A Semantic Framework for Priority-based Service Matching in Pervasive Environments

Ayomi Bandara[1], Terry Payne[1], David De Roure[1], and Tim Lewis[2]

[1] University of Southampton, Southampton, UK
{hmab02r, trp, dder}@ecs.soton.ac.uk,
[2] Telecommunications Research Laboratory, Toshiba Research Europe Ltd, Bristol, UK
Tim.Lewis@toshiba-trel.com

Abstract. The increasing popularity of personal wireless devices has raised new demands for the efficient discovery of heterogeneous devices and services in pervasive environments. The existing approaches such as Jini [1], UPnP [8], etc., describe services at a syntactic level and the matching mechanisms in these approaches are limited to syntactic comparisons based on attributes or interfaces. In order to overcome the limitations in these approaches, there has been an increased interest in the use of semantic description and matching techniques to support effective service discovery. This paper proposes a semantic matching approach which facilitates the discovery of device-based services in a pervasive environment; the approach provides a ranking facility that orders services according to their suitability and also considers priorities placed on individual requirements in a request during the matching process. The evaluation studies have shown that the matcher results correlate reasonably well with human judgement.

1 Introduction

With the current trends in the electronic world, devices of increasing heterogeneity are being introduced into pervasive environments. These vary from headsets to mobile phones, PDAs and laptops; each of which offer a plethora of services. This has raised new demands for the discovery of devices and their services in a dynamically changing environment. The existing device description and discovery solutions (such as UPnP [8], Jini [1], etc.), characterise the services by using predefined service categories and fixed attribute value pairs; also the matching mechanisms in these approaches are limited to string comparisons or key-word based searches. Since discovery is not supported by any form of inferencing, such approaches will be unable to identify a match between logically equivalent services that have syntactically different descriptions.

With recent trends in the Semantic Web, there has been an increased interest in the use of ontologies to describe services and the use of logical reasoning mechanisms to support service matching. The advantage of such frameworks include the ability to extend and adapt the vocabulary used to describe services and to harness the inferential benefits of logical reasoning over such descriptions. Recently, a number of semantic matching approaches have been developed (targeted at different domains), which try to address various limitations in traditional discovery techniques in order to come up with a pragmatic solution to meet the challenges in the service discovery arena.

The work described in [4] and [2] propose Semantic Matching approaches for pervasive environments. Both these approaches use ontologies to describe the services and a Prolog-based reasoning engine to facilitate the semantic matching. They provide 'approximate' matches if no exact match exists for the given request. However, the criteria used for judging the 'closeness' between the service advertisements and the request is not clear from the literature. In both these approaches, the matching process does not perform any form of match ranking. There have also been a number of efforts that use description logic (DL) based approaches for semantically matching web services. For example the matchmaking framework presented in [5] uses a DAML-S based ontology for describing the services. A DL reasoner has been used to compute the matches for a given request. The matches are classified into one of its five “degrees of match” (namely Exact, Plug-In, Subsume, Intersection and Disjoint) by computing the subsumption relationship of the request description w.r.t. all the advertisement descriptions. No ranking is performed in the matching process, although the match class suggesting the ‘degree of match’ gives an indication of how ‘good’ a match is.

The above mentioned semantic matching approaches provide important directions in overcoming the limitations present in traditional syntactic approaches to service discovery. However, these solutions still have limitations and overlooked issues that need to be addressed; particularly these approaches do not have an effective ranking criterion to facilitate the ordering of potential matches, according to their suitability to satisfy the request under concern. Also these approaches do not facilitate priorities/ weights on the individual requirements and thus the matching process will consider that all requirements have equal priority.

In this paper we propose a pragmatic approach, that facilitates the effective matching of resource requests and advertisements in pervasive environments. This semantically compares the request against the available services and provides a ranked list of most suitable services. The rank will indicate the appropriateness of a service to satisfy a given request and thus provides a valuable decision support for the service seeker, in selecting the most suitable service. The matching process also considers the priorities/ weights on the individual requirements of a request; this helps capture any context dependencies involved and subjective preferences of the resource seeker. This is an important facility in any matching system since in many practical scenarios certain service requirements will be more important than others and therefore priority-based matching can produce results that can better meet the users expectations. An OWL ontology is used to describe the services and a Description Logic reasoner is used to support the background reasoning tasks in the matching process. The remainder of this paper is organised as follows: Section 2 discusses the motivation behind the proposed matching framework and identifies the requirements of a pragmatic approach for matching pervasive resources. Section 3 briefly describes the methodology behind the matchmaking framework. Section 4 discusses the prototype implementation of the service matching approach in a pervasive scenario and presents the initial evaluation results. The concluding remarks and future directions of this work are discussed in section 5.

2 Motivating Factors

There are several desirable properties that must be present in an effective service matching framework. In this section we discuss these along with the motivating reasons behind them.

Semantic Description and Matching: An ontological approach for the description of services coupled with reasoning mechanisms to support service discovery and matching enables logical inferencing over these descriptions and therefore offers several benefits over the traditional syntactic approaches. It is often the case, that the service providers usually describe devices in terms of lower-level properties, and the service seekers or clients usually prefer to describe service requests using more abstract or higher level concepts. Semantic matching approaches supported by logical reasoning mechanisms will be able to identify a match between logically equivalent services that have syntactically different descriptions and therefore can offer flexibility in how the service advertisements and requests are described.

Match Ranking: Ranking refers to the ordering of the available advertisements in the order of their suitability to satisfy the given request. In the absence of an exact match, a requester might be willing to consider other advertisements that are closer to the request and thus the ranking will be useful in gaining an understanding of the appropriateness of the advertisement. Most existing matchmaking solutions do not have an effective criterion to rank the available services according to their suitability. Providing a ranking mechanism that will rank the advertisements on the basis of how well it satisfies the properties specified in the request, is one of the main objectives behind the proposed matching framework.

Approximate Matching: Offering approximate matching, is one of the core objectives of semantic matching. i.e. services that deviate from the request in certain aspects should not be discarded but must be ranked or classified appropriately to indicate the suitability. In the current semantic matching approaches [5], [6], the suitability of the advertisement have been determined using subsumption reasoning based on the taxonomic relation between the concepts. However we argue that subsumption reasoning alone is not sufficient in determining similarity for the purpose of resource matching. Depending on the concept involved, reasoning based on the taxonomy alone, will not accurately reflect the similarity between concepts. For example, consider the concept *Processor*; assume there is a request for a computer that has processor *Pentium4* and advertisements of computers with processors *Pentium3* and *Pentium1*. Both *Pentium3* and *Pentium1* will be disjoint from the originally requested concept of *Pentium4*, but a requester will consider *Pentium3* to be a better match than *Pentium1* and will be ranked higher. Thus the type of attribute involved in the individual requirement of a request will have to be considered in approximating and ranking of advertisements. Section 3 describes the types of attributes and the approach taken in judging the similarity between them for the purpose of ranking.

Priority-based Matching The current matchmaking researches do not consider any priorities or preferences that a user/agent may be having with respect to various attributes or properties of a service (except in [6]). In many practical scenarios certain requirements/ attributes in a request will be more important than others, either due to the context involved or the subjective preferences of the user. In such cases, facilitating priority-handling in the matching process will produce match results that are more relevant and suitable for the context involved. For example, consider two users looking for a printer; considering the time to service and quality properties of the printer, both may want to take the printouts as quick as possible and with the highest quality possible. But a user who wants to rush off to a meeting in the next five minutes will definitely be more concerned about the time factor and be willing to compromise on quality. But a user who is working at leisure, will not mind waiting in order to obtain a more quality print. Thus in cases like this it is vital to consider the importance placed on the properties of the service by a user, by taking into account the priorities of the attributes.

Mandatory requirements or strict matching requirements have to be considered when, the resource seekers requires a certain individual property requirement in a request, to be strictly met by any potential resource advertisement; i.e. they will not want to consider any advertisements that will have even a minor deviation, with respect to that property. For example consider the case where a resource seeker needs to utilise a computer to run an application which will only run on the operating system WindowsXP; he will specify the operating system requirement in the request along with the other desirable characteristics. In the context involved the operating system property is a mandatory requirement and hence the resource seeker will not need to consider any available computers which deviates with respect to the operating system requirement (no matter how good it is with respect to other attributes). Hence this needs to be taken into account in the matching process and the available resources that deviate from this strict requirement must not be included in the result set (or ranked as the worst matches). Priority matching is applicable when a resource seeker has varying importance placed on the individual property requirements of the request. Strict matching can in fact be considered as a specific case of priority matching.

This factor will be taken into account in the proposed work by giving a service requester the option of placing priorities/ weights on the specified attributes of the service request. These weights will be considered in the matching process during the ranking of advertisements.

3 The Semantic Matching Methodology

3.1 Service Description

In order to facilitate the use of logical reasoning over the service descriptions, we describe the requests and advertisements in the Web Ontology Language (OWL).

A request will typically consist of several sub requirements to be satisfied. Each individual requirement will specify: the description of the requirement (which is the resource characteristic the resource seekers expect in a resource, for the their needs to be satisfied) and the priority or weight of that individual requirement, which will be a decimal value that indicates the relative importance of the particular requirement. The priority value can also be used to indicate if the requirement considered is a mandatory requirement; i.e. if the requirement should be strictly satisfied in an advertisement for the requester to consider it as a potential match. The description of an individual requirement will include the property or attribute the requesters are interested in and the ideal value desired.
The request will take the form of:
$Request \equiv (Req_1) \sqcap (Req_2) \sqcap \ldots \sqcap (Req_n)$
where Req_i is an individual requirement. The requirement in turn can take the form of:
$Req \sqsubseteq (= 1 hasDescription.RD) \sqcap (= 1 hasPriority.PriorityValue)$
where RD is the requirement description, which can be either a named concept or an existential restriction of the form, $\exists p.C$ where p is a role and C is a named concept or a complex concept. For describing each RD, an ontology that describes the services in the domain concerned can be used. The $PriorityValue$ indicates the relative importance of the individual requirement in the request. This is a decimal value defined between 0 and 1. In addition, to indicate that the requirement is a mandatory requirement that must be strictly met in any potential match, the $PriorityValue$ is defined as 2. The resource seeker must pick the appropriate $PriorityValue$ (according to these pre-defined values) for each individual requirement, to indicate its relative importance.

The resource provider will specify all the relevant characteristics of the available resource in the resource advertisement. The advertisement can take the form of:
$Advertisement \equiv (r_1) \sqcap (r_2) \sqcap \ldots \sqcap (r_n)$; where r_i is either a named concept or an existential restriction describing a characteristic of the resource.

3.2 Ranking Process

As mentioned previously a request will consist of a number of individual requirements along with their priority values. The presence of any mandatory requirements that must be fully satisfied by any potential match will also be indicated by using the appropriate priority value as mentioned in 3.1. In the matching process, the available resource will be checked to see if each mandatory individual requirement (RD) is satisfied in the advertisement description. If the mandatory requirement(s) are met, then the advertisement will be evaluated through approximate matching.

In approximate matching, the available resources should be evaluated according to how well it satisfies each individual requirement specified in a request; i.e. the matching engine should quantify the extent to which each individual requirement description (RD) is satisfied by the resource advertisement. For this, the matching engine will check how similar the advertisement is with respect to each non-mandatory requirement (RD) specified in the request; the similarity will be determined depending on the semantic deviation of the expected value in request and the available value in advertisement for the same requirement, and a score will be assigned accordingly ($Score_{\,i}$).

Each characteristic specified in the request (RD) can be a named concept($C_{\,R}$) or an existential restriction ($\exists p.C_R$). If it is a named concept, similarity will be compared between the corresponding concepts in request and advertisement (*Similarity*($C_{\,R}, C_A$)). If it is an existential restriction, the corresponding existential restriction(s) will be found in the advertisement ($\exists p.C_A$) and the similarity will be compared between the corresponding concepts in request and advertisement. If it is a composite concept the similarity will be judged recursively. The score ($Score_i$) for each individual characteristic in the request will be assigned depending on this similarity.

The degree of similarity between concepts will be determined depending on the type of concept or attribute involved; determination of similarity between concepts will be discussed later in this section. A score ($Score_i$) is assigned for each sub-requirement (RD) specified in the request. The score for the advertisement (match score) will be determined by using the weighted average of these individual scores (the weight will be the corresponding priority value of each individual requirement).
$MatchScore = \sum_{i=1}^{n} w_i.Score_i \div \sum_{i=1}^{n} w_i$
where w_i and $Score_i$ is the priority value and the score of the sub-requirement $RD_{\,i}$. The overall score for the advertisement provides an indication of how good the advertisement is in satisfying the given request. The score for an advertisement will in turn be used as the basis for ranking; the highest score will receive the highest rank and so on.

The attributes in a resource description are categorized into three types, for the purpose of approximating and judging similarity within individual requirements. These are:

Type 1 Attributes: Attributes involving symbolic concepts for which judging similarity using the taxonomic relation is sufficient. In this case the matching engine will make use of a reasoner to judge the similarity by subsumption relation. Say the advertisement specifies that it has concept C_A as its value for a particular property or attribute and request specifies it has concept C_B. When a description logic reasoner is used to find the subsumption relation between these two concepts, it could fall under one of four types. These types, and the scores assigned are represented in Table 1. In the case where C_A is a super concept of C_B, the score assigned must be a value between 0 and 1; the ideal value of t can be determined through a human user study. However, for the purpose of this implementation of the matching system we use a value of 0.6.

Type 2 Attributes: Attributes involving symbolic concepts for which judging similarity using the taxonomic relation is not sufficient. For example *Processor* and *Display Technology* concepts fall into this category. Hence in our work, if we wanted to find similarity between different Processor Types for example, the features/ properties of the Processors such as Clock Speed, Manufactured-By etc. will have to be used in measuring the similarity. However measuring similarity between concepts is out of the scope of the current work and we assume that the knowledge of similarities between such concepts is available to us (measured by using an available similarity measurement approach such as [7], and available as domain knowledge in the ontology) and can be accessed by the matchmaking process.

Type 3 Attributes: When the attributes are numeric (integer or decimal) the degree of similarity between the requested and advertised attribute values must be determined depending on the level of deviation. This deviation can be determined by using either a fuzzy membership function or by computing the percentage deviation from the original requested value.

Subsumption Relation	**Similarity Score**
C_A and C_B refer to the same concept	1.0
C_A is a super concept of C_B	t (where $0 < t < 1$)
C_B is a super concept of C_A	1.0
C_A, C_B are not equivalent and do not have a subsumption relation	0.0

Table 1. Assignment of similarity scores when Subsumption Relation is considered.

4 Application of the Matching Framework in a Pervasive Context

The matching framework presented in the previous sections has been implemented in a pervasive context for matching of device based services. The service requesters seek to utilise specific devices and their services depending on their functionality. The advertisements and the individual requirements in a request are described using the Device Ontology presented in [3] (available at `http://www.ecs.soton.ac.uk/~hmab02r/DeviceOnt/DevOntology.owl`). This facilitates the description of features and functionalities of the devices and their services. The necessary ontologies were developed with the Protégé ontology editor. The matching engine was implemented in Java and the Pellet DL reasoner in combination with the Pellet-API is used to facilitate the necessary reasoning tasks during the matching process.

Figure 1 illustrates the architecture of the matching system. Once the matching system receives the OWL descriptions of the advertisements and request, it checks for the consistency of the descriptions. If they are consistent the matching process begins. Each advertisement is compared with the request using the matching mechanism presented before and depending on the suitability of the advertisement to satisfy the request a score is assigned to the advertisement. Once all the advertisements are compared and scored, the advertisements are ranked on the basis of the score they have received. Then the system returns the advertisements along with their rankings.

As emphasised in Section 2, semantic approaches to service discovery can clearly provide many benefits over syntactic approaches. However, we have to bear in mind the fact that certain resources in pervasive environments (small mobile devices such as mobile phones and PDA's), are heavily constrained in terms of computing power and therefore the standard semantic web tools and technologies can be too heavy-weight for such resources. Hence a feasible architecture has to be chosen for the discovery process, while facilitating the use of semantic descriptions and reasoning mechanisms to provide effective description and matching of services. For example, the matching process could always run centrally on the network and the devices could communicate through the network as appropriate.

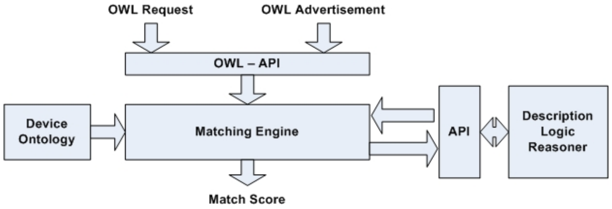

Fig. 1. The Matching System

4.1 Matching Example

We illustrate the application of the proposed matching approach with the use of the following example. We assume a scenario where a user in a pervasive environment seeks a printer with certain characteristics. The request concerned is a *Colour*, *Laser* Printer, that can print on the paper size *A2*. We also assume that the Paper Size *A2* is the most important (highest priority) attribute under the context involved and that the other two attributes (the Print Technology being *Laser* and the *Colour* printing capability) are of lesser priority. The priority values or weights for the three attributes are assigned as 0.6 for the paper size attribute and 0.2 for both the other attributes (the printing colour and printer technology). Since the request is for a *Printer*, this is stated as a mandatory requirement with priority value 2.0. The request will be described in description logic notation as:

$$\begin{array}{l}
Request \sqsubseteq \\
\exists\ hasRequirement\ (Requirement \sqcap \exists hasPriority. = 2.0\ \sqcap \\
\exists hasRequirementDescription.RD1) \sqcap \\
\exists\ hasRequirement\ (Requirement \sqcap \exists hasPriority. = 0.6\ \sqcap \\
\exists hasRequirementDescription.RD2) \sqcap \\
\exists\ hasRequirement\ (Requirement \sqcap \exists hasPriority. = 0.2\ \sqcap \\
\exists hasRequirementDescription.RD3) \sqcap \\
\exists\ hasRequirement\ (Requirement \sqcap \exists hasPriority. = 0.2\ \sqcap \\
\exists hasRequirementDescription.RD4)
\end{array}$$

$$RD1 \equiv ptr : Printer$$

$$\begin{array}{l}
RD2 \equiv \exists dev : hasHardwareDescription \\
(dev : HardwareDescription \sqcap \exists ptr : hasPaperSize\ .\ ptr : A2)
\end{array}$$

$$\begin{array}{l}
RD3 \equiv \exists dev : hasHardwareDescription \\
(dev : HardwareDescription \sqcap \exists ptr : hasPrintTechnology\ .\ ptr : Laser)
\end{array}$$

$$\begin{array}{l}
RD4 \equiv \exists dev : hasHardwareDescription \\
(dev : HardwareDescription \sqcap \exists ptr : hasPrintingColour\ .\ ptr : Colour)
\end{array}$$

For the purpose of this evaluation experiment we assume the availability of twelve advertisements (with varying characteristics) of which we include the descriptions of four. The available advertisements are:

$Advert1 \sqsubseteq\ ptr : Printer \sqcap \exists dev : hasHardwareDescription$
$(dev : HardwareDescription \sqcap \exists ptr : hasPaperSize\ .\ ptr : A2\ \sqcap$
$\exists ptr : hasPrintTechnology\ .\ ptr : Laser \sqcap \exists ptr : hasPrintingColour\ .\ ptr : BW)$

$Advert2 \sqsubseteq\ ptr : Printer \sqcap \exists dev : hasHardwareDescription$
$(dev : HardwareDescription \sqcap \exists ptr : hasPaperSize\ .\ ptr : A2 \sqcap$
$\exists ptr : hasPrintTechnology\ .\ ptr : Laser \sqcap \exists ptr : hasPrintingColour\ .\ ptr :$
$Colour)$

$Advert3 \sqsubseteq\ ptr : Printer \sqcap \exists dev : hasHardwareDescription$
$(dev : HardwareDescription \sqcap \exists ptr : hasPaperSize\ .\ ptr : A2 \sqcap$
$\exists ptr : hasPrintTechnology\ .\ ptr : Inkjet \sqcap \exists ptr : hasPrintingColour\ .\ ptr :$
$BW)$
$\vdots$
$Advert12 \sqsubseteq\ ptr : Printer \sqcap \exists dev : hasHardwareDescription$
$(dev : HardwareDescription \sqcap \exists ptr : hasPaperSize\ .\ ptr : A4 \sqcap$
$\exists ptr : hasPrintTechnology\ .\ ptr : Inkjet \sqcap \exists ptr : hasPrintingColour\ .\ ptr :$
$BW)$

Considering the attributes involved in this example: The Paper Size attribute is a Type-2 attribute and we assume that the similarity values between A2, A3 and A2, A4 are given as 0.6 and 0.25. Printer Technology attribute is also a Type-2 attribute and the similarity values between Laser and Inkjet is 0.7. Printing Colour is a Type-1 attribute where the concept Colour is defined as a subclass of the concept Black_White (since all colour printers can print black & white as well). Therefore considering Advert12, this satisfies the mandatory requirement of being a *Printer* and therefore will proceed through to the approximate matching process. This will get subscores of 0.25, 0.7 and 0.6 for the attributes of Paper Size, Printer Technology and Printing Colour. By considering the weighted average of these subscore values (using the corresponding priority values of the attributes as weights), the Advert will get a match score of .41. Similarly the other advertisements can be evaluated in the same way and by considering the match score, the advertisements could be ranked accordingly.

4.2 Evaluation

The effectiveness of the proposed matching approach is evaluated by comparing the results of the matching system with human perception. This is done by comparing the results obtained through the matching system with the rankings provided by domain users that rank the available resources in the same scenario.

We conducted a study involving human subjects to obtain the human rankings for this evaluation exercise. For each experiment, a scenario or use case (in a pervasive context) is devised that will involve a resource seeking situation where the seeker raises a query for a resource with certain property requirements. For each use case we construct a questionnaire which specifies: the device and the property/ functionality requirements that the resource seeker is interested in, the context that has given rise to the need of the device and the available devices and their properties. We hand out the questionnaire to the subjects involved and request them to assume that they are the resource seeker in the given context and rank the available devices specified, in the order they would consider them for utilising for the specified need. For each experiment, at least 10 subjects were involved and the rankings provided were averaged (to minimise the effects of subjective judgements) for the purpose of comparison with the matchmaker results. Each experiment was designed to test a different aspect of the matching framework.

To judge the degree of conformance of the match results to human perception, the matcher ranking is compared with the average human ranking (obtained for the same experiment) using graphical illustration of the plots and the measurement of standard deviation. It was generally observed in all the experiments that the matcher results were reasonably close to the average human ranking.

Due to space limitations, the detailed results of all the experiments in this evaluation exercise will not be presented in this paper. However, to give an indication of how well the matcher ranking conforms to human judgement, and to show that considering priorities on individual requirements during the matching process improves the effectiveness of the matching system, we present the results obtained for the example scenario discussed in Section 4.1. The average human ranking has been obtained for this scenario through a human study and compared with the results obtained from the proposed priority-based matcher. To illustrate how close the matcher ranking is to the average human ranking, we have computed the difference between the matcher rank and the average human rank and this is graphically illustrated in Figure 2; the y axis of the figure depicts the difference in the rankings. For the sake of comparison, the results were also obtained from the matcher when no priorities are considered (i.e. when all the requirements are assumed to have equal priority/ weight). The difference between this and the human rank is also computed and is illustrated in the Figure 2. Through the observation of these plots we can see that the matcher rank with priority consideration is closer to the human ranking than in the case where priority is not considered. The standard deviation between the matcher ranking (without priority consideration) and the average human ranking is 7.19%. The same value for the matcher rank with priority consideration is 3.35%. This shows that including the priorities in the individual requirements and considering them in the matching process will allow the matcher to produce results that better suit the context involved and that conform to human judgement.

5 Conclusions & Future Work

In this paper we have proposed a semantic approach that provides effective matching of resources in a pervasive environment. The approach facilitates the ranking of potential matches in the order of their suitability to satisfy the request, which aids the users of the matching system to identify the order in which they should consider the returned matches. The ranking mechanism overcomes the limitations present in matchers which uses subsumption reasoning alone. The matching framework also facilitates the specification of priorities in the service request and incorporates priority handling in the matching system; this helps to identify the relative importance of the individual requirements in a request and also to indicate whether certain requirements are mandatory and thus should be strictly met in any potential match. Hence the matching system can produce results that better suit the context involved and the subjective preferences of service seekers. The involvement of match ranking and the priority handling are both important and useful additions to the existing work on service matching.

We have implemented the proposed solution in a pervasive context and results have been obtained. The effectiveness of the solution has been evaluated through the use of

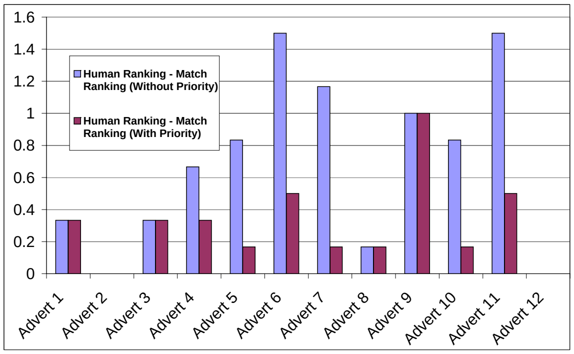

Fig. 2. *The Difference Between the Averaged Human Ranking and the Matcher Rankings*

a human user study and initial results indicate that the matcher results correlate well with human perception. As part of the future work of this research, we plan to formally evaluate the performance of the proposed approach to justify that any compromise in performance resulting from the involvement of an ontological description and reasoning mechanism, is outweighed by the benefits gained from semantic matching. Specifically we plan to investigate the scalability and the performance of the matching approach, in terms of the number of advertisements involved in the matching process and the size of the resource request (in terms of the number of individual requirements).

Acknowledgements: This research is funded and supported by the Telecommunications Research Laboratory of Toshiba Research Europe Ltd.

References

1. K. Arnold, B. OSullivan, R. W. Scheifler, and A. Wollrath J. Waldo. *The Jini Specification*. Addison-Wesley, 1999.
2. S. Avancha, A. Joshi, and T. Finin. Enhancing the bluetooth service discovery protocol. Technical report, University of Maryland Baltimore County, 2001.
3. A. Bandara, T.R. Payne, D. de Roure, and G. Clemo. An ontological framework for semantic description of devices (poster). In *3rd Int. Semantic Web Conference (ISWC 2004)*, 2004.
4. D. Chakraborty, F. Perich, S. Avancha, and A. Joshi. Dreggie: Semantic service discovery for m-commerce applications. In *Workshop on Reliable and Secure Applications in Mobile Environment, Symposium on Reliable Distributed Systems*, 2001.
5. L. Li and I. Horrocks. A software framework for matchmaking based on semantic web technology. In *Proc. of the Twelfth Int. World Wide Web Conference*, pages 331–339. ACM, 2003.
6. M. Paolucci, T. Kawamura, T.R. Payne, and K.P. Sycara. Semantic matching of web services capabilities. In *Int. Semantic Web Conference*, pages 333–347, 2002.
7. Amos Tversky. Features of similarity. *Psychological Review*, 84:327– 352, 1977.
8. Upnp, 2000. `http://www.upnp.org/download/UPnPDA10\_20000613.htm`.