Ontological Issues in Agent-aware Negotiation Services*

Andrea Giovanucci[1] and Juan A. Rodríguez-Aguilar[1,2]

[1] iSOCOLab
Intelligent Software Components, S. A.
Edificio Testa, C/ Alcalde Barnils, 64-68 A
08190 Sant Cugat del Vallès, Barcelona, Spain
{andrea,jar}@isoco.com
http://www.isoco.com
[2] Artificial Intelligence Research Institute, IIIA
Spanish Council for Scientific Research, CSIC
08193 Bellaterra, Barcelona, Spain.
jar@iiia.csic.es
http://www.iiia.csic.es

Abstract. In negotiation events involving multiple, highly customisable goods, buying agents need to express relations, constraints, and preferences over attributes of different items. Not forgetting the provider side, providing agents may also impose constraints or conditions over their offers. Thus, there is the need for providing all trading agents involved with a rich enough language to express their business rules. In this paper we focus on the ontological issues that need to be considered in order to empower the expressiveness offered by negotiation objects and offers to incorporate buyers' and providers' business preferences. We take the stance that a highly expressive ontology is compulsory to enact agent-aware negotiation services in actual-world procurement settings.

1 Introduction

Although negotiation is a key procurement mechanism, to the best of our knowledge most agent-based services deployed so far have primarily focused on infrastructure issues related to negotiation protocols. Furthermore, no special attention has been paid to developing negotiation ontologies that provide the expressiveness required in actual-world settings. Thus, the lack of agent-based decision support for trading agents that help improve current trading practices hinders the adoption of agent technology in procurement scenarios.

Consider the problem faced by a buying agent when negotiating with providing agents. In a negotiation event involving multiple, highly customisable goods, buying agents need to express relations and constraints between attributes of

* Accepted at the *Agentes Inteligentes para el Tercer Milenio* Workshop to be held at the 10th Conference of the Artificial Intelligence Spanish Society (CAEPIA'2003), November 11, 2003, San Sebastian, Spain.

different items. On the other hand, it is common practice to buy different quantities of the very same product from different providing agents (multiple sourcing), either for safety reasons or because offer aggregation is needed to cope with high-volume demands. This introduces the need to express business constraints on the number of providing agents and the amount of business assigned to each of them. Not forgetting the provider side, providing agents may also impose constraints or conditions over their offers. Offers may be only valid if certain configurable attributes (f.i. quantity bought, delivery days) fall within some minimum/maximum values, and assembly or packing constraints need to be considered. Once the buying agent collects offers, he is faced with the burden of determining the winning offers. Furthermore, there is the need for providing all trading agents involved with a rich enough language to express their business preferences.

In [5] we introduced *iBundler* an agent-aware decision support service that relieves buying agents from determining the winning offers based on the formal model thoroughly described in [6]. In this paper we primarily focus on the ontological issues that need to be considered in order to empower the expressiveness offered by negotiation objects and offers to incorporate buyers' and providers' business preferences. At such aim, our approach required the extension of state-of-the-art ontologies for automated negotiation[7, 8].

The paper is organised as follows. Section 2 introduces the market scenario where buyers and providers are to negotiate, along with the expressiveness requirements they may need. Next, a slight description of *iBundler*, an agent-aware decision support service for combinatorial negotiations, is provided in section 3. Thereafter, the ontology required to enact the service is thoroughly described in 4. Finally, section 5 summarises our contributions and proposes future research lines.

2 Market scenario

Next we detail the capabilities required by buyers and providers in an actual-world procurement scenario. The requirements below are intended to capture buyers' constraints and preferences and outline a highly expressive bidding language for providing agents:

Negotiate over multiple items. A negotiation event is usually started with the preparation of a request for proposal (RFQ) form. The RFQ form describes in detail the requirements (including attribute-values such as volume, quality specifications, dates as well as drawings and technical documentation) for the list of items (goods or services) defined by the negotiation event.

Offer aggregation. A specific item of the RFQ can be acquired from several providers simultaneously, either because not a single provider can provide with the requested quantity at requested conditions or because buyers explicit constraints (see below).

Business sharing constraints. Buyers might be interested to restrict the number of providers that will finally trade for a specific item of the RFQ, either for security or strategical reasons. It is also of usual practice to define the minimum amount of business that a provider may gain per item.

Constraints over single items. Every single item within an RFQ is described by a list of negotiable attributes. Since: a) there exists a degree of flexibility in specifying each of these attributes (i.e. several values are acceptable) and b) multiple offers referring the very same item can be finally accepted; buyers need to impose constraints over attribute values. An example of this can be the following: suppose that the deadline for the reception of certain item A is two weeks time. However, although items may arrive any day within two weeks, once the first units arrive, the rest of units might be required to arrive in no more than two days after.

Constraints over multiple items. In daily industrial procurement, it is common that accepting certain configuration for one item affects the configuration of a different item, for example, when dealing with product compatibilities. Also, buyers need to express constraints and relationship between attributes of different items of the RFQ.

Specification of providers' capacities. Buyers cannot risk to award contracts to providers whose production/servicing capabilities prevent them to deliver over-committed offers. At this aim, they must require to have providers' capacities per item declared. Analogously, next we detail the expressiveness of the bidding language required by providers. The features of the language below are intended to capture providing agents' constraints and preferences.

Multiple bids over each item. Providers might be interested in offering alternate conditions/configurations for a same good, i.e., offering alternatives for a same request. A common situation is to offer volume-based discounts. This means that a provider submits several offers and each offer only applies for a minimum (maximum) number of units.

Combinatorial offers. Economy efficiency is enhanced if providers are allowed to offer (bid on) combination of goods. They might lower the price, or improve service assets if they achieve to get more business.

Multi-unit offering. Each provider needs to specify that they will only participate in trading if a minimum (maximum) amount of business is assigned to him.

Homogeneous combinatorial offers. Combinatorial offering may produce inefficiencies when combined with multi-unit offering. Thus a provider may wind up with an award of a small number of units for a certain item, and a large number of units for a different item, being both part of the very same offer (e.g. 10 chairs and 200 tables). It is desirable for providers to be able to specify homogeneity with respect to the number of units for complementary items.

Packing constraints. Packing units are also a constraint, in the sense that it is not possible to serve an arbitrary number of units (e.g. a provider cannot sell 27 units to a buyer because.his items come in 25-unit packages). Thus providers require to be capable of specifying the size of packing units.

Complementary and exclusive offers. Providers usually submit XOR bids, i.e., exclusive offers that cannot be simultaneously accepted. Also, there may exist the need that an offer is selected only if another offer is also selected. We refer to this situation as an AND bid. This type of bids allows to express volume-based discounts. For example, when pricing is expressed as a combination of base price and volume-based price (e.g. first 1000 units at $2.5 p.u. and then $2 each).

Obviously, many more constraints regarding pricing and quantity can be considered here. But we believe these faithfully address the nature of the problem.

3 Agent-aware negotiation support service

The *iBundler* service has been implemented as an agency composed of agents and software components that cooperatively interact to offer a decision-support service for negotiation scenarios. *iBundler* can act as a combinatorial negotiation solver for both multi-item, multi-unit negotiations and auctions. Thus, the service can be employed by both negotiating agents and auctioneers in combinatorial auctions.

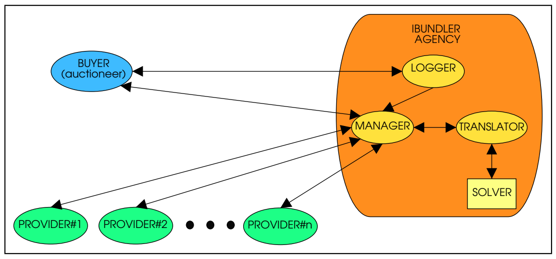

Fig. 1. Architecture of the *iBundler* agency

Figure 1 depicts the components of the agency, along with the fundamental connections of buying and providing agents with the service. Next we make explicit the main functionality of its members:

[Logger agent]. It represents the interface of the *iBundler* agency to the world. The Logger agent is in charge of facilitating registration and unregistration with the *iBundler* service to users (both buyers and providers) as well as their subsequent access to the service via log in and log out procedures.

[Manager agent]. Agent devoted to providing the solution of the problem of choosing the set of bids that best matches a user's requirements. There exists a single Manager agent per user (buying agent or auctioneer), created by the Logger agent, offering the following services: brokering service to forward buying agents' requirements (RFQs) to selected providing agents capable of fulfilling them; collection of bids; winner determination in a combinatorial negotiation/auction; award of contracts on behalf of buying agents. Furthermore, the manager agent is also responsible for: bundling each RFQ and its bids into a negotiation problem in FIPA-compliant[3] format to be conveyed to the Translator agent; and to extract the solution to the negotiation problem handled back by the Translator agent. Observe that figure 1 shows the interplay of buying and providing agents with the Manager as the sole access point to the *iBundler* agency.

[Translator agent]. It creates an XML document representing the negotiation problem in a format understandable by the Solver departing from the FIPA-compliant description received from the Manager. It also translates the solution returned by the Solver into an object of the ontology employed by user agents. It is the bridge between the language spoken by user agents and the language spoken by Solver.

[Solver component]. The *iBundler* component itself extended with the offering of an XML language for expressing offers, constraints, and requirements. The XML specification is parsed into an MIP formulation and solved using available MIP solvers as described in [5].

4 Ontology

Although research on automated negotiation in multi-agent systems has concentrated on the design of negotiation protocols and their associated strategies, ontological aspects of negotiation protocols have recently started to attract researchers' attention (see [7, 8] and the results of the ADMIT project [2]). In [7, 8] we find an ontological approach to automated negotiation founded on the following concepts: *negotiation protocol* (rules followed by participants during a negotiation process), *party* (participants, be them either human agents, software agents or even organisations of agents), process (way to reach an agreement on some issue by modifying negotiation attributes), (negotiation) *object*, *offer* (possible combination of values associated to the negotiation attributes which represent an expression of will), *negotiation rule* (set of rules that govern a specific negotiation protocol). Although satisfactory enough for most concepts, particularly as to negotiation protocols regarded as processes and rules, in this work we had to enrich the concepts of *offer* and *object* in order to accommodate the expressiveness required by the actual-world constraints described in section 2 for bids and RFQs respectively. To the best of our knowledge, no ontology defined in prior work allows us the expressiveness that buying and providing agents require. In other words, there is no adequate ontology for multi-item, multi-unit combinatorial reverse auctions with side constraints. Thus we had to define an *ad-hoc* ontology for the *iBundler* service.

The ontology has been defined with the aid of *Protege 2000* [4]. Furthermore, the conversion from ontological objects to Java classes is realised via the *beangenerator* Protege 2000 plug-in[9]. The automatically-generated Java classes fulfill with the JADE specification in [1].

Figures 2, 3, and 4 provide graphical representations (as shown by the Ontoviz Protégé plug-in) of the core concepts in the *iBundler* ontology, namely, and respectively, the *RFQ*, *ProviderResponse*, *Problem*, and *Solution* concepts.

The *RFQ* concept is employed by buying agents to express their requests for bids. Figure 2 shows that an RFQ is composed of a sequence of *Request* con-

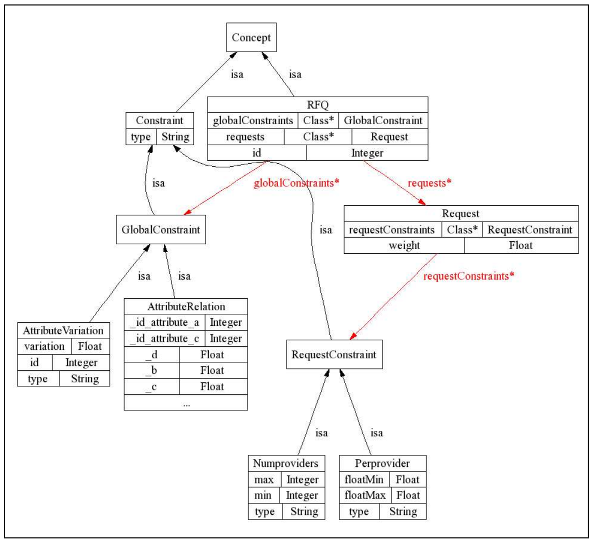

Fig. 2. RFQ concept representation

cepts, one per requested item. A sequence of global constraints (*GlobalConstraint* concept) relating separate, requested items may be part of an RFQ. There are two types of *GlobalConstraint* concepts: constraints that allow to express linear relationships between different attributes of the very same or separate item(s) (*AttributeRelation* concept) and constraints on the values of an item's attribute (AttributeVariation concept). A sequence of constraints on individual items (*RequestConstraint* concept) may be also part of an RFQ. Constraints on individual items can serve to limit the range of providers (*NumProviders* concept) to which the item can be awarded or the range of percentage of units to be awarded to the very same provider (*PerProvider* concept). Notice that all the constraints specified in an *RFQ* stand for the buyer's business rules.

On the provider side, providing agents express their offers in terms of the *ProviderResponse* concept, which in turn is composed of several elements: a list of *Bid* concepts (each *Bid* allows to express a bid per either a single requested item or a bundle of items); constraints on the production/servicing capabilities of the bidding provider (*Capacity* concept); and constraints on bundles of bids formulated with the *BidConstraint* concept (each *BidConstraint* in turn can be of exclusive –xor– or volume-based discount type –and–, corresponding respectively to the *XOR* and *AND* concepts). Whereas constraints on bundles of bids put into relation separate bids, constraints on individual bids (expressed as *SingleBidConstraint* concepts) allow to relate the values offered for separate items within the very same bid. As an example, homogeneity constraints can be declared by providers within some bid to make buyers aware that the quantity of items they can select per item must be the same, or else the provider will not

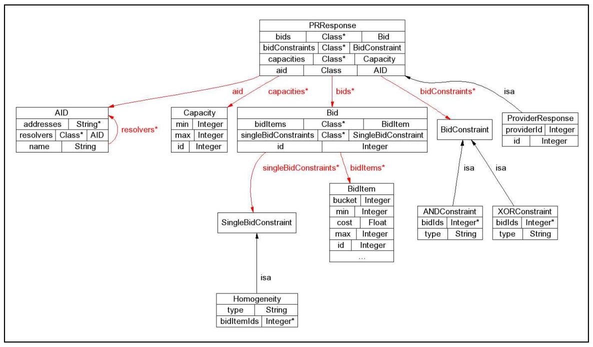

Fig. 3. Bid representation as part of the provider response concept

concede his bid. Such constraint maps to the *Homogeneity* concept, a particular type of *SingleBidConstraint*.

Once the manager collects all offers submitted by providers, he wraps up the *RFQ* concept as received from the buyer along with the offers as *ProviderResponse* concepts to compose the negotiation problem to be solved by the *Solver* component. The resulting concept, *Problem*, is depicted in figure 4.

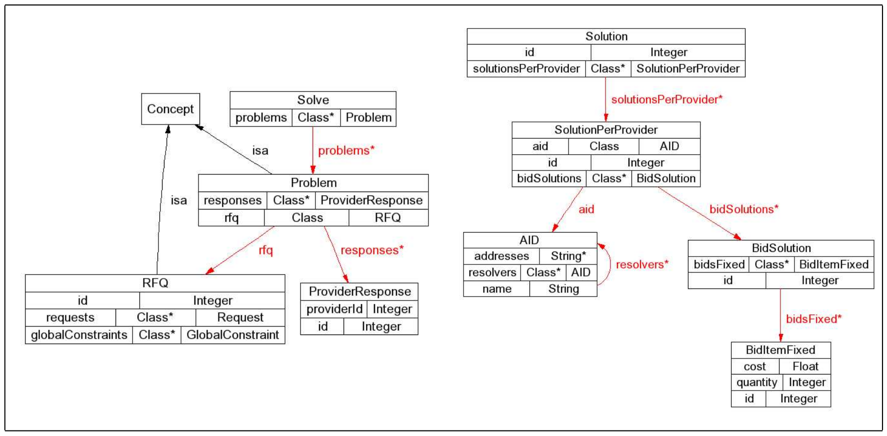

Fig. 4. Problem and Solution concepts

Finally, the solution produced by the Solver component is transformed by the translator agent into a *Solution* concept (see figure 4) that is handed over to the manager. The *Solution* concept contains the specification of the optimal set of offers calculated by Solver. Thus *Solution* contains a list of *SolutionPerProvider* concepts, each one containing the bids selected in the optimal bid set per provider, as a list of *BidSolution* concepts, along with the provider's agent identifier, as an *AID* concept. Each *BidSolution* in turn is composed of a list of *BidItemFixed* concepts containing the number of units selected per bid along with the bid's total cost.

So far we have concentrated on ontological concepts referring to entities with a complex structure that can be defined in terms of slots. Hereafter we shall draw our attention to agent actions, i.e. the special concepts that indicate actions that can be performed by agents in the *iBundler* agency, as well as buyers and providers. Figure 5 depicts all available agent actions as descendants of a single common ancestor: *AgentAction*. Observe that the different agent actions correspond to the services offered by each agent.

Thus, the Logger agent offers the services associated to the following actions:

Login Action requested by trading agents when logging in with the *iBundler* agency.

Logout Action requested by trading agents when logging out of the *iBundler* agency.

Register Action requested by trading agents when signing for the *iBundler* agency. They must provide information about themselves. At the end of the registration, the Logger provides them with a user name and a password.

Unregister Action requested by trading agents when unregistering with the *iBundler* agency. They must provide information about themselves. All the brokering information associated to them is erased by the Logger.

As to the manager, it offers four core services via the following actions:

GetAllBids The buyer specifies an RFQ along with a list of providers. The manager agent forwards the query to all the providers and delivers back all the responses to the buyer.

Solve The buyer sends to the manager an *RFQ* along with a list of *Provider-Response* concepts representing providers' offers. The manager composes a *Problem* out of the *RFQ* and *ProviderResponses* to subsequently request the translator agent for a *Solution*. In this way, the buyer is relieved from the intricate construction of a *Problem* involving the creation of crossed references between *RFQ* and the *Bid* concepts in each *ProviderResponse*. Once the *Solution* is received by the manager it is forwarded to the buyer.

Manage The buyer sends to the manager an *RFQ* along with a list of providers. The manager sends a filtered version of the *RFQ* (removing the buyer's private constraints) to available providers, collects all their offers, constructs a *Problem* to ask the translator for a *Solution*, which is conveyed to the buyer once calculated.

Buy The buyer constructs a *Solution* concept and subsequently asks the manager to the bids and providers in the list of *SolutionsPerProvider*. Notice that the buyer may employ the same *Solution* concept recommended by the manager as an optimal solution.

Finally, providers do offer their services through the following actions:

RequestForQuotations Request for offers received from the manager for a filtered version of the *RFQ* sent by the buyer.

BuySolution Order to buy selected offers received from the manager.

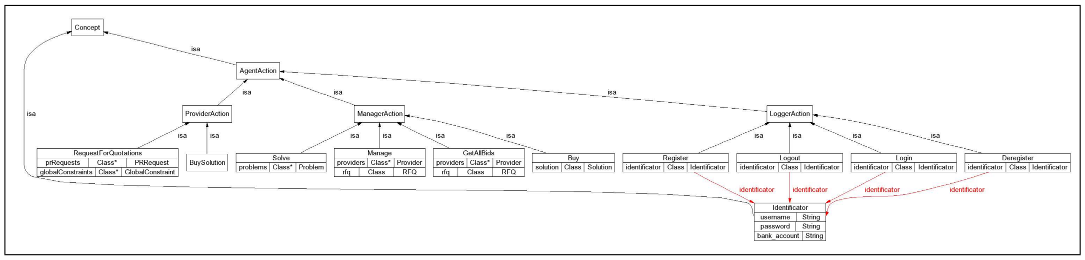

Fig. 5. Agent actions

5 Summary and Future Work

We believe that it is time to deploy agent services aimed at complex problem-solving so that they can be subsequently employed by other agents either to help them team up and cooperatively solve complex problems or to behave more efficiently in competitive scenarios. The *iBundler* service contributes along two main directions. On the one hand, it incorporated new, actual side constraints to the winner determination problem for multi-item, multi-unit, multi-attribute negotiations[6]. On the other hand, it includes a new ontology that accommodates both operational constraints and attribute-value constraints for buying and providing agents.

However, we have identified two important research issues as future work . In particular, it would be interesting to assess the overload added by the use of our ontology when employed in actual-world settings. On the other hand, it might be of interest to merge our proposal with the ontology in [8, 7], more focused on procedural issues of negotiation protocols.

References

1. Giovanni Caire. Jade tutorial. application-defined content languages and ontologies. Technical report, TILAB S.p.A., June 2002.
2. Agentcities Consortium. Demonstration documentation for checkpoint 1. Technical Report Deliverable D5.4, IST-2000-28385, August 2002.
3. Fipa. http://www.fipa.org.
4. Protege 2000. http://protege.stanford.edu.
5. Juan A. Rodríguez-Aguilar, Andrea Giovanucci, Antonio Reyes-Moro, Francesc X. Noria, and Jesús Cerquides. Agent-based decision support for actual-world procurement scenarios. In *2003 IEEE/WIC International Conference on Intelligent Agent Technology*, Halifax, Canada, October 2003.

6. Juan A. Rodríguez-Aguilar, Antonio Reyes-Moro, Andrea Giovanucci, and Jesús Cerquides Francesc X. Noria. Negotiation support in highly-constrained trading scenarios. In *Proceedings of the Conference of the Spanish Association for Articial Intelligence (CAEPIA)*, San Sebastian, Spain, November 2003. Forthcoming.
7. V. Tamma, M. Wooldridge, and I. Dickinson. An ontology for automated negotiation. In *First Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS'02)*, Bologna, Italy, July 2002.
8. Valentina Tamma. An experience in using ontologies for automated negotiation. Presentation at the Agentcities Information Day 4, August 2003.
9. C. J. van Aart. Beangenerator. http://www.swi.psy.uva.nl/usr/aart/beangenerator.