Bargaining agents in wireless contexts: an alternating-offers protocol for multi-issue bilateral negotiation in mobile marketplaces

Azzurra Ragone[1], Michele Ruta[1], Eugenio Di Sciascio[1], Francesco M. Donini[2]

[1] SisInfLab, Politecnico di Bari, Bari, Italy
{a.ragone,m.ruta,disciascio}@poliba.it
[2] Università della Tuscia , Viterbo, Italy
donini@unitus.it

Abstract. We present an approach to multi-issue bilateral negotiation for mobile commerce scenarios. The negotiation mechanism has been integrated in a semantic-based application layer enhancing both RFID and Bluetooth wireless standards. OWL DL has been used to model advertisements and relationships among issues within a shared common ontology. Finally, non standard inference services integrated with utility theory help in finding suitable agreements. We illustrate and motivate the provided theoretical framework in a wireless commerce case study.

1 Introduction and Motivation

Automated negotiation mechanisms call for adopting logical languages to model advertisements allowing to perform inferences wherever is the need to go beyond plain undifferentiated goods and not only a single issue (usually price) is amenable to negotiation. OWL DL is a natural candidate for this purpose: it is one of the reference formalisms for the Semantic Web effort, with a formal logic-based semantics, much more expressive than *e.g.*, Propositional Logic, yet decidable. Indeed the formal semantics of OWL DL is based on Description Logics (DL) one. Hence, it can be useful in a number of negotiation scenarios. In this paper we present a OWL DL-based approach to multi-issue bilateral negotiation in a mobile context. Wireless agents exploit a logic-based alternating-offers protocol integrated in a mobile resource discovery layer for pervasive environments. Basically, we adapt languages and technologies from the Semantic Web vision to pervasive environments. In great detail, we propose an extension of both EPCglobal RFID standard [7] and Bluetooth Service Discovery Protocol [3] supporting formalisms for knowledge representation and enabling advanced negotiation services. Semantic-based annotations are stored on RFID tags attached to products and goods which can describe themselves to a reader gateway. It acts as front end in a possible Bluetooth-based interaction with the user PDA. Noteworthy is the feature to preserve the original code-based discovery of both Bluetooth and RFID technologies, thus keeping a legacy compatibility with basic applications. Furthermore, according to W3C recommendations for mobile applications, our approach copes with limited storage and computational capabilities of mobile and embedded devices, and with reduced bandwidth provided by wireless links. Issues related to the verbosity of semantic annotation languages cannot be neglected. Hence, in order to make our approach sustainable in reality, we exploited a novel efficient XML compression algorithm, specifically targeted for DIG 1.1 [2] document instances. Machine understandable ontological languages are exploited to perform non-standard reasoning services integrating utility theory to find the most suitable agreements. To this aim existing relationships among issues in requests and offers and related preferences of agents (both expressed by means of logical formulas) have to be taken into account. The proposed framework has been devised for intelligent recommendation purposes in furnishings stores. Using a PDA s/he can interrogate the store software application and be able to assemble components, select colors, materials, type of shelves or drawers, complements like hangers, rails, baskets and different type of organizers. Each component is endowed with an RFID tag hosting the semantic annotation of the component itself. Hence, the customer can select his preferred configuration and so instruct his wireless software agent to negotiate with other mobile-agents running on the seller side. Each agent will encourage a specific configuration the retailer prefers to sell[3]. Main contributions of the paper are:

– *The exploitation of a logic-based approach in the negotiation process*. The availability of an ontology modeling the reference domain knowledge allows to annotate relationships among components, so ruling out components not compatible with each other. In the same way, it allows to discover similar components replacing missing ones. Moreover, inference services can be used in order to reveal conflicting information between a request and an offer ("a wardrobe with *sliding doors* w.r.t. another one with *hinged doors*"). Finally, a semantic-based approach allows the system to provide a *result explanation* of the negotiation process: for each configuration the system will show what the user has to give up w.r.t. his initial request and what is still in the configuration.

– *The exploitation of mobile technologies in supporting the semantic-based interaction between negotiating agents*. By attaching an RFID tag on each product, the retailer agent will be able to negotiate only on available goods and, on the other hand, the user agent can interact with it "from everywhere". That is, the negotiation process will happen taking into account objects actually available in a virtual shelf (if a piece has been picked-up by a customer which is still in the store it will be labeled as not available for the negotiation process). This "real time" availability is therefore taken into account in determining a set of preferred configurations the seller wants to offer.

– *An agreement smart suggestion*. The negotiation process could find an agreement (the proposed configuration) beneficial for both the retailer and the buyer, by taking into account utilities of both participants (this is a bilateral matchmaking process rather than an unilateral one, as requirements and preferences from both buyer and retailer are considered).

– *A support for non-expert users*. A user-friendly graphical interface hides any technical underlying detail (*e.g.*, logic expressions). Users equipped with their PDAs only have to drag and drop components on a canvas and set utilities (if they want) for them. Then, after the interaction with the system, they will just scroll the negotiation results. Returned configurations are ranked w.r.t. utility functions, but the user keeps the final

[3] We assume the retailer aims at selling *e.g.*, configurations on special, like offers of the week.

$\texttt{HingedDoors} \sqsubseteq \texttt{Shape}; \texttt{Oak} \sqsubseteq \texttt{Wood}; \texttt{SlidingDoors} \sqsubseteq \texttt{Shape}; \texttt{Birch} \sqsubseteq \texttt{Wood};$

$\texttt{HingedDoors} \sqsubseteq \neg\texttt{SlidingDoors}; \texttt{Oak} \sqsubseteq \neg\texttt{Birch}; \texttt{Basic\_clothes\_rail} \sqsubseteq \texttt{Hangers};$

$\texttt{WireBasket} \sqsubseteq \texttt{Drawer}; \texttt{ShoeBox} \sqsubseteq \texttt{Box};$

$\texttt{ClosedWardrobe} \equiv \texttt{Wardrobe} \sqcap \exists\texttt{material} \sqcap \exists\texttt{basicShape} \sqcap \exists\texttt{shutters};$

$\texttt{Walk\_in\_closet} \equiv \texttt{Wardrobe} \sqcap \neg\exists\texttt{shutters};$

$\texttt{TidyShoeBox} \equiv \texttt{ShoeBox} \sqcap \exists\texttt{material} \sqcap \forall\texttt{material}.\texttt{Plastic};$

$\texttt{Add\_on\_clothes\_rail} \sqsubseteq \texttt{Basic\_clothes\_rail} \sqcap \exists\texttt{complement} \sqcap \forall\texttt{complement}.(\texttt{Tie\_rack} \sqcap \texttt{Trouser\_hanger})$

Fig. 1. The ontology used in the examples

decision; he can either choose one of the proposed configurations or simply drop out and/or start a new negotiation changing preferences.

The rest of the paper is organized as follows: next Section focuses on the adopted logical formalisms, Section 3 presents and motivates the negotiation protocol we devised and exploited and Section 4 outlines the reference communication architecture. Finally, Section 5 clarifies our settings with the aid of a toy example and conclusion closes the paper.

2 Semantic annotations

In this section we briefly illustrate the semantic annotations of resources in our application. We use to annotate resources a fragment of OWL-DL, where besides `owl:Class` and `owl:ObjectProperty`, one is able to express `owl:DataTypeProperty` f (for **F**eatures) on objects such as height, width, length and many others by means of *concrete domains*. In this paper we refer to $\mathcal{AL}(D)$ [1] subset of OWL DL. Hereafter we will use DL syntax which results more compact than OWL one. In order to model the domain knowledge and represent relationships among elements, we use an ontology $\mathcal{O}$ containing Concept Inclusion axioms of the form $A \sqsubseteq C$ and $A \equiv C$, where the concept name A can appear only once on the left-had side of axioms. We restrict $\mathcal{O}$ to be acyclic, *i.e.*, the definition of A should not depend on A itself (see [1] for a precise definition). Using $\mathcal{AL}(D)$ it is possible to express subclass relations and disjointness relations involving concept names, *e.g.*, $\texttt{Birch} \sqsubseteq \texttt{Wood}$ and $\texttt{HingedDoors} \sqsubseteq \neg\texttt{SlidingDoors}$. Formulas representing demands D and configurations S, are expressed as generic OWL-DL expressions. So, an example description can be the following one:

$\texttt{Wardrobe} \sqcap \forall\texttt{basicShape}.\texttt{SlidingDoors} \sqcap \forall\texttt{material}.\texttt{Birch} \sqcap = 4\texttt{drawers} \sqcap =_{200} \texttt{height}$

formally modeling this configuration: "*a wardrobe made from birch wood with sliding doors, four drawers and 200 meters high*". Notice that for what concerns numerical properties, also range expressions are allowed in the form $(f \geq n) \sqcap (f \leq m)$. In order to better explain the approach, in the rest of the paper we will refer to ontology $\mathcal{O}$ in Figure 1, where some attributes proper of a wardrobe are listed and related to each other through the ontology itself. Even though subsumption and satisfiability are basic and useful reasoning tasks in a number of applications, there are typical problems related to negotiation that call for non-standard reasoning services. For instance, suppose you have the buyer's agent β with her **D**emand represented by the concept D and the seller's agent σ with his **S**upply represented by S. In case β's request D and σ's offer S are in conflict with each other with respect to the domain knowledge modeled in the ontology $\mathcal{O}$— in formulae $S \sqcap D \sqsubseteq_{\mathcal{O}} \perp$ — how to suggest to β which parts in D are in conflict with S and conversely to σ which parts in S are conflict with D? The above question is very significant in negotiation scenarios where you need to know "what is wrong" between D and S and negotiate on it. In order to give an answer to the previous question and provide explanations, Concept Contraction [4] can be exploited.

Concept Contraction . Given two concepts C_1 and C_2 and an ontology $\mathcal{O}$, where $C_1 \sqcap C_2 \sqsubseteq_{\mathcal{O}} \perp$ holds, find two concepts K (for Keep) and G (for Give up) such that both $C_1 \equiv K \sqcap G$ and $K \sqcap C_2 \not\sqsubseteq_{\mathcal{O}} \perp$.

In other words K represents a contraction of C_1 which is satisfiable with C_2, whilst G represents some reasons why C_1 and C_2 are not compatible with each other. With Concept Contraction, both *conflicting information* in β's request w.r.t. σ'supply can be computed and vice versa. Actually, for Concept Contraction minimality criteria have to be introduced. Following the Principle of Informational Economy [8], for G we have to give up as little information as possible. In [4] some minimality criteria were introduced and analyzed. In particular, if the adopted DL admits a normal form with conjunctions of concepts as $\mathcal{AL}(D)$, $G_{\exists}$ *minimal* irreducible solutions can be defined [10].

3 The negotiation mechanism

Here, we outline the negotiation mechanism, taking into account the semantics of D and S as well as the domain knowledge modeled within an ontology in the OWL DL fragment we identified in Section 2, exploiting Concept Contraction. We start describing: the *negotiation protocol*, *i.e.*, the set of rules specifing how an agreement can be reached; the *negotiation strategy*, that specifies the actions to take by agents given the protocol [11]; the *utility function* of the agents, used to evaluate negotiation outcomes [9]. We note that the mechanism is a **one-to-many** negotiation, since the buyer's agent will negotiate simultaneously with other *m* different agents – each one linked with a particular configuration the retail prefers to sell. Moreover, it is a negotiation with **incomplete information** as each agent knows its utility function and ignores the opponent disagreement thresholds and utility functions. Finally, as agents are **rationale** they will never accept an agreement if the agent's utility over such an agreement is smaller than disagreement thresholds[4] set by the agent before negotiation starts. The protocol is inspired to Rubinstein's alternating-offers one [12]. Our protocol, anyway, is different from that of Rubinstein; actually we consider *multi-issue negotiation*: negotiation is not on a single item (or on a single bundle of), but on many issues related with each other through an ontology. The protocol has a finite set of steps[5]: the negotiation terminates

[4] Disagreement thresholds, also called disagreement payoffs, or reservation values, are the minimum utility that each agent requires to pursue a deal [11].

[5] In the following, for the sake of simplicity, we always describe an interaction between only two opposite agents; although we notice that multiple negotiations can be performed at the same time, among *one* agent and *many* candidate partners.

either because the agreement has been reached or because one agent opts out. The agent who moves first is selected randomly, at each step the agent who moves has two choices: *concede* or *opt out*, while the other one *stands still*. Agents are forced to concede until a *logical compatibility* is reached between the initial request and the supply, *i.e.*, until the inconsistency sources are eliminated. At each step, amongst all the concessions allowed by the protocol, the agent choose the *minimal concession*, *i.e.*, *minimal* w.r.t. the utility loss paid by the agent who makes the concession [6]. The negotiation ends either if a logical compatibility is reached or if one agent opts out (*conflict deal*). For what concerns **strategy**, if it is its turn to move, an agent can choose to **opt out** if its utility at that step is smaller than its disagreement threshold, then the negotiation ends immediately. Otherwise, it **concedes**: the concession is the *minimum possible concession*, that is the concession less decreasing its utility. Here, our main focus is not on how to compute utility functions, however we give a hint on that topic. We define an agent's **utility function** over all possible outcomes [9] as: $u^p : \mathcal{A} \cup \{Opt\} \rightarrow \Re$ where $p \in \{\beta, \sigma\}$ —β and σ stand for buyer and seller respectively— $\mathcal{A}$ is the set of all possible agreements, *Opt* stands for Opt out. For what concern the buyer, she can set utilities while choosing components through the GUI, however, we point out that she *does not have to*, *i.e.*, she can only set utilities for components she deems very important[6]. For the retailer, instead, the utility of each agent is computed based on several parameters, that can be changed, updated, deleted over time by the retailer itself. A utility value will be coupled with each configuration and will be computed as a weighted function of *e.g.*, the real-time availability of each piece and its cost, so taking into account unsold stocks.

3.1 Logic-based Alternating-offers Protocol

For the sake of clarity, from now on we indicate the buyer's agent with β and her potential partners (retail's agents) with σ. The first step of the protocol is the normalization of both β's demand D and σ's configuration S. The normalization step substitutes A with $A \sqcap C$ everywhere in a concept, if either $A \sqsubseteq C$ or $A \equiv C$ appears in $\mathcal{O}$, then considers the equivalence $\forall R.(A \sqcap B) \equiv \forall R.A \sqcap \forall R.B$ as a recursive rewrite rule from left to right. After the normalization stage, D is a conjunction of elements in the form: $D = \bigsqcap_i C_i$, where C_i represents issues the user is willing to negotiate on. Users can express utilities on single issues or on bundles of them. For instance, w.r.t. the previous request the buyer may set utility values on $\forall$`basicShape`.`SlidingDoors` (single issue) as well as on the whole formula $\forall$`basicShape`.`SlidingDoors` $\sqcap$ $\exists$`material`$\sqcap\forall$`material`.`Birch` (bundle of issue). We indicate these concepts with P_k — for **P**references. To each P_k will be attached a utility value $u^\beta(P_k) \geq 0$. The global utility is a suitable sum of the utilities for preferences entailed by the final agreement[7]. In particular, given a concept expression A representing a final agreement, we

[6] The system will therefore work also for lazy users, unwilling to set a large number of utilities.

[7] Both agents' utilities are normalized to 1 to eliminate outliers, and make them comparable: $\sum_i u^\beta(P_k) = 1$.

define the final utility associated to the agent p, with $p \in \{\beta, \sigma\}$ as:

$$u^p(A) = \sum_k \{u^p(P_k) \mid A \sqsubseteq P_k\} \qquad u^p(Opt) = t_p \tag{1}$$

where t_p is the **disagreement threshold** of agent p. Summing up, before the real negotiation starts (step 0) we have a demand D and a configuration S such that: $D = \bigsqcap_i C_i$ and $S = \bigsqcap_j C_j$. Preferences P_k (for the buyer) and P_h (for the retailer) will be associated to concepts (or bundle of) C_i and C_j. Finally, both for β and σ we have the corresponding **disagreement thresholds** and utility functions t_β, u^β and t_σ, u^σ. If $D \sqcap S \sqsubseteq_{\mathcal{O}} \bot$ then demand and supply descriptions are in conflict with each other and β and σ need to negotiate on conflicting information if they want to reach an agreement. The negotiation will follow an alternating offers pattern: at each step, either β or σ gives up a portion of its conflicting information choosing the item with the minimum utility. Notice that both agents β and σ know D and S, but they have no information neither on counterpart utilities nor preferences. Both β and σ will solve two Concept Contraction problems, computing a $G_\exists$ *minimal irreducible* solution, and rewrite D and S as:

$$D = G_0^\beta \sqcap K_0^\beta \qquad\qquad S = G_0^\sigma \sqcap K_0^\sigma$$

Where G_0^β (respectively G_0^σ) represent the sources of conflict and the reason why D (S) is in conflict with S (D). Since we compute *G-irreducible* solutions we can normalize G_0^β and G_0^σ, following the same procedure for D and S, as:

$$G_0^\beta = G_{(0,1)}^\beta \sqcap G_{(0,2)}^\beta \sqcap \ldots \sqcap G_{(0,n)}^\beta = \bigsqcap_{i=1}^{n} G_{(0,i)}^\beta; \;\; G_0^\sigma = G_{(0,1)}^\sigma \sqcap G_{(0,2)}^\sigma \sqcap \ldots \sqcap G_{(0,m)}^\sigma = \bigsqcap_{j=1}^{m} G_{(0,j)}^\sigma$$

In the previous formulas, indexes $(0, i)$ and $(0, j)$ represent the i-th and j-th conjunctive element in G^β and G^σat round 0. Due to the structure of D, S and $\mathcal{O}$ we have that: *for each $G_{(0,i)}^\beta$ there always exists a C_i in the normalized version of D such that $G_{(0,i)}^\beta = C_i$*. The same relation holds between each $G_{(0,j)}^\sigma$ and C_j in the normalized form of S. Hence, some of P_k and P_h can be partially rewritten in terms of $G_{(0,i)}^\beta$ and $G_{(0,j)}^\sigma$ respectively. Since the information in G_0^β and G_0^σ are the reason why an agreement is not possible, then either β or σ will start conceding one of $G_{(0,i)}^\beta$ or $G_{(0,j)}^\sigma$ reducing their global utility of $u(G_{(0,i)}^\beta)$ or $u(G_{(0,j)}^\sigma)$, respectively. Suppose β starts the negotiation and gives up $G_{(0,2)}^\beta = C_5$ with $P_3 \sqsubseteq_{\mathcal{O}} G_{(0,2)}^\beta$, that is, giving up C_5 preference P_3 will not be satisfied anymore. She reformulates her request as $D_1 = \bigsqcap_{i=1..4,6..} C_i$ and sends it to σ. Notice that since $P_3 \sqsubseteq_{\mathcal{O}} G_{(0,2)}^\beta$, the global utility of β decreases to $u_1^\beta = \sum_{k=1..2,4..} u(P_k)$. Now, σ is able to validate if β really changed her request in round 0, to do so, σ computes $\langle G_1^\beta, K_1^\beta \rangle$ solving a concept contraction problem w.r.t. the new demand D_1 and checks if G_1^β is more general than G_0^β. In formulas, σ checks if $G_0^\beta \sqsubseteq_{\mathcal{O}} G_1^\beta$ holds, in case of positive answer, then σ knows that β did not lie and he continues the negotiation. Otherwise he may decide to leave the negotiation (conflict deal) or ask β to reformulate her counteroffer. If the negotiation continues, σ computes his conflicting information w.r.t. to D_1 and rewrites S as

$S = G_1^\sigma \sqcap K_1^\sigma$ where $G_1^\sigma = \bigsqcap_{j=1}^{m} G_{(1,j)}^\sigma$. Again, for each $G_{(1,j)}$ there exists a C_j in the normalized version of S. Hence, if σ decides to concede $G_{(1,j)}$, his global utility decreases proportionally to the utility of P_h to which $G_{(1,j)}$ belongs to. Once σ sends his counteroffer to β, this latter is able to check if σ lied. The process ends when one of the following two conditions holds:
1. the global utility of an agent is lower than its **disagreement threshold**. In this case the negotiation terminates with a conflict deal.
2. there is nothing more to negotiate on and the global utility of each agent is greater than its disagreement threshold. In this case the negotiation terminates with an agreement. **The agreement** A **is computed** simply as $A = D_{last} \sqcap S_{last}$, where D_{last} and S_{last} are the request and the offer in the last round.
Since users can express a utility value also on bundles, whenever they concede an issue as the **minimum concession** (in term of minimum global utility decrease), the set of all the bundles in which the issue is present has to be taken into account. They choose based on the utility of the whole set. For instance, suppose the buyer sets the following preferences on a `ClosedWardrobe`: $P_1 = \forall$`basicShape.SlidingDoors`; $P_2 = \leq_{210}$ `height`; $P_3 = \forall$`basicShape.SlidingDoors` $\sqcap \forall$`shutters.Glass` with the following utilities: $u^\beta(P_1) = 0.1$, $u^\beta(P_2) = 0.4$ and $u^\beta(P_3) = 0.5$. At the n-th step the conflicting information is:

$$G_n^\beta = \forall \texttt{basicShape.SlidingDoors} \sqcap \leq_{210} \texttt{height}$$

Hence, β can concede whether $\forall$`basicShape.SlidingDoors` or $\leq_{210}$ `height`. If she concedes $\forall$`basicShape.SlidingDoors` then her global utility decreases of $u^\beta(P_1) + u^\beta(P_3) = 0.6$, while conceding $\leq_{210}$ `height` her utility decreases of $u^\beta(P_2) = 0.4$ only. In this case the **minimum concession** is $\leq_{210}$ `height`.
Algorithm 1 defines the behavior of agents during a generic n-th round of the negotiation process. We present only the algorithm related to β's behavior since the behavior of σ is dual w.r.t. β's one.
1-4 If there is nothing in conflict between the old D_{n-1} and just-arrived S_n, then there is nothing more to negotiate on: the agreement is reached and returned.
5-11 If β discovers that σ lied on his concession, then β decides to exit the negotiation and terminates with a conflict deal. If we want β ask σ to concede again it is straightforward to change the protocol to deal with such a behavior.
13-15 If after the minimum concession, the utility of β is less than her **disagreement threshold**, then the negotiation ends with a conflict deal.

4 System Architecture

Main actors of our framework are RFID transponders clung to objects, an RFID reader integrated in a hotspot able to extract tags data and wireless devices interacting with hotspot via Bluetooth. Figure 2 shows the structure of the proposed framework, which is based on a two-level architecture that embeds semantic-enhanced variants of both RFID EPCglobal standard [14] and Bluetooth Service Discovery Protocol [13]. The first one is exploited to interconnect readers and tags dipped in the environment (the left hand side in Figure 2), the second one enables a communication between a reader

if $D_{n-1} \sqcap S_n \not\sqsubseteq_{\mathcal{O}} \perp$ **then**
  agreement A reached;
  return $A = D_{n-1} \sqcap S_n$;
end
if $n > 0$ **then**
  compute $\langle G_n^\sigma, K_n^\sigma \rangle$ from D_{n-1} and S_n;
  if $G_{n-1}^\sigma \not\sqsubseteq_{\mathcal{O}} G_n^\sigma$ **then**
    σ lied;
    conflict deal: **exit**;
  end
end
compute minimum concession $G_{(n-1,i)}^\beta$;
if $u_{n-1}^\beta < t^\beta$ **then**
  conflict deal: **exit**;
end
formulate D_n deleting $G_{(n-1,i)}^\beta$ from D_{n-1};
send D_n to σ;

Algorithm 1: The behavior of β at step n

and a mobile host acting as requester deployed in the wireless context (the right hand side in Figure 2). The hotspot plays a fundamental role in the whole discovery archi-

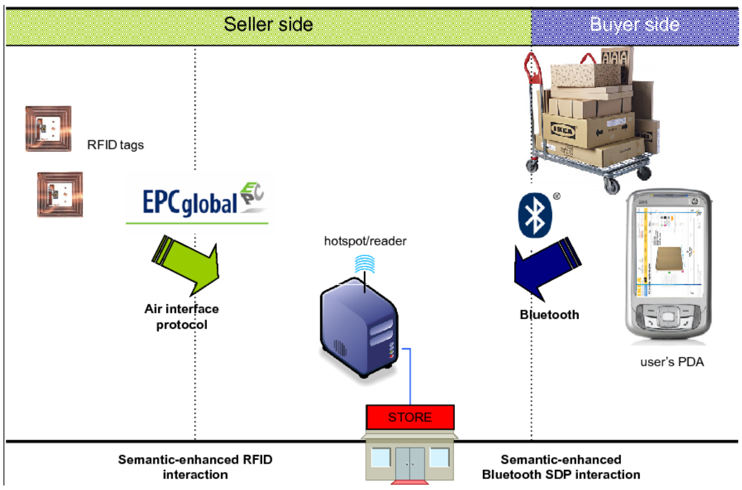

Fig. 2. Actors and interaction in the proposed framework

tecture as it: (i) collects descriptions and data-oriented parameters (such as price for example) referred to tags in its radio range; (ii) accepts and replies to requests received via Bluetooth taking into account resource descriptions extracted from transponders. Basically, the agreement between the buyer and seller wireless agents is reached after a progressive interaction. The negotiation protocol described above, can be seen as a common application layer interconnecting the seller and buyer side of Bluetooth SDP hosts with the final purpose of attempting a possible agreement between them. On the other hand, the seller agent also refers to objects in its radio range via the semantic-based EPCglobal protocol in order to retrieve products descriptions whose annotation are exploited in the negotiation process. In what follows basic features of the architecture layers are briefly recalled. Some details have been omitted for the sake of brevity. For a wider argumentation the reader is referred to [5].

Semantic-based Bluetooth Service Discovery. In [13] a framework has been proposed allowing to manage knowledge-based discovery of resources, by integrating a semantic micro-layer within the OSI Bluetooth stack at application level preserving legacy applications. Unused classes of Universally Unique Identifiers (UUIDs) in the original standard were exploited to unambiguously mark ontologies thus thinking up so called OUUIDs as Ontology Universally Unique IDentifiers. The fundamental assumption is that each resource is semantically annotated. A resource provider stores annotations within resource records, labelled with unique 32-bit identifiers. Each record basically consists of a list of resource attributes: in addition to the OUUID, there are a $ResourceName$ (a human-readable name for the resource), a $ResourceDescription$ (expressed using DIG syntax) and a variable number of $ResourceUtilityAttr_i$ attributes, *i.e.*, numerical values used according to specific applications. By adding some SDP Protocol Data Units (PDUs) to the original standard (exploiting not used PDU ID), further semantic enabled discovery functionalities have been introduced. Anyway, the overall interaction is based on the original SDP in Bluetooth. No modifications have been made to the original structure of transactions whose basic parameters, data structures and functions, have been completely saved.

Semantic-based EPCglobal RFID standard. We refer to RFID transponders compliant with EPCglobal standard of Class 1-Generation 2 UHF [15]. We assume the reader be familiar with basics of this technology. The practical feasibility of an advanced exploitation of RFID must take into account the severe bandwidth and memory limitations. From this standpoint two opposite questions have to be considered. First of all, due to technological advances and growing demand, passive RFID tags with greater memory amounts are expected to be available soon. On the other hand, XML-based ontological languages like OWL (http://www.w3.org/TR/owl-features/) and DIG (http://dl.kr.org/dig/) are far too verbose for a direct storage on RFID tags. In order to enable the semantic enhancements of RFID, the EPCglobal air interface protocol must provide read/write capabilities for semantically annotated product descriptions w.r.t. a reference ontology, along with additional data-oriented attributes (contextual parameters). Neither new commands nor modification to existing ones have been introduced in the proposed approach. Moreover, a mechanism is clearly required to distinguish semantic enabled tags from standard ones, so that semantic based applications can exploit the new features without interfering with legacy ones. In order to accomplish this coexistence, we extend the memory organization of tags compliant with the above standard. We exploit two bits in the EPC tag memory area currently reserved for future purposes. The first one –at 15_{hex} address– is used to indicate whether the tag has a user memory (bit set) or not (bit reset). The second one –at 16_{hex} address– is set to mark semantic enabled tags. In this way, a reader can easily distinguish semantic based tags by means of a *SELECT* command with proper values for reference parameters [14]. The following inventory step –which proceeds in the standard fashion– will skip "non-semantic" tags. The EPC standard requires the content of TID memory up to $1F_{hex}$ bit is fixed. TID bank can be extended to store optional information, generally consisting of tag serial number or manufacturer data. Hence we use the TID memory area starting from 100000_2 address to store a 128-bit OUUID labeling ontology the tag description refers to. In order to retrieve the OUUID stored within a tag, a reader will simply exploit a *READ* command with proper parameters. Contextual parameters (whose meaning may depend on the specific application) are stored within the User memory bank of the tag. There, we also store the semantically annotated description of the product the tag is clung to (opportunely compressed). An RFID reader can perform extraction and storing of a description from/on a tag by means of one or more *READ* or *WRITE* commands, respectively. Both commands are obviously compliant with the RFID air interface protocol.

5 An illustrative example: looking for a new wardrobe in a furniture store

To clarify our approach we present a simple example case study. Let us imagine a user entering a furniture store wishing to buy a new wardrobe. She hopes to find "*a closed wardrobe made from birch, with sliding doors equipped with glass shutters, a number of drawers between 3 and 5, at most two wire baskets, a shoe box organizer in a textile material, between 230 and 260 meters high and with rails for hanging clothes.*". Hence, she uses her PDA to run the *house planner* application. Then she starts to drag and drop components until she composes such a configuration on her PDA and sets utility values. Then her agent β starts a negotiation, while she hopes to find, if not the dreamed wardrobe, at least a good compromise.

On the other side, consider, among others, the following configuration offered by one of the m seller agents σ " *a closed wardrobe made from oak, with hinged doors, equipped with glass shutters, 4 drawers and 2 wire basket, equipped with Tidy shoe, a shoe box organizer, 236 meters high and with add on rails (basic rail plus tie rack and trouser hanger).*". As we have seen in Section 3 to each piece of information (component) it is associate a **utility value**, expressing relative importance of components. Agents also indicate a **threshold**, to rule out agreements less preferable than a conflict deal. Both the request D and the offer S can be formalized as[8]:

[8] `ClosedWardrobe` $\equiv$ `Wardrobe` $\sqcap \exists$`material` $\sqcap \exists$`basicShape` $\sqcap \exists$`shutters` is the object of the negotiation on which users express preferences, for that reason it is not part of the negotiation process.

$P_1^\beta = \forall \texttt{basicShape.SlidingDoors}$ $\quad u^\beta(P_1^\beta) = 0.35$
$P_2^\beta = \forall \texttt{material.Birch}$ $\quad u^\beta(P_2^\beta) = 0.2$
$P_3^\beta = \geq 3\texttt{drawers} \sqcap \leq 5\texttt{drawers} \sqcap \leq 2\texttt{WireBasket}$ $\quad u^\beta(P_3^\beta) = 0.05$
$P_4^\beta = \forall \texttt{shutters.Glass}$ $\quad u^\beta(P_4^\beta) = 0$
$P_5^\beta = \exists \texttt{organizer} \sqcap \forall \texttt{organizer.}(\texttt{ShoeBox} \sqcap \exists \texttt{material} \sqcap \forall \texttt{material.Textile})$ $\quad u^\beta(P_5^\beta) = 0.05$
$P_6^\beta = \exists \texttt{complement} \sqcap \forall \texttt{complement.Basic\_clothes\_rail}$ $\quad u^\beta(P_6^\beta) = 0.1$
$P_7^\beta = \geq_{230} \texttt{height} \sqcap \leq_{260} \texttt{height}$ $\quad u^\beta(P_7^\beta) = 0.25$
$t^\beta = 0.6$

$P_1^\sigma = \forall \texttt{basicShape.HingedDoors}$ $\quad u^\sigma(P_1^\sigma) = 0.3$
$P_2^\sigma = \forall \texttt{material.Oak}$ $\quad u^\sigma(P_2^\sigma) = 0.2$
$P_3^\sigma = = 4\texttt{drawers} \sqcap = 2\texttt{WireBasket} =_{236} \texttt{height}$ $\quad u^\sigma(P_3^\sigma) = 0.25$
$P_4^\sigma = \exists \texttt{complement} \sqcap \forall \texttt{complement.Add\_on\_clothes\_rail}$ $\quad u^\sigma(P_4^\sigma) = 0.05$
$P_5^\sigma = \forall \texttt{shutters.Glass}$ $\quad u^\sigma(P_5^\sigma) = 0.05$
$P_6^\sigma = \forall \texttt{organizer.} \sqcap \forall \texttt{organizer.TidyShoeBox}$ $\quad u^\sigma(P_6^\sigma) = 0.15$
$t^\sigma = 0.5$

K and G are computed for both β and σ.

K_0^β = $\forall \texttt{shutters.Glass} \sqcap \geq 3\texttt{drawers} \sqcap \leq 5\texttt{drawers} \sqcap \leq 2\texttt{WireBasket} \sqcap \exists \texttt{complement} \sqcap \forall \texttt{complement.Basic\_clothes\_rail} \geq_{230} \texttt{height} \sqcap \leq_{260} \texttt{height}$
G_0^β = $\forall \texttt{basicShape.SlidingDoors} \sqcap \forall \texttt{material.Birch} \sqcap \exists \texttt{organizer} \sqcap \forall \texttt{organizer.}(\texttt{ShoeBox} \sqcap \exists \texttt{material} \sqcap \forall \texttt{material.Textile})$

K_0^σ = $\forall \texttt{shutters.Glass} \sqcap = 4\texttt{drawers} \sqcap = 2\texttt{WireBasket} \sqcap =_{236} \texttt{height} \sqcap \exists \texttt{complement} \sqcap \forall \texttt{complement.Add\_on\_clothes\_rail}$
G_0^σ = $\forall \texttt{basicShape.HingedDoors} \sqcap \forall \texttt{material.Oak} \sqcap \exists \texttt{organizer} \sqcap \forall \texttt{organizer.TidyShoeBox}$

Now suppose that by coin tossing, σ moves first. He starts giving up the component with the minimum utility (the TidyShoe box), which is his minimum concession. Then he computes his utility and, since it is greater than the threshold value, decides to go on with the negotiation process. In the following step we have:

K_1^β = $\forall \texttt{shutters.Glass} \sqcap \geq 3\texttt{drawers} \sqcap \leq 5\texttt{drawers} \sqcap \leq 2\texttt{WireBasket} \sqcap \exists \texttt{complement} \sqcap \forall \texttt{complement.Basic\_clothes\_rail} \geq_{230} \texttt{height} \sqcap \leq_{260} \texttt{height} \sqcap \exists \texttt{organizer} \sqcap \forall \texttt{organizer.}(\texttt{ShoeBox} \sqcap \exists \texttt{material} \sqcap \forall \texttt{material.Textile})$
G_1^β = $\forall \texttt{basicShape.SlidingDoors} \sqcap \forall \texttt{material.Birch} \sqcap \exists \texttt{organizer}$

K_1^σ = K_0^σ
G_1^σ = $\forall \texttt{basicShape.HingedDoors} \sqcap \forall \texttt{material.Oak}$

At this point, β gives up $\forall \texttt{material.Birch}$ which is the preference with the minimum utility. The protocol continues until agents reach logical compatibility. A final agreement could then be:

A = $\texttt{ClosedWardrobe} \sqcap \forall \texttt{basicShape.SlidingDoors} \sqcap \forall \texttt{material.Oak} \sqcap \forall \texttt{shutters.Glass} \sqcap = 4\texttt{drawers} \sqcap = 2\texttt{WireBasket} \sqcap =_{236} \texttt{height} \sqcap \exists \texttt{organizer} \sqcap \forall \texttt{organizer.}(\texttt{ShoeBox} \sqcap \exists \texttt{material} \sqcap \forall \texttt{material.Textile}) \sqcap \exists \texttt{complement} \sqcap \forall \texttt{complement.Add\_on\_clothes\_rail}$,

with corresponding utilities $u^\beta = u^\beta(P_1^\beta) + u^\beta(P_3^\beta) + u^\beta(P_5^\beta) + u^\beta(P_6^\beta) + u^\beta(P_7^\beta) = 0.8$ for β, and $u^\sigma = u^\sigma(P_2^\sigma) + u^\sigma(P_3^\sigma) + u^\sigma(P_4^\sigma) + u^\sigma(P_5^\sigma)$ = 0.55 for σ, both above the thresholds.

6 Conclusion

We have presented a framework for semantic-based multi-issue bilateral negotiation grounded in a mobile context. Semantic annotations describing products and good features are compressed and properly stored in RFID tags. Semantic-enhanced versions of both RFID and Bluetooth standards are so exploited to enable an infrastructure-less interaction between buyer and seller wireless negotiation agents. Hence, seller configurations are proposed based on concrete product availability and the buyer can interrogate it from everywhere aided by an intuitive GUI. Non standard inference services and utility theory have been exploited in the negotiation process to reach a suitable agreement. The complete framework has been implemented and it is currently being evaluated in the case study scenario.

Acknowledgments

The authors thank Tommaso Di Noia for fruitful discussions, ideas and suggestions. This research is partially sponsored by *Distributed Production as Innovative System (DIPIS)* and *Telecommunication Facilities and Wireless Sensor Networks in Emergency Management* projects.

References

1. F. Baader, D. Calvanese, D. Mc Guinness, D. Nardi, and P. Patel-Schneider. *The Description Logic Handbook*. Cambridge University Press, 2002.
2. S. Bechhofer, R. Möller, and P. Crowther. The DIG Description Logic Interface. In *DL'03*, 2003.
3. Bluetooth. http://www.bluetooth.com.
4. T. Di Noia, E. Di Sciascio, and F. Donini. Semantic Matchmaking as Non-Monotonic Reasoning: A Description Logic Approach. *JAIR*, 29:269–307, 2007.
5. T. Di Noia, E. Di Sciascio, F. Donini, M. Ruta, F. Scioscia, and E. Tinelli. Semantic-based bluetooth-rfid interaction for advanced resource discovery in pervasive contexts. *IJSWIS*, 4(1):50–74, 2008.
6. U. Endriss. Monotonic concession protocols for multilateral negotiation. In *Proc. of AAMAS'06*, pages 392–399.
7. EPCglobal. http://www.epcglobalinc.org.
8. P. Gärdenfors. *Knowledge in Flux: Modeling the Dynamics of Epistemic States*. Bradford Books, MIT Press, Cambridge, MA, 1988.
9. S. Kraus. *Strategic Negotiation in Multiagent Environments*. The MIT Press, 2001.
10. A. Ragone, T. Di Noia, E. Di Sciascio, and F. M. Donini. Alternating-offers protocol for multi-issue bilateral negotiation in semantic-enabled marketplaces. In *ISWC'07*. 2007.
11. J. Rosenschein and G. Zlotkin. *Rules of Encounter*. MIT Press,, 1994.
12. A. Rubinstein. Perfect equilibrium in a bargaining model. *Econometrica*, 50:97–109, 1982.
13. M. Ruta, T. Di Noia, E. Di Sciascio, and F. Donini. Semantic-Enhanced Bluetooth Discovery Protocol for M-Commerce Applications. *IJWGS*, 2(4):424–452, 2006.
14. M. Ruta, T. Di Noia, E. Di Sciascio, F. Scioscia, and G. Piscitelli. If objects could talk: A novel resource discovery approach for pervasive environments. *International Journal of Internet and Protocol Technology (IJIPT), Special issue on RFID: Technologies, Applications, and Trends*, 2(3/4):199–217, 2007.
15. K. Traub. EPCglobal Architecture Framework. Technical report, EPCglobal, July 2005.