese Conference on Artificial Intelligence. (2001) 191–204
16. Boulle, L., Nesic, M.: Mediation: Principles, Process, Practice. LexisNexis UK (2001)