WAT 2010

PROCEEDINGS

III Workshop on Agreement Technologies
(WAT 2010 at Iberamia 2010)

November 1st, 2010

Universidad Nacional del Sur
Bahía Blanca, Argentina

Marc Esteva, Alberto Fernández, Adriana Giret and Vicente Julián (Eds.)

AGREEMENT TECHNOLOGIES

www.agreement-technologies.org

Preface

Agreement is one of the crucial social concepts that helps human agents to cope with their social environment and is present in all human interactions. In fact, without agreement there is no cooperation and ultimately social systems cannot emerge. Agreement is necessary in our everyday life.

Until recently, the concept of agreement was a domain of study mainly for philosophers, sociologists and was only applicable to human societies. However, this situation has changed in the recent years, especially with the spectacular emergence of information society technologies. Computer science has moved from the paradigm of an isolated machine to the paradigm of a network of systems and of distributed computing. Likewise, artificial intelligence is quickly moving from the paradigm of an isolated and non-situated intelligence to the paradigm of situated, social and collective intelligence. Hence, the concept of agreement has become key for a robust understanding and an efficient implementation of artificial social systems.

In this context, Agreement Technologies is a new approach of Distributed Artificial Intelligence for constructing large-scale open distributed computer systems. This workshop on Agreement Technologies is specifically addressed for any work that aims at developing models, frameworks, methods and algorithms for constructing such systems. In other words, this workshop focuses on approaches and solutions for the needs of next generation computing systems where autonomy, interaction and mobility will be the key issues. Most importantly, it concentrates on techniques that enable software components to reach agreements on the mutual performance of services.

Agreement Technologies integrates many research efforts from different fields of Artificial Intelligence. Hence, this workshop is specifically tailored to research works related to this new approach. Finally, the editors would like to thank all the people that bring about WAT and Iberamia 2010. First of all, thanks to the authors for ensuring the richness of the workshop and the members of the program committee for their professionalism and dedication. Furthermore, we owe particular gratitude to the Iberamia organizing committee.

Marc Esteva, Alberto Fernández, Adriana Giret and Vicente Julián
WAT2010 Organizing Committee

ORGANIZING COMMITTEE

Marc Esteva (Artificial Intelligence Research Institute (IIIA), Spain)
Adriana Giret (Universidad Politécnica de Valencia, Spain)
Alberto Fernández (Universidad Rey Juan Carlos, Spain)
Vicente Julian (Universidad Politécnica de Valencia, Spain)

STEERING COMMITTEE

Carles Sierra (IIIA-CSIC, Spain)
Sascha Ossowski (Universidad Rey Juan Carlos, Spain)
Vicente Botti (Universidad Politecnica de Valencia, Spain)

PROGRAM COMMITTEE

Holger Billhardt (Universidad Rey Juan Carlos, Spain)
Luis Botelho (ISCTE, Lisbon, Portugal)
Vicente Botti (Universidad Politecnica de Valencia, Spain)
Christiano Castelfranchi (Institute of Cognitive Sciences and Technologies, Italy)
Jaelson Castro (Universidade Federal de Pernambuco, Brazil)
Carlos Chesnevar (Universidad Nacional del Sur, Argentina)
John Debenham (University of Technology, Australia)
Frank Dignum (Universiteit Utrecht, The Netherlands)
Marc Esteva (IIIA-CSIC, Spain)
Alberto Fernandez (Universidad Rey Juan Carlos, Spain)
Adriana Giret (Universidad Politecnica de Valencia, Spain)
Carlos Angel Iglesias (Universidad Politecnica de Madrid, Spain)
Vicente Julian (Universidad Politecnica de Valencia, Spain)
Maite Lopez-Sanchez (Universitat de Barcelona, Spain)
Michael Luck (King's College London, UK)
Peter McBurney (University of Liverpool, UK)
Pablo Noriega (IIIA-CSIC, Spain)
Eugenio de Oliveira (Universidade do Porto, Portugal)
Eva Onaindia (Universidad Politecnica de Valencia, Spain)
Sascha Ossowski (Universidad Rey Juan Carlos, Spain)
David Pearce (Universidad Rey Juan Carlos, Spain)
Enric Plaza (IIIA-CSIC, Spain)
Maria Jose Ramirez (Universidad Politecnica de Valencia, Spain)
Juan Antonio Rodriguez-Aguilar (IIIA-CSIC, Spain)
Juan Manuel Serrano (Universidad Rey Juan Carlos, Spain)
Jaime Sichman (EP/USP Sao Paulo)
Viviane Torres da Silva (Universidade Federal Flumiense, Brazil)
Wamberto Vasconcelos (University of Aberdeen, UK)
Michael Wooldridge (University of Liverpool, UK)

Index

An Electronic Institution for Simulating Water-Right Markets

Vicente Botti[1], Antonio Garrido[1], Juan A. Gimeno[1], Adriana Giret[1], Francesc Igual[1], Pablo Noriega[2]

[1] DSIC, Department of Information Systems and Computation, Universitat Politècnica de Valencia,
[2] IIIA, Artificial Intelligence Research Institute, CSIC, Spanish Scientific Research Council,
{vbotti,agarridot,jgimeno,agiret,figual}@dsic.upv.es, pablo@iiia.csic.es

Abstract. In countries like Spain, and particularly in its Mediterranean coast, there is a high degree of public awareness of the main consequences of the scarcity of water and the need of fostering efficient use of water resources. Two new mechanisms for water management already under way are: a heated debate on the need and feasibility of transferring water from one basin to another, and, directly related to this proposal, the regulation of *water banks*[1]. This paper is about *mWater*, an agent-based electronic market of water rights. Our focus is on demand and, in particular, on the type of regulatory and market mechanisms that foster an efficient use of water while preventing conflicts. In this work we present the regulated environment which is implemented as an Electronic Institution for simulating water-right markets in order to evaluate the impacts of different regulations on the market behaviour.

1 Introduction

Water scarcity is becoming a major concern in most countries, not only because it threatens the economic viability of current agricultural practices, but because it is likely to alter an already precarious balance among its many types of use: human consumption, industrial use, energy production, recreation, etc. Underneath this emergent situation, the crude reality of conflict over water rights of use and the need of accurate assessment of water needs and use become more salient than ever.

It has been sufficiently argued that more efficient uses of water may be achieved within an institutional framework where water rights may be exchanged

[1] The 2001 Water Law of the National Hidrological Plan (NHP) —'Real Decreto Legislativo 1/2001, BOE 176' (see www.boe.es/boe/dias/2001/07/24/pdfs/A26791-26817.pdf, in Spanish)— and its amendment in 2005 regulates the power of right-holders to engage in voluntary water transfers, and of basin authorities to setup water markets, banks, and trading centers for the exchange of water rights in cases of drought or other severe scarcity problems.

more freely, not only under exceptional conditions but on a day to day basis [3, 9, 12]. It has been claimed that if farmers cannot sell their extra water allotment, they have no incentive to use the allotment efficiently and it may become wasteful [5]. Moreover, a straightforward extension to other types of stakeholders would promote trading for non-irrigation uses, such as industrial uses, aquiculture, leisure or navigation, thus improving market conditions and hence efficiency of water use [3]. We propose to implement such a market with a regulated open multi-agent system, *mWater*, whose main features we discuss in this paper. Our focus is on demand and, in particular, on the type of regulatory and market mechanisms that foster an efficient use of water while preventing conflicts.

Considerable effort has been invested in the development of sophisticated basin simulation models and in improvement and innovation of water use practices. Literature abounds in examples of decision support systems for water management [8], sustainable planning of water volumes [2, 6], or the use of shared visions for negotiation and conflict resolution [7]. We explore an alternative approach in which individual and collective agents are an essential component because their behavior (and effects) may be influenced by policy-making. There are few projects along this line, but one may point to the NEGOWAT project (http://www.negowat.org/ingles/inicio/Inicio.htm), whose goal is to help negotiations between stakeholders in peri-urban catchment areas when water conflicts arise. Closer to our own approach, the recent effort is project MAELIA (http://www.iaai-maelia.eu), which involves simulation of socio-environmental impact of norms for water and other renewable natural resources and the environment.

We are interested in the institutional framework that simulates the "rules of the game" that may allow one to study the role that regulation, social environment, coordination, conflict resolution mechanisms, reputation or trust play in the decisions participating agents make and their aggregate results. Ideally, the institutional framework should add flexibility to current water use practices without increasing the number or complexity of disputes. To this end, we have designed *mWater* as an agent-based system that simulates an electronic market of water rights in which we use agreement technologies such as: normative reasoning, negotiation rules, argumentation, trust, collective decision-making, social conventions, sanctioning mechanisms, as well as organizational and institutional environments preferences, among others.

The main goal of this paper is to describe the *mWater* regulated environment that fosters efficient use of water resources by means of water-right transfer agreements (Section 2). We propose *mWater* as our particular setting; nonetheless, it can be useful for other markets not related to water problems. In order to be more concise, Section 3 devises the simulation environment for this electronic market. Section 4 provides a particular case study on regulatory aspects in *mWater*, which again can be extrapolated to other domains. Finally, we conclude the paper with some remarks in Section 5.

2 An institutional framework for *mWater*

The *mWater* framework is rooted on traditional practices and regulations for the use and transfer of water rights that are either currently established by the Spanish National Hydrological Plan or are to be part of the forthcoming Basin Hydrological Plans. However, it is somewhat idealized in order to provide a richer sandbox for agreement technologies and a more malleable platform for demand and water use modeling and simulation in an hydrographic basin. The core component of *mWater* is an agent-based virtual market for water usage rights that intends to grasp the components of an electronic market where water rights are traded with flexibility and under different price-fixing mechanisms. In addition to trading proper, *mWater* also includes those activities that follow trading; namely, the agreement on a contract, the use and misuse of rights and the grievances and corrective actions taken therein. These ancillary activities are particularly prone to conflict albeit regulated through legal and social norms, and therefore a crucial objective in policy-making and a natural environment for agreement technologies.

For the construction of *mWater* we have followed the IIIA *Electronic Institution* (EI) conceptual model [1], whereas for the actual specification and implementation of *mWater* we use the EIDE platform[2].

Procedural conventions in the *mWater* institution are specified through a nested performative structure (Fig. 1) with multiple processes. As seen in the figure, there are several roles: (i) guests, i.e. users before really entering the market; (ii) water users, i.e. the guests that have valid water rights; (iii) buyer/seller, thus representing the particular role the water user currently joins for the market; (iv) third parties, i.e. those water users that are direct or indirectly affected by a water transfer —usually conflicting parties; and (v) market facilitator and basin authority, thus representing the governing roles of the market. The top structure, *mWaterPS*, describes the overall market environment and includes other performative structures. *TradingHall* (Fig. 2) provides updated information about the market, and at the same time users and trading staff can initiate most trading and ancillary operations here. Finally, *TradingTables* establishes the trading procedures. An outline of their constitutive processes (performative structures and scene protocols) follows.

Top structure, *mWaterPS*. *Entitlement.* Only bona fide right-holders may trade water rights in the market and there are only two ways of becoming the owner of a right. Firstly when an existing right is legally acquired from its previous owner outside of *mWater* (through inheritance or pecuniary compensation for example). Secondly when a new right is created by the *mWater* authorities

[2] EIDE is a development environment for Electronic Institutions, built at the IIIA, http://e-institutor.iiia.csic.es/eide/pub. It is composed of a set of software tools that support all the stages of an Electronic Institution (EI) engineering, namely: 1) ISLANDER, a tool for EI specification; 2) aBUILDER, a tool to support the automatic generation of agent (code) skeletons from ISLANDER specifications of an EI; 3) the AMELI middleware that handles the enactment of the institution; and 4) SIMDEI, a testing and monitoring tool.

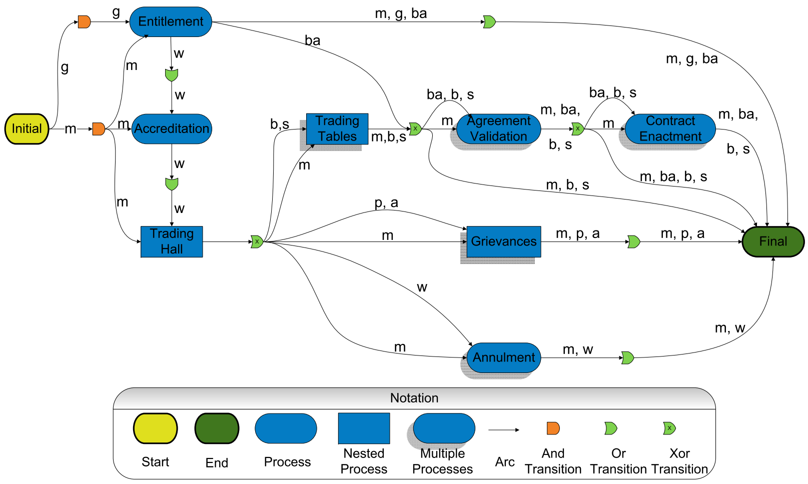


Fig. 1. *mWater* performative structure. Participating Roles: *g* - Guest, *w* - Water user, *b* - Buyer, *s* - Seller, *p* - Third Party, *m* - Market Facilitator, *ba* - Basin Authority

and an eligible holder claims it and gets it granted. *Entitlement* scene gives access to the market to new right holders who prove they are entitled to trade. It is also used to bootstrap the market.

Accreditation. This scene allows legally entitled right-holders to enter the market and trade by registering their rights and individual data for management and enforcement purposes. Staff have to validate admission conventions and right-holder variables are given default variables. When a right suspension is overridden or an agreement is void, rightful owners need to register again.

Agreement Validation and *Contract Enactment.* Once an agreement on transferring a water right has been reached, it is managed according to the market conventions. *mWater* staff check whether or not the agreement satisfies formal conditions and the hydrological plan normative conventions (*Agreement Validation* scene of Fig. 1). If the agreement complies with these, a transfer contract is agreed upon and signed by the parties involved in the *Contract Enactment* scene, and then the agreement becomes active.

Annulment. This scene in the *mWater* performative structure deals with anomalies that deserve a temporary or permanent withdrawal of rights.

***TradingHall* performative structure.** Intuitively, in this complex performative structure, see Fig. 2, right-holders become aware of the market activity (*Open Trades* and *Ongoing Agreements* scenes), and initiate concurrent activities: get invitations to trade and/or initiate trading processes (*Recruiting* scene), initiate grievance procedures as described below in Fig. 3 (*Ongoing Agreements* scene), and get informed about anomalous situations, for example severe drought situations, (*Critical Situations* scene). Actual trading starts inside the *TradingHall* scene. On the one hand, updated information about existing trade-

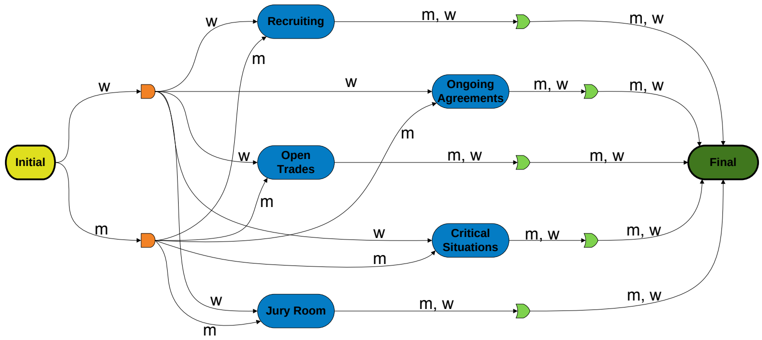


Fig. 2. *TradingHall* performative structure

able rights, as well as ongoing deals, active contracts and grievances is made available here to all participants. On the other, as shown in Fig. 2, users and trading staff can initiate most trading and ancillary operations here (from the *Recruiting* scene): open, request trading parties and enter a trading table; query about different agreements and initiate a grievance procedure from the *Ongoing Agreements* scene or, in the same scene, get informed about a dispute in which the water user is affected. Members of the Jury may also be required to mediate in a dispute at the *Jury Room* scene. Technically speaking, all these scenes are "stay-and-go" scenes: while the users are inside the market, they have to *stay* permanently in these scenes but they may also *go* (as *alteroids*, clone-like instantiations of the same agent that allow the agent to be active simultaneously in different scenes) to other trading table scenes and contract enactment scenes where they are involved: these scenes where user alteroids become involved are created (as a new *instance* of the corresponding performative structures) when a staff agent creates one at the request of a user, of an authority, or because of a pre-established convention (like weekly auctions).

***TradingTable* performative structure.** In our *mWater* performative structure (recall Fig. 1), a market facilitator can open a new trading table whenever a new auction period starts (i.e. automatically) or whenever a right-holder requests to trade a right (i.e. on demand). In such a case, a right-holder chooses a negotiation protocol from a set of available ones (e.g., face to face negotiation, closed bids, standard double auction exchange or any others that are agreed upon). Consequently, in order to accommodate different trading mechanisms, we assemble the *TradingTable* performative structure as a list of different scenes, each corresponding to a valid trading mechanism or negotiation protocol. Each instance of a *Trading Table* scene is managed by a *Negotiation Table Manager*, *tm*, who knows the structure, specific data and management protocol of the given negotiation protocol. Among other negotiation mechanisms, we have included face-to-face, Dutch auction, English auction, standard double auction and blind double auction with mediator negotiation. Moreover, new negotia-

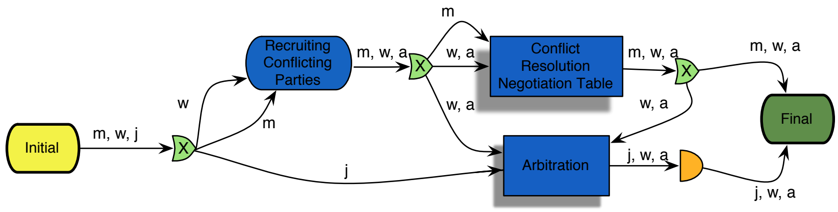


Fig. 3. Grievances performative structure

tion protocols may be easily added providing that the new protocol definition complies with the generic structure.

Every generic negotiation table is defined as a three-scene performative structure. The first scene is *Registration*, in which the *tm* applies a filtering process to assure that only valid water users can enter a given trading table (recall situations when a private trading table is being executed or only a sub-group of water users that fulfill a set of constraints may participate in the table). The specific filtering process will depend on the given negotiation protocol and possibly on domain specific features. The second scene is the negotiation protocol itself, in which the set of steps of the given protocol are specified. Finally, in the last scene, *Validation*, a set of closing activities are executed, for example registering the final deals, stating the following steps for the agreement settlement, verifying that the party that leaves the table satisfies the exit norms of the trading table, etc.

Grievances. Once an agreement is active, it may be executed by the new right-holder and, consequently, other right-holders and some external stakeholders may initiate a grievance procedure that may overturn or modify the transfer agreement. Even if there are no grievances that modify a contract, parties might not fulfill the contract properly and there might be some contract reparation actions. If things proceed smoothly, the right subsists until maturity.

Fig. 3 shows the different scenes of the complex Grievances performative structure. In this structure any conflict can be solved by means of two alternative processes (these processes are similar to those used in Alternative Dispute Resolutions and Online Dispute Resolutions [10, 11]). On the one hand, conflict resolution can be solved by means of negotiation tables (Conflict Resolution Negotiation Table performative structure). In this mechanism a negotiation table is created on demand whenever any water user wants to solve a conflict with other/s water user/s, negotiating with them with or without mediator. Such a negotiation table can use a different negotiation protocol, such as face to face, standard double auction, etc. On the other hand, arbitration mechanisms for conflict resolution can also be employed (Arbitration performative structure). In this last mechanism, a jury solves the conflict sanctioning the offenses.

There are three steps in the arbitration process. In the first one, the grievance is stated by the plaintive water user. In the second step, the different conflicting parties present their allegations to the jury. Finally, in the last step, the jury, after hearing the dispute, passes a sentence on the conflict. The difference among the two mechanisms for conflict resolution is that the arbitration process is binding, meanwhile the negotiation is not. In this way, if any of the conflicting parties is not satisfied with the negotiation results he/she can activate an arbitration process in order to solve the conflict.

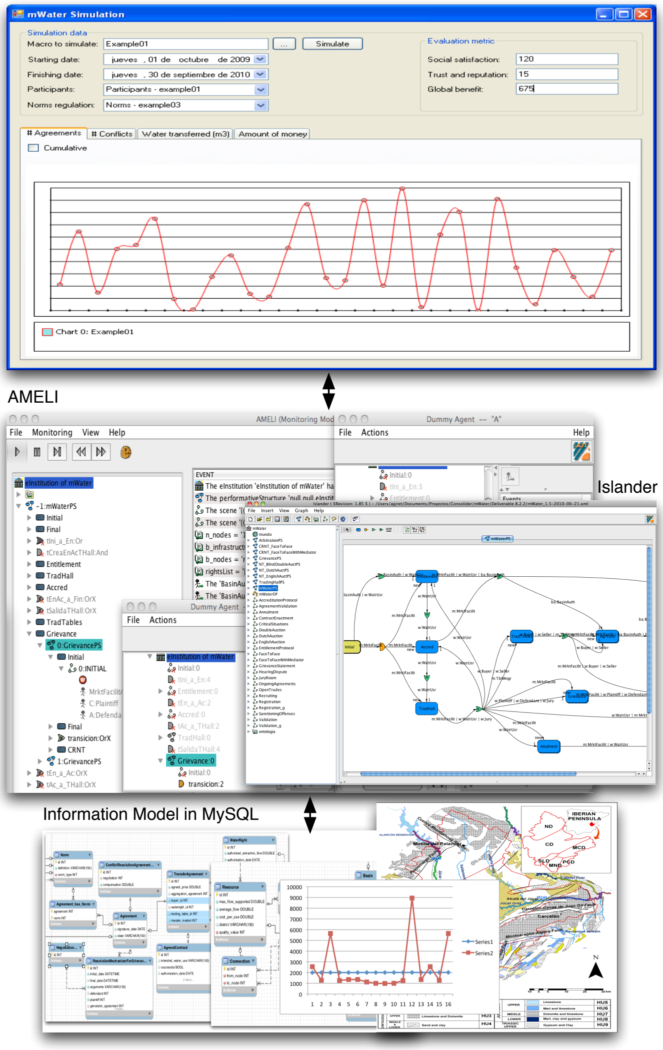


Fig. 4. *mWater* Simulation Environment

3 The *mWater* simulation environment

Fig. 4 depicts the overall structure of the *mWater* simulation environment. The interface of the simulation tool is simple and intuitive, in which the user can configure a given simulation with the following data: the starting and finishing date for the period to be simulated, the water users that will participate in the market (different groups/type of water users can lead to different results; consider for example a group in which some water users do not trust on other members of the group, this situation will probably result in a low number of agreements and a high number of conflicts), the regulation that will be applied in the simulation (in next section a case study example is presented with two different norm regulations applied to the same water user population and simulation period). The interface also provides graphical data that reflects how the market reacted to the input data in terms of the number of transfer agreements signed in the market (historical data including information about real or simulated users), volume of water transferred, number of conflicts generated, etc. Apart from these straightforward functions, other quantitative results are shown. These results are from a group of "social" functions in order to asses values such as the trust and reputation levels of the market, or degree of water user satisfaction, among others.

The central element in the simulation tool is the EI described in last section, that is specified in ISLANDER and executed in the AMELI runtime platform (recall Fig. 4). In order to start a simulation of the market, *mWater* feeds from an Information Model (implemented in MySQL) in which historical data from a given basin are registered. In this way, policy makers can simulate the market with real data from sever drought periods, rain spell, etc. depending on the starting and ending dates defined for the simulation. Moreover, the Information Model registers all the changes in the market in order to provide statistical data to the policy makers about the market behaviour for the simulated period, the water users that participated in the market, and the regulations selected for the particular simulation.

Note that we have mainly considered *mWater* as a simulation environment, but actually we are also interested in it as an open environment to human users for conducting social and participatory simulations. In such situations, human subjects take part in the simulation to see the effects of their interaction with virtual agents, applicable norms and their adaptation. This is part of our current work.

4 A case study simulation in *mWater*

The emphasis on regulatory aspects in *mWater* is motivated by the fact that the main objective policy makers have in mind is to achieve an adequate behavior of users. And regulation is the main tool that policy makers have to modify behavior. However, in practice, users are prone to achieve "order without law" [4], or at least to keep on adapting to regulations in order to preserve their successful practices while policy makers keep on adapting regulations to guide users in a constantly changing environmental and political media. Thus, our *mWater* demonstrator provides the foundations for the study of that interplay. In order to show the way *mWater* can be used as a simulation tool for testing how regulations and norms can modify the users' behavior we show a simplified case study that mainly focuses on the interplay of norms and relaxes the other aspects in the market. In this case study, we test the user behavior when a single norm is modified. This scenario is related with the registration of transfer agreements.

In *mWater* we have three different types of regulations: (i) government laws, issued by the Spanish Ministry of Environment (stated in the National Hydrological Plan); (ii) basin or local norms, defined and regimented by the basin authorities; and (iii) social norms, stated by the members of a given user assembly and/or organization. The norms applied in this case study are currently defined in the NHP. However, policy makers have observed that only few water rights transfer agreements are registered in the basin while a lot of non-registered transfers are taking place by means of private commitments. This situation appears due to the interplay of the following norms:

Government law - (N0): A water-user can use a given volume of water from a given extraction point, if and only if he/she owns the specific water-right or has a transfer agreement that endows him/her.

Government law - (N1): Every water-right transfer agreement must be registered within the fifteen days after its signing and wait for the Basin Authorities' approval in order to be executed.

Local norm - (N2): The registration process of a water-right transfer agreement is started voluntarily by the agreement signing parties.

Social norm - (N3): Whenever a conflict appears, a water user can start a grievance procedure in order to solve it.

In order to include norm *N1* in the current EI implementation of *mWater* we have designed the *Agreement Validation* scene (see Fig. 1) as a successor scene for any Trading Table. When any water user enters this scene, the Market Facilitator verifies the constraint of fifteen days from the agreement statement process related to norm *N1*. If this constraint is satisfied the water-right transfer agreement is forwarded to the Basin Authority, who activates a Normative Reasoning process in order to approve, or not, the agreement, based on the basin normative regulation. If the agreement gets approved it is published in the Trading Hall in order for every water user of the basin to be informed of the transfer agreement.

On the other hand, norm *N2* is automatically included in the *mWater* institution due to the EIDE implementation feature by which no participating agent in the electronic institution can be forced to go to a given scene. For the particular *mWater* example, neither the buyer nor the seller can be forced to go through the transition between the Trading Table scene and the Agreement Validation scene (see Fig. 1). This way, whenever the buyer and/or the seller goes to the Agreement Validation scene he/she starts the scene voluntarily, so norm *N2* is satisfied.

The implementation of norm *N3* requires a specific performative structure, named *Grievances* (Fig. 3), in order to deal with conflict resolution processes.

Finally, the observance of norm compliance is delegated to every water user. Hence, the enforceability of norm *N0* is delegated to every water user.

The implementation described above is fully NHP-compliant, but it leads to a low number of transfer agreement registration and, moreover, to the following very critical situation for the reliable execution of *mWater*. Let us suppose there is a water user A who has a water-right w_1 and wants to sell it. A starts a Trading Table inside the *TradingTables* process (see Fig. 1) in order to sell w_1. The water user B enters the Trading Table and as a result there is an agreement Agr_1 between A and B, by which B buys w_1 from A for the period $[t_1, t_2]$, and pays the quantity p_1 for such a transfer. A and B belong to $Basin_x$, in which norms *N1*, *N2* and *N3* apply. A and B do not register Agr_1 due to norm $N2$ (in other words, A and B do not go to the Agreement Validation scene of Fig. 1). Since there is no mechanism in $Basin_x$ by which water-right w_1 is blocked from A after its selling (due to Agr_1 is not registered and w_1 is still owned by A in time periods not overlapped with $[t_1, t_2]$), A continues to operate in the market. Afterwards A starts a new Trading Table to sell w_1 for period $[t_3, t_4]$, with $t_1 < t_3 < t_2$ and $t_4 > t_2$ (the new period $[t_3, t_4]$ is overlapped with $[t_1, t_2]$). In this second Trading Table A and C sign Agr_2, by which A sells w_1 to C for the period $[t_3, t_4]$ and C pays p_2 to A. A and C belong to $Basin_x$. In this case C registers Agr_2 in the *Agreement Validation* scene, due to *N1* and *N2*, and obtains the basin approval for executing Agr_2. At time t_3 (the transfer starting time) C attempts to execute Agr_2, but there is no water in the water transportation node, since B is also executing Agr_1. At this moment C has a conflict with B, and in order to solve it he/she has to start a grievance procedure due to *N3*.

Although the previous described situation is critical, *mWater* can overcome it thanks to the Grievance performative structure. When C cannot execute Agr_2 (because there is no water in the water transportation node), C believes that B is not complying norm *N0*. C believes there is a conflict because Agr_2 endows him/her to use the water, and moreover, there is no transfer agreement published in the Trading Hall that endows B to do the same. In order to enforce norm *N0* and to execute Agr_2, C starts a grievance procedure. In this procedure, water users C and B are recruited as conflicting parties and A as third party because he/she is the seller of w_1 as stated in Agr_2 (Recruiting Conflicting Parties scene of Fig. 3). Let us assume C chooses as conflict resolution mechanism arbitration, because he/she does not want to negotiate with B. After stating the grievance, C and B present their allegations to the jury. In this process B presents Agr_1 by which he/she believes there is fulfillment of norm *N0*. However, in the last step, by means of a Normative Reasoning function, the jury analyzes the presented allegations and the normative regulations of the basin and deduces that there is an offense. Both B and A do not conform with norm *N1*, and additionally, A has sold the same water right twice within an overlapped time period. In this last step, the jury imposes the corresponding sanctions to A and B.

Table 1. Market behaviour with varying regulations

Regulation	*Neg. Tables*	*Agreements*	*Water Volume*	*Conflicts*	*Periods*
$\{N0, N1, N2, N3\}$	100	48,3	12.352	23,5	5
$\{N0, N1, N2', N3\}$	100	25,6	3.521	6,2	5

mWater allows to simulate changes in the regulation in order to test what will happen, for example, if norm *N2* is replaced by norm *N2'*:

mWater - Local norm - (N2'): The registration process of a water-right transfer agreement is started automatically by the institution whenever a water-right transfer negotiation ends successfully.

Norm *N2'* is implemented directly in the Negotiation Table, allowing the Negotiation Table Manager to monitor all the negotiation protocol in order to detect the *Accept* message for a given bid. When the message is detected the Negotiation Table Manager informs the Basin Authority of the new agreement and it gets automatically published in the Trading Hall. In this way if the policy maker decides to include norm *N2'* as a regulation for the market in a given simulation the market participants that negotiated a successful agreement in a Negotiation Table go directly to Contract Enactment scene without going through Agreement Validation, and all the water user in the market get informed of the new agreement.

Recall in the previously described situation the first transfer agreement Agr_1. Norm *N2'* makes Agr_1 public just after the *Accept* message issued by A or B in the given Negotiation Table. In this way, when A tries to open a new Trading Table to sell w_1 for the period $[t_3, t_4]$, the Market Facilitator verifies that the new period overlaps with the period associated in Agr_1 (that affects the same water right) and consequently rejects the request of A. In this way, norm *N2'* reduces the number of conflicts caused by second selling of the same water right.

In order to test the market behaviour with the different group of norm regulations described above, we executed various simulations in *mWater* varying the regulation and the simulated period (5 different periods) with the same group of water users. Table 1 shows the results of these evaluations. From this table, we can conclude that regulation $\{N0, N1, N2, N3\}$ leads to a higher number of agreements (Table 1 reflects both registered and non-registered agreements) and indeed a higher amount of water transferred. Unfortunately, the number of conflicts is also higher. On the other hand, $\{N0, N1, N2', N3\}$ leads to fewer agreements, not only because the water-rights cannot be re-sold any more, but because not all the water users wanted to participate in the market due to the obligation to make public all the transfer agreements[3]. In order to evaluate other type of market reactions to regulation changes we are now working on "social"

[3] This situation happens in Spanish basins, and it was deliberately included in the agent behaviour of the water user participants in order to observe its effects on the market.

functions in order to asses values such as the trust and reputation levels of the market, or degree of water user satisfaction, among others. We believe that this type of measures will provide the policy makers with valuable data for decision making about new or modified regulations.

5 Conclusions and Future Work

As a whole, *mWater* constitutes a rather sophisticated regulated open multi-agent system. It is designed with three objectives in mind. First, as a demonstrator in the AT project (www.agreement-technologies.org), it provides a testing environment and inspiring problem domain for conceptual proposals and tools. Second, it may be used as the demand component of a sophisticated basin model to simulate, visualize and explore different water management policies, users and norms. That is, it helps explore the interactions between the basin hydrographic resources and infrastructures, together with the use of water as it is being modulated by market mechanisms and policy directives and regulations. Third, given the possibility of the creation of an actual market for water rights or analogous public goods, *mWater* would be a first proof of concept version to build upon.

The work we report in this paper provides insights on the regulated environment of *mWater* as an Electronic Institution for simulating water-rights markets. We are now developing a richer normative regulation in order to allow to simulate different types and group of norms. We are also working on defining performance measures that can evaluate 'social" issues in the market behaviour. At the same time we are developing different populations of water users in order to simulate varying type of members in a basin and to observe what are the effects of a given regulation when different type of water users are interacting the market.

Acknowledgements

This paper was partially funded by the Consolider programme of the Spanish Ministry of Science and Innovation through project AT (CSD2007-0022, INGENIO 2010) and MICINN project TIN2008-06701-C03-03. This research has also been partially funded by the Valencian Prometeo project 2008/051.

References

1. J. Arcos, M. Esteva, P. Noriega, J. Rodriguez-Aguilar, and C. Sierra. Engineering open environments with electronic institutions. *Engineering Applications of Artificial Intelligence*, (18):191–204, 2005.
2. X. Cai, L. Lasdon, and A.M. Michelsen. Group decision making in water resources planning using multiple objective analysis. *Journal of Water Resources Planning and Management*, 130(1):4–14, 2004.
3. J. Calatrava. Mercados y bancos de agua en España. *Agricultura Familiar en España*, pages 99–105, 2006.

4. R. C. Ellickson. *Order without law : How neighbors settle disputes.* Harvard University Press, Cambridge, Mass. :, 1991.
5. J. Honey-Roses. Assessing the potential of water trading in Spain. In *ENR 319 Advanced International Environmental Economics. Professor Theo Panayotou at Harvard's John F. Kennedy School of Government*, 2007.
6. F. Martin de Santa Olalla, A. Dominguez, F. Ortega, A. Artigao, and C. Fabeiro. Bayesian networks in planning a large aquifer in Eastern Mancha, Spain. *Environmental Modelling and Software*, 22:1089–1100, 2007.
7. R.N. Palmer, W.J. Werick, A. MacEwan, and A.W Woods. Modeling water resources opportunities, challenges and trade-offs: The use of shared vision modeling for negotiation and conflict resolution. In *Proc. of the Water Resources Planning and Management Conference*, 1999.
8. V. Rajasekaram and K.D.W. Nandalal. Decision support system for reservoir water management conflict resolution. *Journal of Water Resources Planning and Management*, pages 410–419, 2005.
9. L. Riesgo and J.A. Gomez-Limon. Mercados del agua. analisis de las opciones elegidas para su aplicacion en España. In *Proc. of the IV Congreso Nacional de Economía Agraria*, 2004.
10. T. Schultz, G. Kaufmann-Kohler, D. Langer, and V. Bonnet. Online dispute resolution: The state of the art and the issues. In *Available at SSRN: http://ssrn.com/abstarct=899079.*
11. W.K. Slate. Online dispute resolution: Click here to settle your dispute. *Dispute Resolution Journal*, 56(4):8–14, 2002.
12. M. Thobani. Formal water markets: Why, when and how to introduce tradable water rights. *The World Bank Research Observer*, 12(2):161–179, 1997.

Semantic Service Discovery in MAS Using Social Networks

E. del Val, M. Rebollo, and V. Botti

Grupo de Tecnología Informática - Inteligencia Artificial
Departamento de Sistemas Informáticos y Computación
Universidad Politécnica de Valencia
Camino de Vera S/N 46022 Valencia (Spain)
{edelval,mrebollo,vbotti}@dsic.upv.es

Abstract. Service-Oriented Multi-Agent Systems are dynamic systems populated by heterogeneous agents. These agents model their functionality as services in order to allow heterogeneous agents or other entities to interact in a standardized way. Furthermore, due to the large-scale and the adaptive needs of the system, the traditional directory facilitators or middle-agents are not suitable for the management of the agents services. In this paper we present a distributed system where there is no a ce. The proposal provides a fully decentralized structure and allows agents to locate services using only local information. The system is enhanced using semantic information in the generation of the system structure and also in the search process.

1 Introduction

Service-Oriented Multi-Agent Systems (SOMAS) can be described as open and dynamic systems, where agents provide basic functionality through services and new agents can enter to the system and existing ones leave. An important issue that has raised great interest in the research community in the latest years is service discovery. In open systems where there is a large number of agents and the available agents change dynamically, finding the appropriate agent which offers the service required is not an easy task. Conventional approaches to locate agents with certain functionality in SOMAS, such as registries or middle-agents, are centralized approaches which are not always appropriated for large-scale and highly dynamic environments. These proposals present some weakness such as bottlenecks, complexity or the huge amount of memory needed to keep all the information about the agent's functionality when the system scales. Distributed approaches, such as agent coalitions or federations of registries, have been proposed to solve some of these problems but the required coordination effort to create the coalitions and to maintain data consistency between distributed registries makes these proposals not suitable for highly dynamic environments.

An alternative for traditional proposals is the use of social networks[25][28]. Human beings create social structures in a decentralized way which allows to locate other individual in a few steps considering only local information. This fact was observed by Milgram in the well known experiment of 'six degrees of separation'[24]. The results of this experiment arose two questions: how is the structure of these social networks and how an effective search of individuals is carried on only with local information. In the wake of this experiment, several works started to pay attention on the analysis of the underlying structures in human societies and the properties of these structures.

As a result of that, several models based on mathematical functions have been proposed to simulate the structure in real social networks. These models try to reflect how social links are established between individuals to form a network which can guide the search. These networks such as small-world or preferential attachment network, are called *navigable social networks* and it can be ensured that short paths between two random individuals can be found using only local information. How an effective search is carried on only with local information is the other important aspect. Which criteria should be follow by the individuals in order to guide the search towards the target? This depends on the structure of the network. There are some strategies that have better results depending of the underlying structure where it is applied. For instance in small-world networks similarity or geographical distance can be considered a good parameters to consider while strategies guided by degree are not so suitable.

In this work, we propose the use of a social network model as the underlying structure of a service discovery system for agents. The structure relies on a social feature present in many social networks called *homophily*[15]. *Homophily* expresses the idea that similar people interact with higher frequency than dissimilar people. Therefore, in our system agents with similar roles and services have more probability to be linked. The system provides a fully decentralized structure and allows agents to locate services using only local information. The system is enhanced using semantic information in the generation of the system structure and also in the search process.

2 Related Work

Open and dynamic environments where the scalability and the workload are low make use of middleagents to facilitate service discovery [14][22][23]. The matchmakers could provide an optimal matching due to they consider all the registered services in the system. Unfortunately, this kind of agents could be a bottleneck when the workload increases. Other drawbacks are their complexity, the huge amount of memory needed to keep service advertisements and the cost of service composition as the number of services grows significantly. Different approaches have been suggested to overcome the above mentioned problems. *Peer-to-peer* approaches [12][2][20][31] broadcast a query using local knowledge The drawback of this approach to service discovery is that the communication among agents is essential and the overall communication traffic overhead may be large. Another distributed way to locate distributed services is to form *coalitions* or clusters[19][18][16]. Nevertheless, the choice of what coalitions are going to be formed is a difficult task. This entails recursively to calculate the values of the coalitions and later selecting the coalition with the best result. A third way for agents to discover services in efficiently is the distribution of the *middleagents* or *facilitators* [21] [17] [13]. These proposals suggest to split the function of the service facilitator among a group of agents. The system designer assigns a local matchmaker to each host or segment of the system, which provides matchmaking services to agents in its vicinity (its segment). In systems with very large segments the problems of scalability are only marginally relieved by this approach because the large segments become overloaded systems which have local bottlenecks. Another case in which this approach is not useful is in systems with many cross-links between segments. In this case the overhead of coordinating tasks among local matchmakers might be greater than the benefit obtained from their distribution.

3 Proposed System and Definitions

The proposal that we present here tries to overcome drawbacks of current discovery approaches in open SOMAS through a completely distributed approach, considering semantic information about organizational roles and services. This approach is based on social networks as underlying structure. The advantages and contributions of this proposal compared to others are:

- System which integrates services and agents. Agents have social and proactive capabilities which provide more flexibility and adaptability to the system. Services facilitate the reusability and interoperability. Here, agents functionality is described in terms of services, therefore we obtain the advantages of both technologies.
- System structure that guarantees, in general, that a service if it exits is going to be found in a bounded number of steps.
- Service discovery strategy that only needs local information to navigate the network in order to reach the required service.
- The use of semantic information to create the system structure and to lead the service search.
- Inclusion of organizational information in the service discovery process.

DEFINITION **1** *(Agent-Service Discovery System). An* SDS *is defined as* $\mathcal{SDS}$ = $(\mathcal{A}, \mathcal{L})$*, where* $\mathcal{A}$ *is the set of agents that are part of the* $\mathcal{SDS}$ *(nodes):* $\mathcal{A}= \{a_1,...,a_n\}$*, and each edge* $\ell=(a_i, a_j)\in\mathcal{L}$ *indicates the existence of a knowledge or communication relationship between agent* a_i *and* a_j *in the system (undirected links).*

Agents are social entities which have local knowledge about its immediate neighbors, including their identity, degree, organizational information and the semantic description of the services they offer, but it is unaware of the rest of the agents present in the system.

DEFINITION **2** *(Agent). An agent* $a_i=(\mathcal{R},\mathcal{N})$ $|a_i \in \mathcal{A}$ *is a social entity which can play several roles in different organizational units* $\mathcal{R}=\{r_1,\ldots,r_n\}$*:*$|\mathcal{R}| > 0$*, has a neighborhood* $\mathcal{N}=\{a_k,...,a_n\}|$ $a_k \in \mathcal{N}$*,* $\exists$ $(a_i,a_k)\in\mathcal{L}$*,* $|\mathcal{N}| > 0$*.*

The agent role determines the kind of services an agent offers. The role provides an abstract layer over the services that the agent offers and it is used to create the structure of the system. Roles are defined inside an organization unit ou. The organization unit establishes a set of policies responsible of the structure of the system. These policies are related to basic system operations (*join*, *leave*, *discover*...).

DEFINITION **3** *(Role). A role in our system is defined as* r = $(\phi,\mathcal{S},ou) \in \mathcal{R}$ *where* ϕ *is a semantic concept for the role,* ou *is the organization where the agent plays the role* r *and* $\mathcal{S}$=$\{s_1,\ldots s_n\}$ *is the set of services offered by the agent.*

Each *service* s_i is a semantic service defined by the tuple: s_i = $(\mathcal{I},\mathcal{O})$, where $\mathcal{I}$ denotes the set of inputs and $\mathcal{O}$ denotes the set of outputs. The $\mathcal{I}$ and $\mathcal{O}$ of the service are semantic concepts defined in a common ontology. To simplify the system, we are going to consider that each agent plays one role and offers one service.

The agent-service discovery system that we present relies on a property present in many real social networks: homophily. This word expresses the idea that similar people tend to interact and establish links with higher probability than dissimilar people. There are two types of homophily[15]:

- *choice homophily*, where patterns of interaction are driven by preferences for similarity. This kind of homophily has two forms: *status homophily*, where the individuals are considered similiar if they share a cultural background, and *value homophily*, where individuals are considered similar on the basis of shared values, attitudes, and believes.
- *induced homophily*, emerges not from individual choice, but from influence dynamics that make individuals more similar over time[29].

In this work we focus on choice homophily and its two forms. In general, homophily has demonstrated that is one of the most pronounced features in social networks[3][27]. Due to the efficiency of the social networks with this feature, we consider important to consider this property in our system.

DEFINITION **4** *(Agent homophily). In our* $\mathcal{SDS}$ *the homophily between two agents is based on the status homophily and the value homophily:*

- *value homophily* ($\mathcal{H}_v(\mathcal{S}_i,\mathcal{S}_j)$) *is defined over the agent's services and it is considered as the semantic similarity between the services offered by the agents.*
- *status homophily* ($\mathcal{H}_s(r_i,r_j)$) *is defined over the agent's role and it is considered as the semantic similarity between the roles played by the agents*

Therefore, the homophily between two agents is defined as the linear combination of value and status homophily:

$$\mathcal{H} = \alpha * \mathcal{H}_v + (1 - \alpha) * \mathcal{H}_s \tag{1}$$

We are going to describe with more detail how are calculated each kind of homophily. The homophily function $\mathcal{H}_s(r_i,r_j)$, means the degree of match dom (*exact*, *subsumes*, *plug-in*, *fail*) between the semantic concept of the roles played by the agents.

$$\mathcal{H}_s(r_i, r_j) = role\_match(r_i.\phi, r_j.\phi) \tag{2}$$

where *role_match* is the function which calculates the semantic similarity between $r_i.\phi_i$ and $r_j.\phi_j$. The homophily function $\mathcal{H}_v(r_i.\mathcal{S},r_j.\mathcal{S})$, means the degree of match between the services offered by the agents.

$$\mathcal{H}_v(r_i.\mathcal{S}, r_j.\mathcal{S}) = \beta * match(\mathcal{I}_i, \mathcal{I}_j) + (1 - \beta) * match(\mathcal{O}_i, \mathcal{O}_j) \quad (3)$$

$$\mathcal{I} = \bigcup_{\forall s \in r.\mathcal{S}} s.\mathcal{I}, \mathcal{O} = \bigcup_{\forall s \in r.\mathcal{S}} s.\mathcal{O} \quad (4)$$

$$match(\mathcal{I}_i, \mathcal{I}_j) = max(W_{G'=(\mathcal{I}_i \cup \mathcal{I}_j, E')}) \quad (5)$$

where function $match$ solves a *bipartite matching problem* between semantic services. In our case we have two bipartite graphs, one where the vertexes are the inputs of the services and the other where the vertexes are the outputs. In the case of matching between the inputs of the services (the process is the same for the outputs), the bipartite graph G=($\mathcal{I}_i \cup \mathcal{I}_j$, E) has a set of vertexes with $\mathcal{I}_i$ inputs and the other set with $\mathcal{I}_j$ inputs. Given a bipartite graph for the service inputs, G=($\mathcal{I}_i \cup \mathcal{I}_j$,$E$), its matching G'=($\mathcal{I}_i \cup \mathcal{I}_j$,$E'$) $E' \subseteq E$ is a graph where all the vertexes of one set are connected with the other set of vertexes only with one edge. In this graph, we allow that edges share a vertex to give more flexibility to the matching. The sum of weights ($W_{G'}$) of the edges in the matching is maximized:

$$W_{G'=(\mathcal{I}_i \cup \mathcal{I}_j, E')} = \frac{\sum_{\forall e_k \in E'} \omega_k}{max(|I_i|, |I_j|)} \quad (6)$$

4 System Operations

4.1 Join

The process that an agent should follow to get into the $\mathcal{SDS}$ is as following (see Alg. 1): the agent a_i tries to establish a set of connections with other agents already present in the system. The number of connections that the agent is going to establish is generated by a random function which follows an exponential distribution. The idea is to generate a system with an exponential degree distribution to achieve the structure of a *preferential attachment network*[1]. A preferential attachment network it is characterized by a degree distribution which follows a power-law degree distribution, p(dg) $\propto dg^{\lambda}$, where $p(dg)$ indicates the probability to be connected to a node with degree dg. This means that there are some nodes have a high degree and the majority has a low degree. This structure ensures that the diameter of the network is $ln\ |\mathcal{A}|$, where $|\mathcal{A}|$ is the number of agents in the $\mathcal{SDS}$ and in some situations, when 2<λ<3, is $lnln\ |\mathcal{A}|$[11]. This model is present in many *'online communities'* such as *WWW*, electronic mail or citation graphs [26]. These networks are the result of a growth process in which new nodes that join the system prefer to be connected to well connected nodes.

Once the agent knows the number of connections, it should decide which agents are going to be its neighbors. The probability of an agent a_i to establish a connection with agent a_j is directly proportional to the homophily degree between the agents, if the agents are more similar, they have more probability to be connected. This condition allows a new agent not only to establish 'short connections' between agents with similar roles and semantic services, but also between agents that are not similar ('long connections'). The idea of 'long connections' is to create short paths between groups of agents that do not offer similar services and reduce the number of hops needed to discover services. The probability to establish a connection between two agents in the system is based on the homophily degree between them $\mathcal{H}(a_i,a_j)$.

Algorithm 1 Join, where a is the new agent and SD the system

```
function Join(a, SD)
connections ← ExpRandom(λ)
connected ← False
dg ← 0
while ¬connected ∧ dg ≤ connections do
  a_r ← random(SD)
  if H(a, a_r) ≥ UniRandom(0,1) then
    ℓ(a,a_r)
    a.dg ← a.dg+1
    updateNeighbors(a, a_r)
    connected ← True
  end if
end while
end function
```

4.2 Leave

When an agent leave the system could be for three reasons: the agent decide voluntary to leave the system, failure or 'sabotage'. Periodically an agent sends a *keep alive* message to its neighbors. The agent will notice that one of its links is broken whether after sending a message, the time to receive an answer from the neighbor expires. In that case, the agent deletes the neighbor from its neighbor list and establishes a new link with other agent in the network to keep their degree (see Alg. 2 and Alg. 3).

Algorithm 2 Leave, where a is the agent and SD the system

```
function Leave(a, SD)
local
for a_i ∈ a.N do
  removeLink(a, a_i)
  newLink(a_i,SD)
end for
end function
```

4.3 Search

DEFINITION **5** *(Service discovery problem) Given a set of agents $\mathcal{A}$ situated in a $\mathcal{SDS}$= ($\mathcal{A}$,$\mathcal{L}$), the **service discovery problem** is defined as a probabilistic decision-making task in which an agent $a_i \in \mathcal{A}$ is looking for an agent $a_j \in \mathcal{A}$, which offers the required service s_t.*

Algorithm 3 newLink, where a is the agent and SD the system

```
function Link(a, SD)
  connected←False
  while ¬connected do
    a_r←random(SD)
    if sim(a, a_r) ≥ UniRandom(0, 1) then
      ℓ(a,a_r)
      a_r.dg←a_r.dg+1
      updateNeighbors(a, a_r)
      connected←True
    end if
  end while
end function
```

The search process in the system is based only on the agent local knowledge. When the agent a_i is looking for an agent a_j which offers the required service s_t, a_i selects which of its neighbors is the most appropriated to redirect the query instead of broadcast the query to all the neighborhood. In many networks which reflects power-law characteristics, the search is suggested to be based on degree. However, this makes that highly connected nodes could be overloaded with requests. Our selection criteria is based on previous proposals presented in [27][30], where the selection of the most promising neighbor is based on two criteria: *degree* and the *similarity*. In our sytem the selection criteria is based on the agent degree and the semantic similarity between agents services and roles. Until the target agent a_j is found, all future agents involve in the discovery process will make their decision similarly (see Alg. 4).

Algorithm 4 Search where a_s is the source agent, s_t is the required service and S the system and Θ is the similarity threshold

```
function Search(a_s, s_t, S, Θ)
  s←getService(Ag_s)
  a←a_s
  steps←0
  while sim(s, s_t) ≥ Θ ∧ steps≤TTL do
    p_max←0
    for a_i ∈ a.𝒩 do
      dg ← a.dg
      s ← a.s
      p ← P(ℋ(a, a_i),dg)
      if p > p_max then
        p_max ← p
        a ← a_i
      end if
    end for
  end while
  return a
end function
```

5 Simulation Results

The test can be divided in two groups. The first group compares the performance of typical distributed search strategies (*degree*, *similarity*, *random*) to the proposal presented in this paper. The second test evaluates the fault tolerance of the $\mathcal{SDS}$ when relationships between agents in the system are broken randomly (an agent leaves the system) or following some patterns (which corresponds to deliberate failures 'sabotage').

5.1 System Characterization

The experiments have been done in a set of networks that simulate the $\mathcal{SDS}$ structure. These networks are preferential attachment networks and have generated as the result of the *join* operation of agents. We have implemented two kind of networks: A where the agents have not roles and Network B where each agent plays a role. We consider 10 types of different roles. Each network is composed of 1000 agents with one semantic service each one. The services and roles have been assigned to the agents using a uniform distribution.

5.2 Performance

In this section we evaluate the *search* operation in our $\mathcal{SDS}$. Due to the similarities of our system and p2p systems, we compared the *search operation* to other typical search strategies used in p2p systems: *random*, *degree*, *similarity*, *similarity and degree*. We have analysed the behavior of each strategy in 5000 searches in networks A and B.

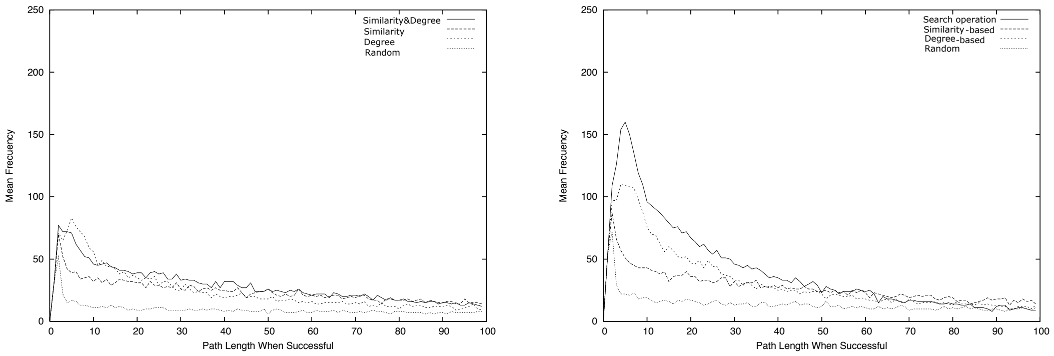
In figures 1a and 1b, the results obtained after the service search process are presented. We see that in general the strategies in a network with organizational information have a better performance than the same strategies in a network without this information. That shows that organizational information in the system can guide the search process better than the systems that only provide information related to the degree and services. Between all the strategies, the *search* operation that we present in this paper has a better performance than the others. This is because it considers, apart form the degree and semantic service information, the roles that agents play. This information reduces the set of possible agents suitable to offer the service.

An important parameter to consider in $\mathcal{SDS}$ is the number of steps to reach the target agent. Figure 2a shows the mean path length obtained with each strategy in networks with role information (B). In general, all the strategies return paths with more steps as the number of agents in the network grows. When the size of the network is over 700 agents, the path length does not increase significantly. This shows that the structure of the $\mathcal{SDS}$ is suitable for large-scale systems.

In Figure 2b the success rate of each search strategy in $\mathcal{SDS}$ is depicted. An obvious result is that as the system scale increases, the percentage successful searches decreases. The *search* operation presented here is the algorithm less influenced by the number of agents in the system. In general, the *search* operation in the 80% of searches finds a path between the source agent to the target agent.

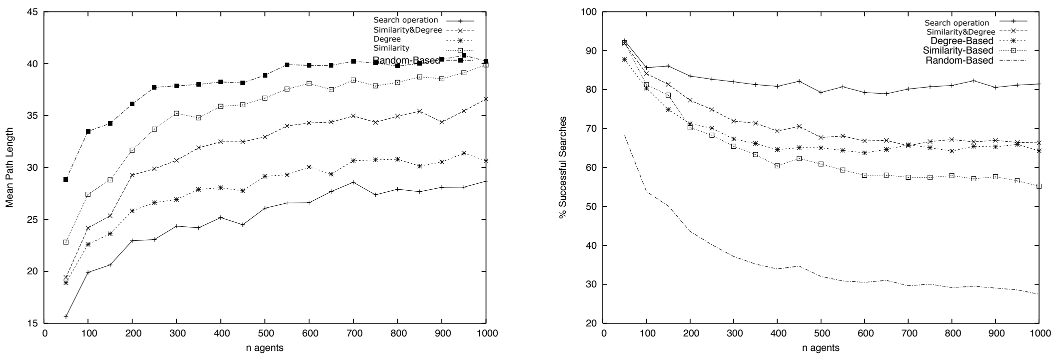
5.3 Fault Tolerance

The last and very important check is the behavior of the $\mathcal{SDS}$ under failures. The problem appears when a broken link splits the system into two isolated parts, since some



(a) (b)

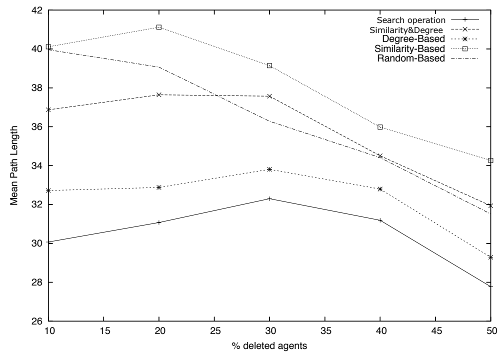
Fig. 1: Search performance without/with role information



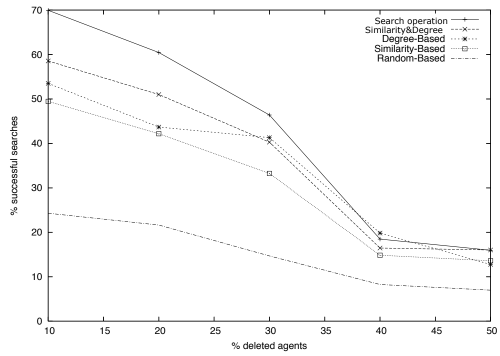
(a) (b)

Fig. 2: Mean path length and success

agents will no longer be reachable. To analyse it, agent failures have been modelled as a failure of all its connexions. When some links are broken, an alternative path has to be found. For random failures (see Fig.3a and Fig.3b), it can be observed that when the number of deleted agents is from 10% to 30%, the path length increases, due to there are alternative paths (with more steps) to find the agent with the required service. When the number of deleted agents ranges from 30% to 50%, the network is divided in several isolated parts. Only the searches inside the isle will success, so the number of successful searches decreases and the path length decreases because the isle diameter are smaller.



(a)



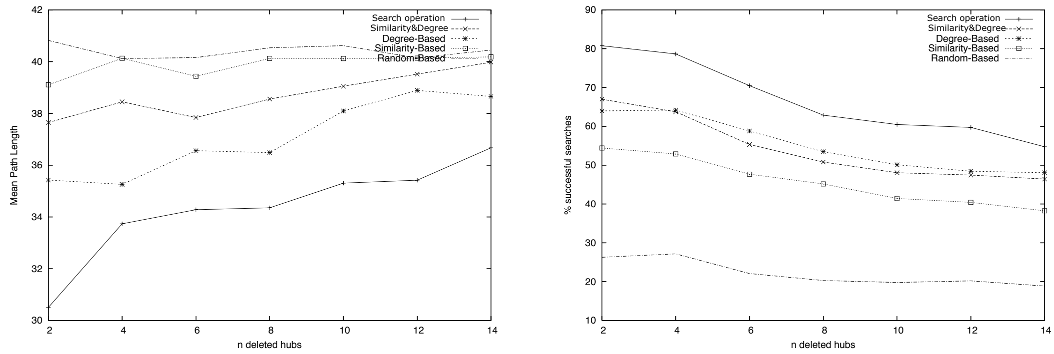
(b)

Fig. 3: Mean path length and success with random failures

An interesting case is what happens when a deliberate failure is provoked. In the case of systems that follow a power-law, the worst case occurs when agents with highest degree (hubs) are disconnected. Figure 4a and 4b shows how *'sabotage'* affects the performance of the search process. In this case, the path length increases due to only a few highly connected hubs have been deleted and an alternative path exists. The performance attending the number of successful searches decreases considerably as the number of deleted hub increases.

6 Conclusions

The aim of this work is to provide an alternative to traditional approaches that deal with the service discovery task in large-scale open SOMAS. Our proposal tries to overcome drawbacks present in other centralized (bottlenecks, complexity, huge amount of memory needed, global knowledge) and distributed (network traffic, congestion, coordination effort, data consistency between distributed registries, update data) discovery approaches. We consider that structures used in social networks facilitate the task of locating agent services in a few steps using only local information. For that reason we investigate the use of social networks as underlying structure of a service discovery system. This structure is based on the concept of similarity between individuals, considering organization role and services, and uses semantics to calculate this similarity. Furthermore, we provide several operations for the agents to be part of the system. An evaluation of the search functionality compared to other traditional p2p strategies is also



(a) (b)

Fig. 4: Mean path length and success under 'sabotage' conditions

provided. The behavior of the system under failure and 'sabotage' circumstances have been also evaluated. The results of the experiments show that the system is robust under failure and that the search functionality performs well.

Acknowledgment

This work is supported by TIN2009-13839-C03-01 and TIN2008-04446 projects, CONSOLIDER-INGENIO 2010 under grant CSD2007-00022, FPU grant AP-2008-00601 awarded to E. del Val.

References

1. Barabasi, A. L. and Albert, R. Emergence of Scaling in Random Networks. *Science*, 286(5439):509–512, 1999.
2. E. Bircher and T. Braun. *An Agent-Based Architecture for Service Discovery and Negotiation in Wireless Networks*. 2004.
3. Basit Chaudhry, Chris Marton, and Hana Shepherd. Homophily and structure in multiplex networks.
4. Miller, Lynn, and James M. Birds of a feather: Homophily in social networks. *Annual Review of Sociology*, 2001.
5. M. Moore and T. Suda. A decentralized and self-organizing discovery mechanism. In *Proc. Of the First Annual Symposium on Autonomous Intelligent Networks and Systems*, 2002.
6. Jeffrey Travers and Stanley Milgram. An experimental study of the small world problem. *Sociometry*, 1969.
7. Yamini Upadrashta, Julita Vassileva, and Winfried Grassmann. Social networks in peer-to-peer systems. paper presented at the. In *38th Hawaii International Conference on System Sciences*, pages 3–6, 2005.

8. Alexei Vázquez. Growing network with local rules: Preferential attachment, clustering hierarchy, and degree correlations. *Physical Review E*, 67(5), May 2003.
9. Duncan J. Watts, Peter Sheridan Dodds, and M. E. J. Newman. Identity and search in social networks. *Science*, 296:1302–1305, 2002.
10. Hui Zhang, Ashish Goel, and Ramesh Govindan. Using the small-world model to improve freenet performance, 2002.
11. R. Cohen and S. Havlin. Scale-Free Networks Are Ultrasmall. *Physical Review Letters*, 90(5), February 2003.
12. J. Dang and M. Hungs. *Concurrent Multiple-Issue Negotiation for Internet-Based Services*. Number Vol.10 - 6. 2006.
13. S. Jha, P. Chalasani, O. Shehory, and K. Sycara. A formal treatment of distributed matchmaking. In *Proc. of the 2nd Int. Conference on Autonomous Agents*, number Vol.3, pages 457–458, 1998.
14. M. Klusch, B. Fries, and K. Sycara. Automated semantic web service discovery with owls-mx. In *AAMAS*, 2006.
15. Miller, Lynn, and James M. Birds of a feather: Homophily in social networks. *Annual Review of Sociology*, 2001.
16. M. Moore and T. Suda. A decentralized and self-organizing discovery mechanism. In *Proc. Of the First Annual Symposium on Autonomous Intelligent Networks and Systems*, 2002.
17. S. Mullender and P. Vitanyi. *Distributed Match-Making*. Number Vol.3. 1988.
18. E. Ogston and S. Vassiliadis. Local distributed agent matchmaking. In *Proceedings of the 9th International Conference on Cooperative Information Systems*, 2001.
19. E. Ogston and S. Vassiliadis. Matchmaking among minimal agents without a facilitator. In *Proceedings of the 5th International Conference on Autonomous Agents*, pages 608–615, 2001.
20. A. Ouksel, Y. Babad, and T. Tesch. Matchmaking software agents in b2b markets. In *Proceedings of the 37th Annual Hawaii International Conference on System Sciences (HICSS'04)*, 2004.
21. K. Sigdel, K. Bertels, B. Pourebrahimi, S. Vassiliadis, and L.S. Shuai. A framework for adaptive matchmaking in distributed computing. In *In proceeding of GRID Workshop*, 2005.
22. K. Sycara and M. Klusch. Brokering and matchmaking for coordination of agent societies: A survey. *Coordination of Internet Agents: Models, Technologies and Applications*, 2001.
23. Katia Sycara, Matthias Klusch, Seth Widoff, and Jianguo Lu. Dynamic service matchmaking among agents in open information environments. *SIGMOD Record*, 28:47–53, 1999.
24. Jeffrey Travers and Stanley Milgram. An experimental study of the small world problem. *Sociometry*, 1969.
25. Yamini Upadrashta, Julita Vassileva, and Winfried Grassmann. Social networks in peer-to-peer systems. paper presented at the. In *38th Hawaii International Conference on System Sciences*, pages 3–6, 2005.
26. Alexei Vázquez. Growing network with local rules: Preferential attachment, clustering hierarchy, and degree correlations. *Physical Review E*, 67(5), May 2003.
27. Duncan J. Watts, Peter Sheridan Dodds, and M. E. J. Newman. Identity and search in social networks. *Science*, 296:1302–1305, 2002.
28. Hui Zhang, Ashish Goel, and Ramesh Govindan. Using the small-world model to improve freenet performance, 2002.
29. M. McPherson, L. Smith-Lovin, and J. Cook. Birds of a feather: Homophily in social networks. *Annual Review of Sociology*, 2001.
30. Şimşek and Jensen. Navigating networks by using homophily and degree. *NAS*, 2008.
31. Antnio L. Lopes and Lus M. Botelho. Improving Multi-Agent Based Resource Coordination in Peer-to-Peer Networks. Journal of Network, 2008.

Trust and Belief, Interrelation

Besik Dundua[1] and Levan Uridia[2]

[1] DCC-FC & LIACC, University of Porto, Portugal
bdundua@dcc.fc.up.pt
[2] Universidad Rey Juan Carlos de Madrid, Spain
uridia@ia.urjc.es

Abstract. We introduce the modal system $\mathbf{B}^2_T$ which is a multi-modal language designed to talk about trust and belief of two agents. The belief operators are based on modal system **KS**. This gives the main difference with already known system **BA** introduced by Churn-Jung Liau [2] and also carries its own intuitive meaning. As a main result we prove that $\mathbf{B}^2_T$ is sound and complete with respect to the given semantics, which is a mixture of the Kripke and neighborhood semantics.

1 Introduction

In a notion derived from Plato's dialogue Theaetetus, philosophy has traditionally defined *knowledge* as *justified true belief*. The relationship between belief and knowledge is that a belief is a knowledge if the belief is true, and if the believer has a justification for believing it is true. Dropping out the part about justification we can express this connection in a standard modal language by well known translation Sp (splitting translation from **S5** to **KS**), The main clause of which states: $Sp(\mathcal{K}p) = \mathcal{B}p \wedge p$. Mainly because of this connection together with some doxastic properties of the axiom system, the modal logic **KS** could be adopted as a good concurrent to the classical doxastic logic **KD45**. First introduced in the 1980s by Segerberg as a modal logic of *some other time* [8], **KS** was rediscovered as a modal logic of *inequality* [5], [7] and recently it was investigated as a doxastic logic which carries quite adequate properties for modelling beliefs of agents [1]. In this paper we consider only those agents whose doxastic properties are formalized in the modal logic **KS**.

Different types of trust have been proposed and studied in the disciplines like philosophy, economics, computer science, etc. In this paper we focus on the interrelation of trust and belief in the two agents case. On the one hand, we follow the ideas introduced in [2], but on the other hand, we simplify the language in a sense that we leave only two types of modalities: $\Box_i$ as a belief operator of agent i and $T_{i,j}$ for trust of agent-i in agent-j. So the logic $\mathbf{B}^2_T$ we introduce is especially designed to talk about belief and trust and their interrelation. As for belief operators, as we already mentioned, they satisfy the axioms of **KS** and this makes the main difference with the logic **BA** discussed in [2].

The paper is organized in the following way: In section 2 we recall the logic **KS**, provide basic definitions and some known facts. In section 3 we introduce the logic $\mathbf{B}^2_T$ and also give its semantics on Kripke's structures. In the same section we prove the main result of the paper which states that the logic $\mathbf{B}^2_T$ is sound and complete with respect to the given semantics.

2 The Modal Logic KS

In 1976 Krister Segerberg [8] explicitly formulated a modal logic **KS** in which the diamond modality $\Diamond$ is interpreted as "somewhere else". In this section we define the system KS and its Kripke semantics.

2.1 Syntax

Definition 2.11 *The normal modal logic* **KS** *is defined in a standard modal language with infinite set of propositional letters* $p, q, r, ..$ *and connectives* $\wedge, \Box, \neg$*,*

- *The axioms are all classical tautologies plus three axioms containing modal operators. Namely:*

$\Box(p \to q) \to \Box p \to \Box q,$

$\Box p \wedge p \to \Box\Box p,$

$p \to \Box\Diamond p,$ *where* $\Diamond p \equiv \neg\Box\neg p.$

- *The rules of inference are: Modus-ponens, Substitution and Necessitation.*

Observe that doxastic interpretation of the last axiom states that *If* p *is true then agent believes that it is not the case that he believes the negation of* p*.*

It is an easy task to show that if we add axiom $\Box p \to p$ to the **KS** we will get the classical epistemic system **S5**. Following Smullyan [9] this means that if the **KS**-reasoner is *accurate* (never believes any false proposition) then his beliefs coincide with his knowledge.

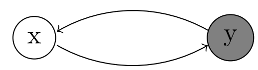
2.2 Kripke semantics

Kripke semantics for the modal logic **KS** is provided by weakly-transitive and symmetric Kripke frames. Below we give the definition of weakly-transitive relation.

Definition 2.21 *We will say that a relation* $R \subseteq W \times W$ *is weakly-transitive if* $(\forall x, y, z)(xRy \wedge yRz \wedge x \neq z \Rightarrow xRz)$*.*

Obviously every transitive relation is weakly-transitive also. Moreover it is immediate to notice that weakly-transitive relations differ from transitive ones just by the occurrence of irreflexive points in clusters. As you can see the frame on the picture is weak transitive, but not transitive.

The picture represents the diagrammatic view of Kripke structure, where irreflexive points are colored by grey and reflexive ones are uncolored. Arrows represent the relation between two distinct points. So as we can see yRx and xRy, but we do not have yRy, which contradicts transitivity, but not weak transitivity as $y = y$.



pic. 1

In the study of modal logic the class of rooted frames plays a central role. Recall that a frame (W, R) is rooted if it contains a point $w \in W$, which can see all other points in W. That is $R(w) \supseteq W - \{w\}$, where $R(w)$ is the set of all successors of w. The class of all rooted, weakly-transitive and symmetric frames can be characterized by the property which we call weak - cluster.

Definition 2.22 *We will say that a relation* $R \subseteq W \times W$ *is weak-cluster if* $(\forall x, y)(x \neq y \Rightarrow xRy)$.

It is easy to see that every weak-cluster is just a cluster where we allow irreflexive points. We will see the detailed characterization of finite weak-clusters in the next section. The following proposition makes the link between weak-clusters and rooted, weakly-transitive, symmetric frames.

Proposition 2.23 *A frame* (W, R) *is rooted, weakly-transitive and symmetric iff it is weak-cluster.*

Proof. It is immediate that every weak-cluster is rooted, weakly-transitive and symmetric frame. For the other direction let (W, R) be rooted, weakly-transitive and symmetric frame. Let $w \in W$ be the root. Take arbitrary two distinct points $x, y \in W$. As w is the root, we have: wRx and wRy. Because of symmetry we get xRw. Now as R is weakly-transitive, from $xRy \wedge yRw$ and $x \neq y$ we get xRy. Hence R is a weak-cluster.

For the sake of completeness we will just briefly state the main definitions, like: Kripke model, satisfaction and validity of modal formulas. These definitions are standard and can be found in any modal logic book.

Definition 2.24 *The pair* (W, R), *with* W *an arbitrary set (set of possible worlds) and* $R \subseteq W \times W$ *is called a Kripke frame.*

If we additionally have a third component $V : Prop \times W \to \{0, 1\}$, *then we say that we have a Kripke model* $M = (W, R, V)$ *(Here* $Prop$ *denotes the set of all propositional letters).*

The satisfaction and validity of a modal formula are defined inductively. We just state the base and modal cases here.

Definition 2.25 *For a given Kripke model $M = (W, R, V)$ the satisfaction of a formula at a point $w \in W$ is defined inductively as follows: $w \Vdash p$ iff $V(p, w) = 1$, the boolean cases are standard, $w \Vdash \Box\phi$ iff $(\forall v)(wRv \Rightarrow v \Vdash \phi)$.*

We will say that a formula ϕ is valid in a model (W, R, V) if for every point $w \in W$ we have $w \Vdash \phi$. We will say that a formula ϕ is valid in a frame (W, R) if it is valid in every model (W, R, V) based on a frame (W, R). We will say that a formula ϕ is valid in a class of frames C if ϕ is valid in every frame $(W, R) \in C$.

So far we defined the modal logic **KS** syntactically and we gave the definition of weak-cluster relation. The following theorem links these two notions:

Theorem 2.26 *[4] The modal logic* **KS** *is sound and complete w.r.t. the class of all* ***finite, irreflexive*** *weak-cluster relations.*

Mainly because of the theorem 2.26 the modal logic **KS** is called the modal logic of inequality. As the reader can easily check the interpretation of box in irreflexive weak-clusters boils down to the following: $w \Vdash \Box\phi$ iff $(\forall v)(w \neq v \Rightarrow v \Vdash \phi)$.

3 Trust

In this section we extend the language for multi-agent case and in addition we add modalities for trust. We restrict the language for the case with two agents as far as the other cases (for finite agents) follow as an easy generalizations of two agent case. We take the ideas from [2] and introduce modal logic which has enough expressive power to talk about trust and belief. We do not consider the same language as in [2], but just its fragment, since we are only interested in interrelation between trust and belief. We give the semantics for this logic and as a main result of the paper we prove the completeness of the described logic with respect to the semantics. Main crucial difference from [2] lies in the fact that the doxastic properties of agents follow **KS** axioms not **KD45** axioms as is classically adopted.

3.1 Syntax

The language consists of infinite set of propositional letters $p, q, r, ...$, connectives $\vee, \wedge, \neg, \rightarrow$, and modalities $\Box_1, \Box_2$ $T_{1,2}, T_{2,1}$

Axioms:Each $\Box_i$ satisfies **KS** axioms,

$\vdash T_{1,2}p \leftrightarrow \Box_1 T_{1,2}p$,
$\vdash T_{2,1}p \leftrightarrow \Box_2 T_{2,1}p$.

Rules of inference are: Modus ponens and substitution for each modality, necessitation for $\Box_i$ where $i \in \{1, 2\}$ and the following rule $\frac{\vdash p \leftrightarrow q}{\vdash T_{i,j}p \leftrightarrow T_{i,j}q}$ for each

$T_{i,j}$ with $i, j \in \{1, 2\}$ and $i \neq j$.

The desired interpretation of $T_{i,j}p$ carries the following idea: Agent-i trusts agent-j about the claim p. In these settings the last two axioms have very intuitive meaning, mainly: Agent-i trusts agent-j about the claim p iff Agent-i believes that he trusts agent-j about p. Hence trust does not contradict one's beliefs. These are the only (very natural) restrictions we have on the interrelation between trust and belief.

3.2 Semantics

Kripke semantics is provided by bi-relational Kripke frames together with two neighborhood functions. More formally

Definition 3.21 *A $\mathbf{B}^2_T$-frame $\mathfrak{F}$ is a tuple $(W, R_1, R_2, u_{1,2}, u_{2,1})$, where:*

W is a set of possible worlds,

$R_1, R_2 \subseteq W \times W$ are weakly transitive and symmetric relations,

$u_1 : W \to PP(W), u_2 : W \to PP(W)$ are functions (neighborhood maps), such that the following equalities take place: $u_{i,j}(w) = \bigcap_{v \in R_i(w)} u_{i,j}(v)$, for every $i, j \in \{1, 2\}$ where $i \neq j$.

$\mathbf{B}^2_T$-model is a pair $M = (\mathfrak{F}, V)$, where $\mathfrak{F}$ is a $\mathbf{B}^2_T$-frame and $V : Prop \to P(W)$ is a valuation.

Definition 3.22 *A satisfaction of a formula in a given $\mathbf{B}^2_T$-model $M = (\mathfrak{F}, V)$ and a point $w \in W$ is defined inductively as follows:*
$w \Vdash p$ *iff* $w \in V(p)$,
$w \Vdash \neg\alpha$ *iff* $w \nVdash \alpha$,
$w \Vdash \alpha \wedge \beta$ *iff* $w \Vdash \alpha$ *and* $w \Vdash \beta$,
$w \Vdash \Box_i\alpha$ *iff* $(\forall w')(wR_iw' \Rightarrow w' \Vdash \alpha)$,
$w \Vdash T_{i,j}\alpha$ *iff* $|\alpha| \in u_{i,j}(w)$. *Here* $|\alpha|$ *denotes the set* $\{v | v \Vdash \phi\}$.

A formula is valid in in a given $\mathbf{B}^2_T$-model if it is true at every point of the model. A formula is valid in a $\mathbf{B}^2_T$-frame if it is valid in every model based on the frame. A formula is valid in a class of $\mathbf{B}^2_T$-frames if it is valid in every frame in the class.

Theorem 3.23 *The logic $\mathbf{B}^2_T$ is sound and complete with respect to the class of all $\mathbf{B}^2_T$-frames.*

Proof. The soundness easily follows by direct check as for completeness, the proof is standard and therefore we just give a sketch.

Let W be the set of all maximally consistent subsets of formulas in a logic $\mathbf{B}^2_T$. Let us define the relations R_1 and R_2 on W in the following way: For every

$\Gamma, \Gamma' \in W$ we define $\Gamma R_i \Gamma'$ iff $(\forall \alpha)(\Box_i \alpha \in \Gamma \Rightarrow \alpha \in \Gamma')$, where $i \in \{1, 2\}$. The following lemma is proved in [4] when proving completeness of the logic KS. It also directly follows from Sahlqvist theorem and the observation that KS-axioms characterize the class of all weakly-transitive and symmetric frames.

Lemma 1. *[4] Each R_i is weakly-transitive and symmetric.*

So far we defined a set W with two weakly-transitive and symmetric relations R_1, R_2 on it. Now we define functions $u_{1,2}$ and $u_{2,1}$ in the following way:

$$u_{i,j}(\Gamma) = \{\{\Gamma' | \phi \in \Gamma'\} | T_{i,j}\phi \in \Gamma\}.$$

It immediately follows that $u_{i,j}$ are functions defined from W to $PP(W)$. Before we show that $u_{i,j}(\Gamma) = \bigcap_{\Gamma' \in R_i(\Gamma)} u_{i,j}(\Gamma')$, for every $i, j \in \{1, 2\}$, where $i \neq j$ and every $\Gamma \in W$, let us define the valuation and prove the truth lemma.

The valuation V is defined in the following way: $V(\Gamma) = \{p | p \in \Gamma\}$.

Lemma 2 (*Truth*). *For every formula $\alpha \in \mathbf{B}^2_T$ and every point $\Gamma \in W$ of the canonical model, the following equivalence holds: $\Gamma \Vdash \alpha$ iff $\alpha \in \Gamma$.*

Proof. The proof goes by induction on the length of formula. Base case follows immediately from the definition of valuation. Assume for all $\alpha \in \mathbf{B}^2_T$ with length less then k holds: $\Gamma \Vdash \alpha$ iff $\alpha \in \Gamma$.

Let us prove the claim for α with length equal to k. If α is conjunction or negation of two formulas then the result easily follows from the definition of satisfaction relation and the properties of maximal consistent sets, so we skip the proofs. Assume $\alpha = \Box_i \beta$ and assume $\Gamma \Vdash \alpha$. Take a set $B = \{\gamma | \Box_i \gamma \in \Gamma\} \cup \{\neg\beta\}$. The sub claim is that B is inconsistent. Assume not, then there exists $\Gamma' \in W$ such that $\Gamma' \supseteq B$. This by definition of the relation R_i means that $\Gamma R_i \Gamma'$. Now as $\neg\beta \in \Gamma'$, by inductive assumption we get $\Gamma' \Vdash \neg\beta$. Hence we get a contradiction with our assumption that $\Gamma \Vdash \Box_i \beta$. So B is inconsistent. This means that there exists $\gamma_1, \gamma_2, ...\gamma_n \in B$ such that $\vdash \gamma_1 \wedge \gamma_2 \wedge .. \wedge \gamma_n \rightarrow \beta$. Applying necessitation rule for $\Box_i$ we get $\vdash \Box_i \gamma_1 \wedge .. \wedge \Box_i \gamma_n \rightarrow \Box_i \beta$ so $\Gamma \vdash \Box_i \beta$, hence we get that $\Box_i \beta \in \Gamma$.

We just showed the left-to-right direction of our claim for $\alpha = \Box_i \beta$. For the right-to-left implication assume that $\Box_i \beta \in \Gamma$. By definition of R_i for every Γ' with $\Gamma R_i \Gamma'$ we have $\beta \in \Gamma'$. From this by inductive assumption it follows that $\Gamma' \vdash \beta$. So we imply that $\Gamma \Vdash \Box_i \beta$.

Now assume $\alpha = T_{i,j}\phi$. Assume $\Gamma \Vdash T_{i,j}\phi$. By definition this means that $|\phi| \in u_{i,j}(\Gamma)$. Hence there exists β such that $\{\Gamma'' | \beta \in \Gamma''\} = |\phi|$ with $T_{i,j}\beta \in \Gamma$. This means that we have $\vdash \beta \leftrightarrow \phi$ in $\mathbf{B}^2_T$. Hence by the rule for trust modality we have $\vdash T_{i,j}\beta \leftrightarrow T_{i,j}\phi$. But the last implies that $T_{i,j}\phi \in \Gamma$.

Conversely assume that $T_{i,j}\phi \in \Gamma$ this implies that $\{\Gamma'' | \phi \in \Gamma''\} \in u_{i,j}(\Gamma)$. Now by inductive assumption we know that $\Gamma'' \Vdash \phi$ iff $\phi \in \Gamma''$ hence $|\phi| \in u_{i,j}(\Gamma)$. Hence $\Gamma \Vdash T_{i,j}\phi$.

Now let us show that the model we constructed falls into the class of $\mathbf{B}^2_T$-models. The only thing left to show is the following equality:

$$u_{i,j}(\Gamma) = \bigcap_{\Gamma' \in R_i(\Gamma)} u_{i,j}(\Gamma').$$

Assume $X \in u_{i,j}(\Gamma)$. This means that X is of the form $\{\Gamma'' | \phi \in \Gamma''\}$ for some ϕ with $T_{i,j}\phi \in \Gamma$. Because of the $\mathbf{B}^2_T$ axioms and because Γ is maximally consistent set, we imply that $\Box_i T_{i,j}\phi \in \Gamma$. From this we imply that $T_{i,j}\phi \in \Gamma'$ for every $\Gamma' \in R_i(\Gamma)$. Now by definition of $u_{i,j}$ this means that $\{\Gamma'' | \phi \in \Gamma''\} \in u_{i,j}(\Gamma')$ and as Γ' was arbitrary member of $R_i(\Gamma)$, we get that $X = \{\Gamma'' | \phi \in \Gamma''\} \in \bigcap_{\Gamma' \in R_i(\Gamma)} u_{i,j}(\Gamma')$.

Conversely assume some set $X \subseteq W$ belongs to $\bigcap_{\Gamma' \in R_i(\Gamma)} u_{i,j}(\Gamma')$. By definition this means that there exists a formula ϕ such that $T_{i,j}\phi \in \bigcap_{\Gamma' \in R_i(\Gamma)} \Gamma'$ and $X = \{\Gamma'' | \phi \in \Gamma''\}$. Now as far as $(\forall \Gamma')(\Gamma R_i \Gamma' \Rightarrow T_{i,j}\phi \in \Gamma')$ by truth lemma we get that $(\forall \Gamma')(\Gamma R_i \Gamma' \Rightarrow \Gamma' \Vdash T_{i,j}\phi)$. Hence $\Gamma \Vdash \Box_i T_{i,j}\phi$. Now applying axioms for trust modality we get that $\Gamma \Vdash T_{i,j}\phi$ and hence $X \in u_{i,j}(\Gamma)$. This completes the proof.

4 Conclusions

As a conclusion we mention that the logic described is very much alike to the fragment of BIT defined in [2] and techniques used are also much similar. The only motivation for considering $\mathbf{B}^2_T$ and hence moving from "$KD45$-reasoner" to "KS-reasoner" lies in the future perspective to generalize the semantics of the logic and impose topology instead of neighborhood maps. It is well known that KS has much closer connection with topology then $KD45$. As a future work we use the system PρLog [3] to implement a reasoner for reasoning questions in the $\mathbf{B}^2_T$.

5 Acknowledgments

This research has been partially funded by LIACC through Programa de Financiamento Plurianual of the Fundao para a Cincia e Tecnologia (FCT), by the FCT fellowship (ref. SFRH/BD/62058/2009), by the MCICINN projects TIN2006-15455-CO3 and CSD2007-00022.

References

1. P. Blackburn, J.F.A.K. van Benthem, Frank Wolter. *Handbook of Modal Logic.* 2002.
2. Churn-Jung Liau. *Belief, information acquisition, and trust in multi-agent systems - A modal logic formulation.* Journal of Artificial Intelligence, 149, 3160, 2003.

3. J. Coelho, B. Dundua, M. Florido, and T. Kutsia. *A rule-based approach to xml processing and web reasoning.* In P. Hitzler and T. Lukasiewicz, editors, RR, volume 6333 of Lecture Notes in Computer Science, pages 164172. Springer, 2010.
4. L. Esakia. *Weak transitivity-restitution.* Logical Studies 2001, vol 8, 244-255.
5. V. Goranko. *Modal definability in enriched language.* Notre Dame Journal of Formal Logic, 31, 81-105, 1990.
6. P.R. Halmos. *Algebraic Logic*. Chelsea Publishing Company, New York, 1962.
7. M. de Rijke. *The Modal Logic of Inequality.* J. Symbolic Logic, 57, 566-587, 1992.
8. K. Segerberg. *"Somewhere else" and "some other time". In: Wright and wrong: mini-essay in honor of G.H.Wright*, Publ. the group in logic and methodology of Real Finland, 61-64, 1976.
9. R. M. Smullyan. *What Is the Name of This Book?: The Riddle of Dracula and Other Logical Puzzles.*

Behavioral Similarity of Semantic Web Services*

Zijie Cong and Alberto Fernández

CETINIA, Universidad Rey Juan Carlos, Madrid, Spain
zijie@ia.urjc.es, alberto.fernandez@urjc.es

Abstract. Service matchmaking is an integral link of service discovery, composition, invocation and other similar task under Service-Oriented Architecture (SOA). Most current approaches measure the degree of match of two service based merely on their I/O pairs which could leads to false result. This paper presents an approach for matchmaking in Semantic Web Services (SWS) that considers each service as a sub-graph of semantic network, which is formed by inputs, outputs, pre-conditions and post-condition, with contribution of syntactical information such as keywords from the service description. Thus the similarity between services is defined as the similarity between two sub-graphs. The aim of this approach is to reveal the internal work flow and intention of services, i.e. behaviors, thus it agrees with human intuition to a larger extent than previous approaches.

1 Introducction

The original intention of adding semantic annotations to web services is to improve the automation of services discovery, selection, invocation and interoperation by letting service descriptions to be machine-processable [7]. One integral part of such automation is matchmaking among services.

Various approaches have been proposed in previous studies. Without concerns about semantics of its components, one primitive method to calculate the similarity of services is based on the syntactical information - e.g. keywords, tag-clouds and textual descriptions.

For services with semantic information, inputs/outputs (I/O) matching is a common method for measuring the similarity. Inputs and outputs of a semantic service are instances of ontological concepts, the similarity of two services is determined by the minimal distance in the taxonomy tree between corresponding concepts of I/O pair, the result is a degree of semantic simialrity, such as exact, plug-in and subsumes [6]. Some studies, such as [5], aimed to achieve higher robustness and precision by combining both.

More recently, various graph based approaches have been proposed. In [4], a service was considered as a composition of processes and thus could be represented as a finite-state machines (FSMs), the similarity between services was

* Work partially supported by the Spanish Ministry of Science and Innovation through grants TIN2009-13839-C03-02 and CSD2007-0022(CONSOLIDER-INGENIO 2010)

defined as the similarity between two FSMs. Like other similar graph-based approaches [[3,2],[3,2]], it concentrated on structural similarity of services instead of the semantic similarity of atomic units of functionality.

This paper presents a novel but preliminary approach for service matchmaking. The main notion behind this approach is that a service could be considered as a sub-graph of a semantic network, which maps input concepts to output concepts via elements specified in preconditions, post-conditions or retrieved from textual description, it reveals the behavior of service that could be a more intuitive option for calculating the degree of match of services.

2 Motivation

Although an appropriate measurement of degree of match is difficult to define, it is consensus that the result of matching should agree with human intuition. Inputs and outputs sometimes may not provide sufficient information about service's behavior, and replying solely on them may lead to false result. An example is presented in the rest of this section.

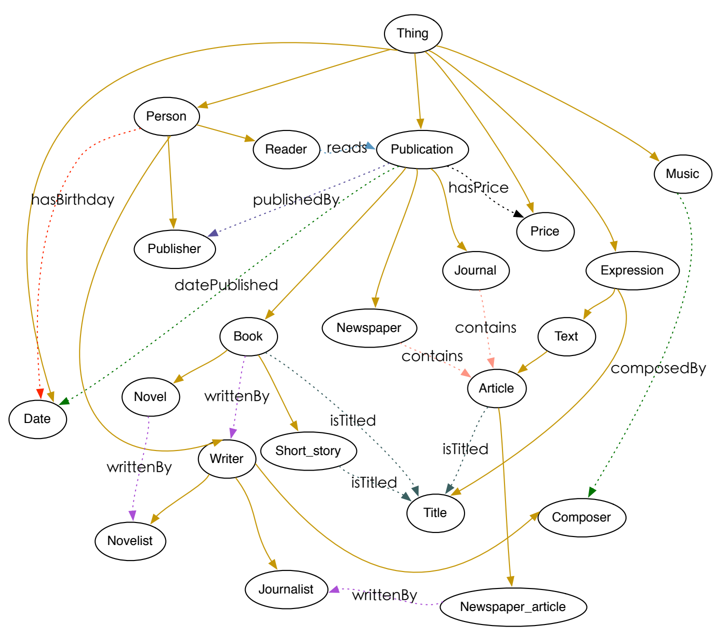


Fig. 1. An ontology of publications with 22 concepts and 10 relations, brown solid lines represent subsumption relations.

Figure 1 [1] illustrates an ontology of publication1 with 22 concepts and 10 relations connecting them. Table 1 presents three services using this ontology. Every service description used in this paper is a quintuple (T; I;O; P;Q), where:

- T is the syntactical information of service, such as description, key-words, etc.
- I is a set of input concepts.
- O is a set of output concepts.
- P is a set of predicates that must be true prior to the invocation of the service, i.e. preconditions. These predicates are relations between concepts defined in ontology, such as writtenBy(Book, Writer).
- Q is a set of predicates that must be true after the execution of the service, i.e. post-conditions. Same as for P, these predicates are relations defined in ontology as well.

$$S_1 = \begin{cases} T & returns\ the\ birthday\ of\ a\ given\ novelist \\ I & \{Novelist,\ Novel\} \\ O & \{Date\} \\ P & writtenBy(Novel,\ Novelist) \\ Q & hasBirthday(Novelist,\ Date) \end{cases}$$

$$S_2 = \begin{cases} T & the\ date\ of\ publish\ of\ a\ writer's\ first\ book \\ I & \{Writer,\ Book\} \\ O & \{Date\} \\ P & writtenBy(Book,\ Writer) \\ Q & datePublished(Book,\ Date) \end{cases}$$

$$S_3 = \begin{cases} T & published\ date\ of\ a\ novelist's\ premier\ work \\ I & \{Novelist,\ Novel\} \\ O & \{Date\} \\ P & writtenBy(Novel,\ Novelist) \\ Q & \emptyset \end{cases}$$

Fig. 2. Services using ontology of publication

By using I/O matching approaches, matchmaker will not be able to distinguish between S_1 and S_3 as their inputs and outputs are identical, thus these two services matches exactly, even though the functionality of these two services are different. On the other hand, S_2 would give a lower degree of match against S_3 despite they are more similar behaviorally.

Therefore the aim of our approach is to overcome above limitations by exploiting the behavioral information of services.

[1] This ontology is partially adopted from "books.owl" of OWL-TC3

3 Service Behavioral Graph (SBG)

To exploit the behavioral information of a service, we consider a service as a function that maps its inputs to outputs. In SWS, inputs and outputs are ontological concepts, this mapping is defined by relations in the same domain ontology. As an ontology can be represented by *multi-relational graph* where each vertex denotes a concept and each edge denotes a relation between concepts, a service thus can be further considered as a sub-graph of an ontology. More formally,

Definition 1. *if* $G = (V \times E \subseteq (V \times V))$ *where* V *is the set of concepts and* E *is the set of relations of heterogeneous types, is a multi-relational graph of an ontology, then service* S *is denoted as* $G_S = \left(V^{'}, E^{'}\right)$ *where* $V^{'} \subseteq V$ *and* $E^{'} \subseteq (V^{'} \times V^{'}) \subseteq E^{'}$.

This graph is referred as *service behavioral graph* (SBG) in this paper, it can be discovered from the ontology graph using critical elements and behaviorally correct path defined in the following sections. Algorithm 1 is the pseudo-code for SBG discovery[2], details can be found in Section 3.3.

An example can be seen in Figure 3, the elements in blue of the graph depict the SBG of S_1 defined in Figure 2.

3.1 Critical Elements

We have mentioned in the beginning of this section that the mapping from inputs to outputs is defined by the relations in the domain ontology, this mapping is, in fact, a set of paths from input concepts to output concepts, consisting one or more relations. There may exist multiple paths between a pair I/O concepts, therefore, finding proper paths is critical for describing the the service's behavior correctly.

Such paths is determined by several components in the ontology, which can be concepts or relations, they are referred as *critical elements* in this paper. P(*Precondition*), Q(Post-condition) and *T(Syntactical information)* of service description may offer some clues to these critical elements.

Syntactical Information Syntactical information is valuable for revealing services' behaviors. For example, even though $S_{1(I,O)} = S_{3(I,O)}$, the textual descriptions *(T)* differ these two services at human-readable level. To find the critical elements, syntactical information and ontological components' identifiers (besides those in I/O sets) need to processed using Information Retrieval techniques to transform into a set of keywords with irrelevant words and morphological variants removed. Then components with keywords appeared in the syntactical information of the service is considered to be a critical element. For example, in S_2, relation *datePublish,* concepts *Date,*

[2] We do not concentrate on finding the shortest behaviorally correct path in this paper, various existing approaches on shortest path problem can be adopted with minimal effort.

Publication, Writer and *Book* are identified as a critical elements as the word "publish", "date", "writer" and "book" have appeared in $S_{2(T)}$.

Preconditions The preconditions is a set of predicates that must be true before the service can be invoked. It is not concerned with the behavior of the service, but indicate the relations among input elements. An *inputs* sub-graph of service behavioral graph can be formed. For example, in S_2, $(Book, writtenBy, Writer)$ is the *inputs* sub-graph of service behavioral graph.

Post-conditions The post-conditions is a set of predicates that must be true after the execution of the service. These predicates often connect input elements with output elements, hence reveal valuable information.

3.2 Behaviorally Correct Path (BCP)

To connect inputs with outputs, a path containing critical elements defined in the previous section needs to be find, we refer this path as a *behaviorally correct path (BCP).*

In semantic networks, concepts are usually connected by heterogeneous links, including hierarchical relations as well as other relations. There may exist multiple paths with same length (in term of number of edges) from on concept to another. For similarity measuring purpose, it is necessary to have unique path between two elements, and such path should not only contain the critical elements, but also be semantically correct.

In [1], Aleksovski et al. considered a path to be semantically correct if and only if no hierarchical links appear after a non-hierarchical one. For example, in Figure 1, a path *{ShortStory, is_ a, Book, writtenBy, Writer}* is semantically correct, while *{ShortStory, is_ a, Book, writtenBy, Writer, is_ a, Person}* is not.

In practice, however, there is a high possibility that no semantically correct path exists between two concept using Aleksovki's definition. Therefore, for the purpose of this paper, we define a behaviorally correct path as:

Definition 2. A behaviorally correct path is a path in semantic network between two concepts containing critical elements with maximum one turn from non-hierarchical relation to hierarchical relation.

And two assumptions must be hold to ensure the existence of a BCP:

1. Any relation in an ontology is at least partially symmetric.

This assumption implies that if a relation exists between two concepts, then there also exists an inverse relation between same two concepts which is at least partially symmetric.

2. All relations are inheritable from a super-concept to a sub-concept.

This assumption implies that if there exists a relation p between concepts x and y, i.e. *p(x, y)*, and *is_ a(z, x)*, then *p(z, y)*. This eliminates sequence of subsumption relation that might be appeared in the beginning of a BCP and also reduces the length of BCP.

Together, Definition 2, Assumption 1 and 2, ensure that there always exist a behaviorally correct path between two concepts.

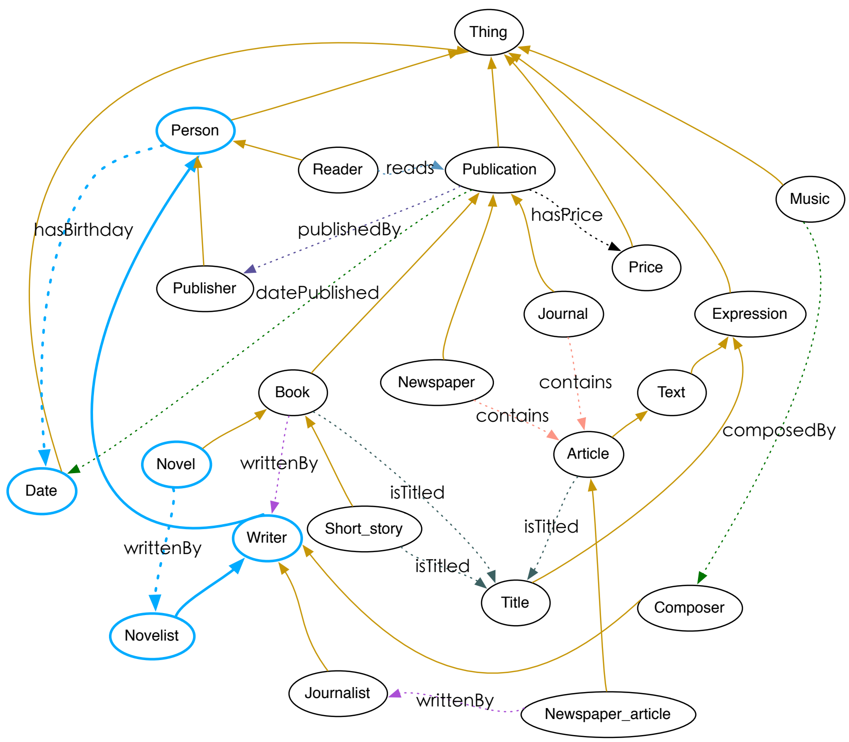


Fig. 3. SBG of service that returns a novelist's birthday in blue

3.3 Algorithm

Algorithm 1 presents how a SGB is discovered. In line 3, inputGraph denotes a graph formed by input concepts and preconditions. CriticalElements in line 4 determines the critical elements described in Section 3.1, this procedure can be implemented using IR techniques. Variable P at line 6 denotes a set of paths from a input concept to output concept via various relations, these concepts and relations are elements of the set of critical elements CES. The final result SBG is a union of all shortest paths connecting critical elements and inputGraph.

4 Calculating Similarity

The similarity of services is defined as the similarity of their corresponding SBGs. As sub-graphs of a semantic network, SBGs are multi-relational graphs, which can be represented by binary 3-way tensor.

A tensor is an object that extends the notion of scalar, vector and matrix to higher orders. A single-relational graph has representation of a adjacent matrix

Algorithm 1 Algorithm for SBG discovery.

procedure SBG(S) ▷ S is the service description quintuple
    $SBG \leftarrow \emptyset$
    $inputGraph = (S_I, S_P)$
    $CES \leftarrow$ CriticalElements(S_P, S_T, S_Q) ▷ critical elements
    for $\forall i \in CES \cap S_I,\ R = CES \cap (S_P \cup S_Q),\ o \in CES \cap S_O$ **do**
        $P \leftarrow \text{BCP}(i, r, o)$ ▷ set of behaviroally correct paths
        $SBG \leftarrow (SBG \cup (\underset{p \in P}{argmin}(\mid p \mid))$ ▷ shortest path
    end for
    return $SBG \cup InputGraph(S)$
end procedure

that can be seen as a 2-way tensor, if we consider a multi-relational graph as a union of multiple single-relational graphs, it thus can be represented using a 3-way tensor. A tensor representation of an ontology with n concepts and m relations is

$$A \subseteq \{0,1\}^{n \times n \times m}$$

where

$$A^k_{i,j} = \begin{cases} 1 & if\ (i,j) \subseteq E_k, k < m \\ 0 & otherwise \end{cases}$$

Figure 4 illustrates such tensor in a visualized form.

An intuitive approach for calculating the similarity between two tensors is subtraction, as two services share same ontology, the order of tensors of services are equal. The result of subtraction will be a tensor of symmetric difference of two SBGs. Under service discovery scenario, a service request is compared against a candidate service advertisement:

$$SBGDiff(S^R, S^A) = SBG(S^R) - SBG(S^A) \quad (1)$$

$$inputGraphDiff(S^R, S^A) = InputGraph(S^R) - InputGraph(S^A) \quad (2)$$

where S^R and S^A are service request and service advertisement tuples respectively. Since the SBGs and InputGraphs are binary three-way tensor, the result tensor is $R = (-1, 1, 0)^{n \times n \times m}$ where each -1 indicates an element appear in service advertisement only, 0 indicates an common elements and 1 indicates an element appear in service request only.

We further define six degrees of match based on this resulting tensor. Let D denotes the symmetric difference tensor between SBGs of service request S^R and service advertisement S^A; G denotes the symmetric difference tensor of InputGraphs. The degrees of match of a service request and service advertisement are,

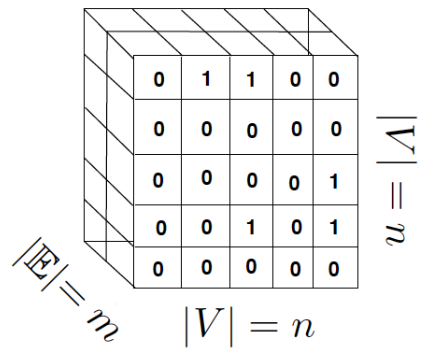


Fig. 4. 3rd-order tensor representation of an ontology with n concepts and m relations

Exact S^R EXACTLY matches $S^A \iff \forall d \in D : d = 0$. This degree indicates that two services match exactly at both behavioral and structural level.

Tail-match S^R is TAIL-MATCH with $S^A \iff \forall d \in D : d \geq 0 \wedge d \in G$. This degree indicates that service request provides extra excessive inputs but matches with service advertisement's behaviors and outputs.

Head-Match S^R is HEAD-ALIGNED with $S^A \iff \forall g \in G : g = 0 \wedge \forall d \in D : d \leq 0 \wedge \forall o_r \in S^R_O \exists o_a \in S^A_O : o_r > o_a$. This degree indicates that service request requires only a subset of advertisement's outputs but matches its behaviors and inputs.

SubgraphOf S^R is a sub-graph OF $S^A \iff \forall d \in D : d \leq 0$. This degree indicates that the SBG of service request is a sub-graph of service advertisement, which cannot be invoked directly but might be padded through service composition.

Subsumes S^R SUBSUMES $S^A \iff \forall d \in D : d \geq 1$. This degree implies that the service advertisement is a subgraph of request, invocation might be done after service composition.

Fail S^R does not matches S^A.

The service advertisements that match with service request on first three degrees are invokable while the following two degrees, subgraphof and subsumes, require extra units of functionality,i.e. services, to be participated.

5 Conclusion and Future work

This paper presents a novel but preliminary approach of calculating the similarity between two services. This approach intends to reveal the behavioral information of services, and by comparing their similarity to achieve higher accuracy, robustness and in agreement with human intuition. The main notion behind this approach is that we consider a service as a sub-graph of semantic network that connects its inputs concepts and output concepts via critical elements, referred as Service Behavioral Graph (SBG). We use syntactical information and conditions to determine the critical elements, and a SBG is discovered by exploiting these elements. The similarity of services is defined as six degrees of match in this paper based on the differences between two SBGs. In practise, services do not often belong to the same ontology, alignment needs to be performed in case of multiple ontologies are participated in matchmaking.

Experiments with actual realistic test cases are necessary to access the practicability of our approach. One expectable limitation of our approach is that it depends on the quality (in term of richness) of the ontology to a large extent, which is highly unstable in practise. Our future work includes implementation, experiments and evaluation of this approach, also solving open issues such as diminish the deviation caused by the instability of the quality of ontologies and refine the degree of match.

References

1. Z. Aleksovski, W. ten Kate, and F. van Harmelen. Exploiting the structure of background knowledge used in ontology matching. In *Ontology Matching Workshop at International Semantic Web Conference (ISWC)*. Citeseer, 2006.
2. J. Corrales, D. Grigori, and M. Bouzeghoub. Bpel processes matchmaking for service discovery. *On the Move to Meaningful Internet Systems 2006: CoopIS, DOA, GADA, and ODBASE*, pages 237–254, 2006.
3. D. Grigori, J.C. Corrales, and M. Bouzeghoub. Behavioral matchmaking for service retrieval: Application to conversation protocols. *Information Systems*, 33(7-8):681–698, 2008.
4. A. Günay and P. Yolum. Structural and semantic similarity metrics for web service matchmaking. In *Proceedings of the 8th international conference on E-commerce and web technologies*, pages 129–138. Springer-Verlag, 2007.
5. M. Klusch, B. Fries, M. Khalid, and K. Sycara. Owls-mx: Hybrid owl-s service matchmaking. In *Proceedings of 1st Intl. AAAI Fall Symposium on Agents and the Semantic Web*, volume 142, 2005.
6. M. Paolucci, T. Kawamura, T. Payne, and K. Sycara. Semantic matching of web services capabilities. *The Semantic Web (ISWC 2002)*, pages 333–347, 2002.
7. K. Sivashanmugam, K. Verma, A. Sheth, and J. Miller. Adding semantics to web services standards. In *Proceedings of the International Conference on Web Services*, pages 395–401. Citeseer, 2003.

An Ontology for Formalising Agreement Patterns in Auction Markets

José J. Durán[1] and Carlos A. Iglesias[2]

[1] Centro para las Tecnologías Inteligentes de la Información y sus Aplicaciones (CETINIA), Universidad Rey Juan Carlos (Spain)
[2] Depto. Ingeniería de Sistemas Telemáticos, Universidad Politécnica de Madrid (Spain)

Abstract. Knowledge and best practices on auction systems are currently disseminated across the research literature, which limits its access, reuse, evaluation and feedback by practitioners. This article presents a systematic approach to collect this knowledge as design patterns, in order to provide assistance to software developers. An ontology has been defined for formalising design patterns in auction systems, with the aim of improving its searchability by software developers. Finally, a case study illustrates how the proposed pattern ontology provides assistance in the development of a dynamic pricing model for an e-commerce service.

1 Introduction

Auctions provide a market system model that enable the exchange of resources on the basis of supply and demand. They have proved to be an effective model for dynamic pricing of resources in different scenarios such as electronic commerce [1], resource allocation [2], service pricing [3] or sponsored search pricing [4].

Nevertheless, software engineers have few available resources that provide them support in the design of auction mechanisms and automatic bidders, since current knowledge and best practices on auctions are disseminated across research publications. It is a good practice to identify the elements of good and reusable designs in auctions, and provide a systematic framework for formalising the experience with these designs. This is precisely the role of *design patterns* [5], which describe general reusable solutions to commonly recurring problems in software design. The notion of design patterns was originated in the object-oriented software engineering community and has been widely accepted by this community, having a strong impact on how object-oriented software is designed, implemented and communicated nowadays.

The purpose of this article is to provide a structured and formalised schema for describing design patterns in the field of agreement technologies and validate it through its application in the domain of auctions.

The rest of the article is organised as follows. Section 2 presents an ontology for describing agreement patterns, in order to provide standard facilities for retrieving patterns based on the user requirements. This agreement patterns ontology has been linked with an auction domain ontology for describing the main concepts of the auction domain, in order to provide a common language for describing the auction patterns. In order to show the applicability of our approach, a case study is developed within section 3. Finally, section 4 summarises the main contributions of this paper and the future research activities.

2 Modeling an Ontology for Auction Patterns

Even though design patterns are usually expressed in natural language, several works have proposed its formalisation in order to provide tool support for consulting a pattern catalogue and guide developers in its application. Ontologies [6] have been considered as a natural formalisation technique that enables an infrastructure for sharing and interconnecting semantically pattern languages in the web. Several works have used ontologies [7] for formalising both the structure of the patterns [8,9] (how to apply the pattern) as well as its intention (when to apply the pattern) [7,10].

This article presents an ontology for formalising the intention of agreement patterns in order to improve its findability. The ontology is organised into three levels as shown in fig. 1 which are detailed below. The first level defines an ontology (*APO*) for agreement patterns (section 2.1), with the aim of facilitating the searchability of the patterns by software developers. A domain ontology (*AUTERMS*) for the auction domain (section 2.2) provides the common terminology for describing auction patterns. Finally, an ontology (*AUPA*) has been defined for describing the auction patterns (section 2.3) . This ontology extends the *APO* ontology and is described using the *AUTERMS* ontology.

<table>
<tr><td colspan="2">AUPA
Auction Pattern Ontology</td></tr>
<tr><td>APO
Agreement Pattern Ontology</td><td>AUTERMS
Auction Terms Ontology</td></tr>
</table>

Fig. 1. Layers of the Auction Pattern Ontology

2.1 The Agreement Pattern Ontology (*APO*)

Agreement patterns [11] provide a way to collect best practices for reaching agreements in a structured way. In this section, the Agreement Patterns Ontology (*APO*) is introduced in order to catalogue agreement patterns. This catalogue should support developers in choosing a pattern for a given problem.

In order to determine the scope of the ontology, the ontology should be able to answer the following *competency questions* [12]:

- Which design patterns can solve a given problem?
- Which design patterns can solve a given problem and are applicable in a given context?
- Which design patterns are related to a given domain concept?
- Which design patterns are available for a given task?

The proposed ontology is based on the structure of the DPIO ontology[7]. That ontology describes the relationship between patterns and problem types, which are described by problem constraints. We have extended DPIO by focusing on pattern constraints, instead of problem type constraints.

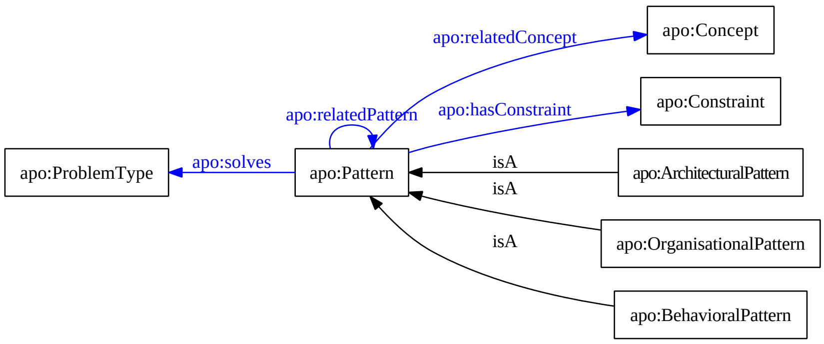


Fig. 2. Ontology for agreement patterns

The structure of the APO ontology is depicted in fig. 2. The main relationships of the core ontology are described below. A *Pattern* can solve one or more *ProblemTypes* and is applicable only if some *Constraints* are fulfilled. A *Pattern* can be related with other *Patterns*. Domain *Concepts* can be related with one or more *ProblemTypes* and one or more *Patterns*. *Patterns* are organised according to the task they solve into:

- *Organisational Patterns.* These patterns collect social structure patterns which define the norms, social and interaction model [13] which form the society as a whole, and which determine, to some varying degree, the actions of the individuals socialised into that structure.
- *Behavioural Patterns.* These patterns collect individual behaviours of the participants in the agreement in order to satisfy a goal.
- *Architectural Patterns.* These patterns describe architectural patterns describing the software architecture of the participants in the agreement.

2.2 The Auction terms Ontology (*AUTERMS*)

The objective of the Auction Terms Ontology (*AUTERMS*) is modelling the auction domain in order to provide a common vocabulary for describing auction patterns. Previous work have also proposed ontologies in the negotiation domain [14,15,16], and have been extended to the auction domain, but we have not found a specific ontology in the auction domain. We have modelled a basic ontology based mainly on the auction patterns proposed by Ré [17], the research survey on auctions by Parsons et al. [1] and the ontology for the agent trading competition developed by Wellman [18]. The core structure of the proposed ontology AUPA is shown in fig. 3.

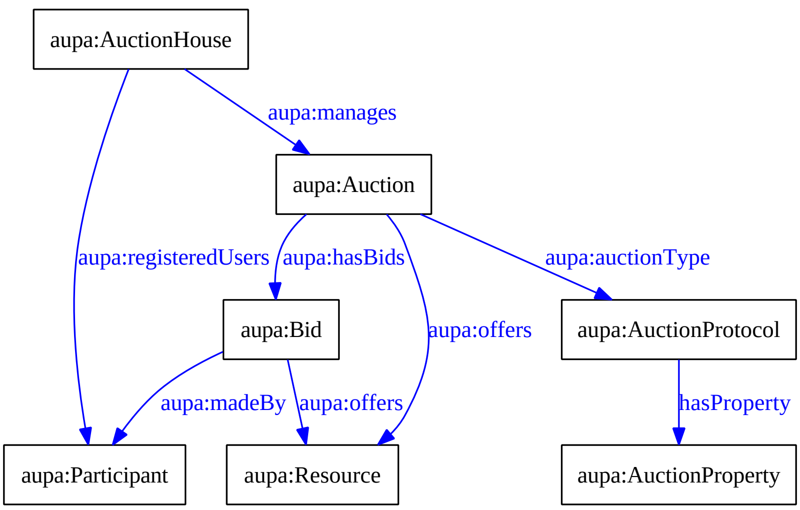


Fig. 3. Auction Ontology

The main concepts of this ontology are AuctionHouse, Auction, Participant, AuctionProtocol and AuctionProperty. *Auctions* are a negotiation process in which different *Participants* exchange information in the form of *Offers* in order to obtain a *Resource*. Those *Auctions* are placed in different *AuctionHouses* which represents the auctioneer of an *Auction*. An *AuctionHouse* has the responsibility of sharing *Auction* information between different *Participants* following the rules of a specific *AuctionProtocol*. An *AuctionProtocol* represents which *AuctionProperties* must be meet in a specific *Auction*. An *AuctionProperty* helps to classify an *AuctionProtocol* in order to find the most suitable based on the problem constraints.

2.3 Auction pattern Ontology (*AUPA*)

Previous works have also proposed the usage of design patterns in agreement technologies and auctions. Iglesias et al. [11] propose *Agreement Patterns* for formalising recurring solutions to agreement problems. Ré et. al. [17] propose auction patterns from an object oriented perspective. In the field of multiagent systms, Oluyomi [19] proposes a classification scheme for agent oriented patterns which is applied to a relevant number of agent patterns. Jureta et al. [20] describe three agent oriented patterns in the auction domain.

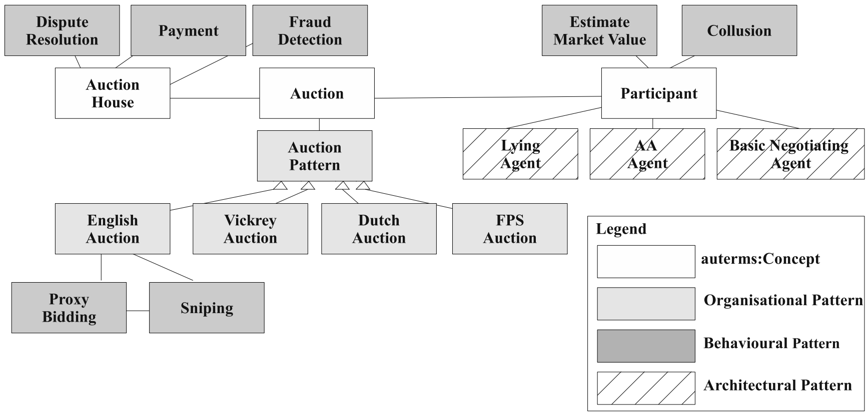


Fig. 4. Auction Pattern Catalogue

This article extends the work developed by Iglesias et al. [11] in order to organise, identify and describe semantically auction patterns as a specific family of agreement patterns. In order to validate our approach, several patterns have been formalised within this ontology. Fig. 4 shows these patterns grouped by the related Concept in the domain on *AUTERMS* ontology. In addition, the auction patterns are classified in Architectural, Organisational and Behavioural according to the *APO* Ontology.

The pattern catalogue includes the following patterns:

- *Organisational:* AuctionPattern, English Auction, Dutch Auction, Vickrey Auction, First Price Sealed Auction
- *Architectural:* Lying Agent [21], AA Agent [22], Basic Negotiation Agent [19]
- *Behavioural:* Proxy Bidding [20], Sniping, Dispute Resolution [20], Payment [20], Fraud Detection [20], Colussion, Estimate Market Value.

In order to illustrate how auction patterns are described, the pattern *Vickrey Auction* is described in natural language in table 1, while its semantic description is depicted in fig. 5 and fig. 6. The ontology has been defined using the ontology editor Protégé [23]. Thanks to the semantic description, the catalogue can be consulted and filtered according to the user constraints, as described in section 3.

Name	Vickrey auction
Alias	Second-price sealed-bid auction.
Keywords	Auction, service pricing.
Problem Type	Resource assignation.
Problem	Auction of multiple similar resources, or items, in which participants should be encouraged to share their true valuation using incentives.
Context	There are different consumers willing to get the resources, and the real value of the resource is not known.
Solution	This auction is defined by the next properties, as seen on fig. 5 : – **One round:** Bids are received in a unique round, which last until all participants have made their offer, or until a specific time. – **Sealed:** Offers are not visible, in any mean, to participants, until the auction has finished. – **Multiple items per bid:** Offers should be for more than a resource, and associated with a price. Only an unique bid per participant. – **Second pricing:** the highest n bids are awarded the resource and pay a price equal to the n+1 highest amount bid. – **Incentive compatibility:** Each participant maximizes its expected utility, by revealing their true valuation, as final price is not dependant on the offer made, but in the last highest offer. Also, in scenarios with different auctioned resources, a preference assignation of resources, will be an incentive to bid as high as possible [24]. The most important advantage of this auction, is that sellers don't require to have knowledge about buyers willingness to pay. Because of that, this auction is highly suitable in scenarios without that information.
Examples	Mobile spectrum assignment[1], Internet advertising[25].
Related patterns	Sealed auction, Fake bidding.

Table 1: Vickrey Organisational Pattern in Natural Language

3 Case study

In order to illustrate the practical application of agreement patterns, this section describes a case study consisting of the development of an opera ticket selling service.

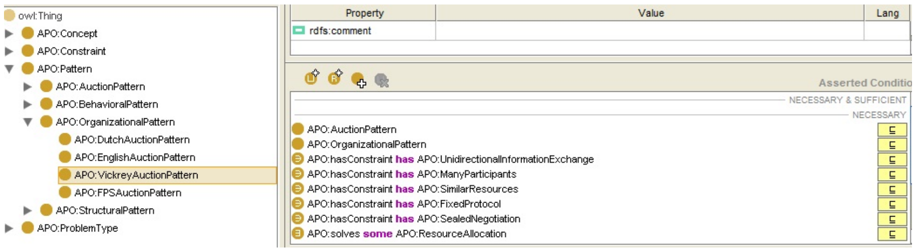


Fig. 5. Screenshot of Protégé of the semantic description of the Vickrey Pattern

```
<owl:Class rdf:ID="VickreyAuctionPattern">
  <rdfs:subClassOf rdf:resource="#AuctionPattern"/>
  <rdfs:subClassOf>
    <owl:Class rdf:about="#OrganisationalPattern"/>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty>
        <owl:ObjectProperty rdf:about="#hasConstraint"/>
      </owl:onProperty>
      <owl:hasValue rdf:resource="#ManyParticipants"/>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:hasValue rdf:resource="#DivisibleResources"/>
      <owl:onProperty>
        <owl:ObjectProperty rdf:about="#hasConstraint"/>
      </owl:onProperty>
    </owl:Restriction>
  </rdfs:subClassOf>

   // similar restrictions for #FixedProtocol, #SealedNegotiation,
   // and #UnidirectinalInformationExchange
```

Fig. 6. OWL class of Vickrey Auction Pattern

An opera thether desires to maximize its benefit from sold tickets, which are sold by different online enterntainment ticket selling services. Several strategies for offering this service are available, such as defining a fixed pricing policy based on position of the seat or dynamic pricing based on an auction protocol. The opera theather desires to explore this second alternative.

In order to evaluate the ontology, we are going to review the competency questions enumerated in section 2.1, translate these consults into formal questions in SPARQL, and evaluate the result set of the queries.

Which design patterns can solve a given problem?. In our case, we are interested in consulting all the organisational patterns that solve the problem *Resource Allocation*. Fig. 7 shows the corresponding SPARQL query. The result set contains all the Organisational Auction Patterns of the ontology. In a full version of the ontology we could obtain other patterns not related with auctions, such as bargaining.

```
SELECT DISTINCT ?pattern
WHERE {
  ?pattern rdfs:subClassOf apo:OrganisationalPattern.
  ?pattern rdfs:subClassOf apo:Pattern.
  ?pattern rdfs:subClassOf [
    rdf:type owl:Restriction;
    owl:someValuesFrom apo:ResourceAllocation;
    owl:onProperty apo:solves
  ].
}
```

Fig. 7. SPARQL query for organisational patterns for resource allocation problem

Which design patterns can solve a given problem and are applicable in a given context?. The results of the previous query can be filtered specifying constraints. In our domain, opera seats are auctioned. We can decide whether opera seats are divisible or indivisible. In our case, we can sell all the seats for each opera performances, but each seat can be sold at a different price, so we add the constraint *DivisibleResource*. Fig. 8 shows the corresponding SPARQL query, which would return the *Vickrey Auction Pattern*.

Which design patterns are related to a given domain concept?. In our domain, we could be interested in which design patterns are related with the *Auction House* concept as shown in Figre 9.

Which design patterns are available for a given task?. In case we desire to define the architecture of a bidder, we can query the *Architectural Patterns* related with the concept *Participant* (Fig. 10).

```
SELECT DISTINCT ?pattern
WHERE {
  ?pattern rdfs:subClassOf apo:Pattern.
  ?pattern rdfs:subClassOf apo:OrganisationalPattern.
  ?pattern rdfs:subClassOf [
    rdf:type owl:Restriction;
    owl:someValuesFrom apo:ResourceAllocation;
    owl:onProperty apo:solves
  ].
  ?pattern rdfs:subClassOf [
    rdf:type owl:Restriction;
    owl:hasValue apo:DivisibleResources;
    owl:onProperty apo:hasConstraint
  ].
}
```

Fig. 8. SPARQL query for organisational patterns of resource allocation of divisible resources

```
SELECT DISTINCT ?pattern
WHERE {
  ?pattern rdfs:subClassOf apo:Pattern.
  ?pattern rdfs:subClassOf [
    rdf:type owl:Restriction;
    owl:someValuesFrom auterms:auctionHouse;
    owl:onProperty apo:relatedDomainConcept
  ].
}
```

Fig. 9. SPARQL query for design patterns related with the Auction House Concept

```
SELECT DISTINCT ?pattern
WHERE {
  ?pattern rdfs:subClassOf apo:Pattern.
  ?pattern rdfs:subClassOf apo:ArchitecturalPattern.
  ?pattern rdfs:subClassOf [
  rdf:type owl:Restriction;
    owl:someValuesFrom auterms:participant;
    owl:onProperty apo:relatedDomainConcept
  ].
}
```

Fig. 10. SPARQL query for architectural patterns related with the concept Participant

4 Conclusions and future work

Design patterns can contribute to promote knowledge reuse and advance in the field of agreement technologies, since the expertise of practices is formalised and can be easily shared among practitioners, allowing them to share their experiences and understand better the advantages. limitations and applicability of these patterns.

In this paper, several auction patterns have been identified in the research literature and described and classified according to a pattern form. In addition, a domain ontology for auctions has been defined in order to provide automated reasoning on the application of patterns.

Current work is focused on several directions. First, our aim is progressing on the formalisation of the patterns themselves in order to provide and ontology-based design pattern repository as [26,27,7]. Second, since the targeted users of this research are software developers, our goal is providing at hand support during their development tasks, through the integration of this tool in a standard IDE such as Eclipse or Netbeans. In addition, this paper has presented an initial set of identified patterns, and our aim is enlarging this set, analysing available systems and documented practices as well as with the cooperation of other researchers and users.

Acknowledgements

This research has been partly funded by the Spanish Ministry of Science and Innovation through the projects Ingenio Consolider2010 AT *Agreement Technologies* (CSD2007-0022)), OVAMAH (TIN2009-13839-C03-02) and T2C2 (TIN2008-06739-C04-03/TSI) as well as the Spanish Ministry of Industry, Tourism and Trade through the project RESULTA (TSI-020301-2009-31).

References

1. Parsons, S., Rodríguez-Aguilar, J.A., Klein, M.: Auctions and bidding: A guide for computer scientists. **24**(2) (August 2004) 271–287
2. Lin, W.Y., Lin, G.Y., Wei, H.Y.: Dynamic Auction Mechanism for Cloud Resource Allocation. IEEE (May 2010)
3. Szymanski, B.: A Novel Auction Mechanism for Selling Time-Sensitive E-Services. IEEE
4. Asdemir, K.: Bidding patterns in search engine auctions (2006)
5. Gamma, E., Helm, R., Johnson, R., Vlissides, J.: Design Patterns: Elements of Resusable Object-Oriented Software. Addison-Wesley Professional (1995)
6. Henninger, S., Corrêa, V.: Software pattern communities Current practices and challenges. In: Proceedings of the 14th Conference on Pattern Languages of Programs (PLoP). (September 2007) 2007
7. Kampffmeyer, H., Zschaler, S.: Finding the pattern you need: The design pattern intent ontology. Lecture Notes in Computer Science **4735** (2007) 211

8. Henninger, S., Ashokkumar, P.: An Ontology-Based Metamodel for Software Patterns. at 18th Int. Conf. on Software Engineering . . . (2006)
9. Montero, S., Díaz, P., Aedo, I.: Formalization of web design patterns using ontologies. Lecture Notes In Computer Science (2003) 179–188
10. Harb, D., Bouhours, C., Leblanc, H.: Using an Ontology to Suggest Software Design Patterns Integration. Lecture Notes In Computer Science (2009) 318
11. Iglesias, C.A., Garijo, Fernández-Villamor, J.I., Durán, J.J.: Agreement patterns. In: Proceedings of the CAEPIA09 Workshop on Agreement Technologies(WAT 2009), Sevilla, WAT09-CAEPIA09 (November)
12. Noy, N.F., McGuinness, D.: Ontology Development 101: A Guide to Creating your First Ontology. Technical Report SMI-2001-0880, Stanford Medical Informatics, Stanford (2001)
13. Dignum, V., Vázquez-Salceda, J., Dignum, F. In: Omni: Introducing social structure, norms and ontologies into agent organizations. Springer-Verlag, New York (2005) 181–198
14. Tamma, V., Wooldridge, M., Dickinson, I.: An ontology based approach to automated negotiation. In Padget, J.A., Shehory, O., C. Parkes, D., Sadeh, N.M., Walsh, W.E., eds.: Agent Mediated Electronic IV. Designing Mechanisms and Systems. AAMAS 2002 Workshop on agent mediated electronic commerce (AMEC IV), held at the AAMAS 02 conference, Bologna, Springer Verlag (2002) 219–237
15. Tamma, V., Phelps, S., Dickinson, I., Wooldridge, M.: Ontologies for supporting negotiation in e-commerce. Engineering Applications of Artificial Intelligence **18**(2) (March 2005) 223–236
16. Paschke, A., Kiss, C., Al-Hunaty, S.: NPL: Negotiation Pattern Language. economics and management (2005)
17. Ré, R., Braga, R., Masiero, P.: A pattern language for online auctions management. Proceedings of PLoP (2001)
18. Wellman, M., Wurman, P., O'Malley, K., Bangera, R., Lin, S.d., Reeves, D., Walsh, W.: Designing the market game for a trading agent competition. IEEE Internet Computing **5**(2) (2001) 43–51
19. Oluyomi, A.O.: Patterns and Protocols for Agent-Oriented Software Development. PhD thesis, Faculty of Engineering. University of Melbourne, Australia. (November 01 2006)
20. Jureta, I., Kolp, M., Faulkner, S., Do, T.T.: Patterns for Agent Oriented e-Bidding Practices. In: Proceedigns of 9th International Conference on Knowledge-Based Intelligent Information and Engineering Systems (KES 2005), Part IIKES (2), Melbourne,Australia (2005) 814–820
21. Koolmanojwong, S., Jiamthapthaksin, R., Daengdej, J.: An agent architecture for competitive application environment. Aerospace Conference, 2004. Proceedings. 2004 IEEE 3079–3089
22. Vytelingum, P., Cliff, D., Jennings, N.R.: Strategic bidding in continuous double auctions. Artificial Intelligence **172**(14) (2008) 1700–1729
23. Stanford Center for Biomedical Informatics Research: Protégé ontology editor and knowledge acquisition system. http://protege.stanford.edu/ (2006)
24. Nishino, N., Fukuya, K., Ueda, K.: Service Design in Movie Theaters Using Auction Mechanism with Seat Reservations
25. Edelman, B., Ostrovsky, M.: Strategic bidder behavior in sponsored search auctions. Decision Support Systems **43**(1) (2007) 192–198
26. Harb, D., Bouhours, C., Leblanc, H.: Using an ontology to suggest software design patterns integration. In Chaudron, M., ed.: Models in Software Engineering.

Volume 5421 of Lecture Notes in Computer Science. Springer Berlin / Heidelberg (2009) 318–331
27. Pavlic, L., Hericko, M., Podgorelec, V.: Improving design pattern adoption with Ontology-Based Design Pattern Repository. IEEE (June 2008)