Formal Aspects of Classifying and Selecting Business Services

(Extended Abstract)

Yathiraj B. Udupi
Cisco Systems Inc.
170 W. Tasman Dr.
San Jose, CA 95134
ybudupi@gmail.com

Munindar P. Singh
Department of Computer Science
North Carolina State University
Raleigh, NC 27695-8206
singh@ncsu.edu

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Multiagent systems*; H.1.0 [**Information Systems**]: Models and Principles—*General*; C.2.4 [**Computer-Communication Networks**]: Distributed Systems—*Distributed application*

General Terms

Design, Theory, Algorithms

Keywords

Matching, Service selection, Vectors

1. INTRODUCTION

This paper uses the term "service" for a service instance, and the term "agent" for an agent who provides or consumes a service. We consider real-life or business services (e.g., business process outsourcing, software development service involving human experts). We distinguish business from computational (e.g., Web or grid) services based on the fact that business services lack the typical input-output structure of computational services. For example, one can model a temperature service as one that takes a zipcode as input and produces the current temperature as output. By contrast, it would not help to model a software development service as one that takes a "business problem" as input and produces a suite of "software modules" as output. First, it is clearly beyond the scope of current practice to create formal classes or a type system of business problems and software modules. Second, business services are not invoked but are engaged, and would rarely take single-shot inputs and produce single-shot outputs. Third, business service providers would offer a continuum of *expertise* along which they can provide effective services. For example, a provider who is good at payroll management may also be able to provide retirement plan management, in contrast with the temperature service example above, which has no other function. Fourth, the selection of business services relies on the agents' evaluation of previous engagements.

Service selection inherently considers instances (runtime), not types (design-time). A service can be judged along many different dimensions in an empirical basis. Since agents are autonomous, it is important to select the services they provide based on their known performance along the various dimensions of interest. Services naturally fall into a taxonomy. For example, we know that ophthalmologists are physicians. We would prefer them for eye-related concerns over physicians in general.

Cite as: Formal Aspects of Classifying and Selecting Business Services, (Extended Abstract), Yathiraj B. Udupi, Munindar P. Singh, *Proc. of 8th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2009),* Decker, Sichman, Sierra and Castelfranchi (eds.), May, 10–15, 2009, Budapest, Hungary, pp. 1247–1248

Conventional approaches for selection are either empirical [1] or taxonomic, but not both. Existing ontology-based and similar models facilitate matching available services with service requests via a process akin to type matching [2]. In contrast with the existing approaches, our approach combines empirical and taxonomic properties. The empirical ratings are multidimensional, and are used to induce a taxonomy. Thus this paper applies in settings where requests are matched to providers based on (empirical) data regarding services. To support the above criteria, we represent both the knowledge of each service and the request for a service as vectors. Our approach uses vectors primarily as a formal representation for classifying and selecting business services. Each dimension of a service description vector captures one aspect of a service's performance. Likewise, a service request vector specifies values on the different dimensions that jointly characterize the kind of service being sought. Such vectors would be produced in an application-specific manner. An advantage of vectors is that they are simple and amenable to manipulation and a variety of similarity measures, which can be used as the basis of performing a match.

2. FORMALIZING SERVICE SELECTION

This section motivates several key properties of service selection. To frame our problem and approach, let's now consider a situation where services are associated with a taxonomy or more generally a concept ontology. Each service may be associated with a concept in the ontology. Here, the subconcepts of a concept correspond to different specializations or (thinking in terms of human experts) specialties of the given concept. It is reasonable to take the view that specialists exhibit greater expertise for their specialty and potentially (but not necessarily) lower expertise in other specialties than generalists and those who specialize in other arenas.

Figure 1 (ignore the numbers for now) shows an example taxonomy, where the concept *Medicine* forms the root. The subconcepts reflect various medical specialties. Here, Alice (Retinal Ophthalmology) and John (Ophthalmology) are medical practitioners viewed as service instances, whose specialization is indicated through the taxonomy.

Properties of Taxonomic Matching. Service selection becomes interesting when an exact match for a requested specialty is not found. The strength of the match between a requested service and a concept determines whether the services associated with that con-

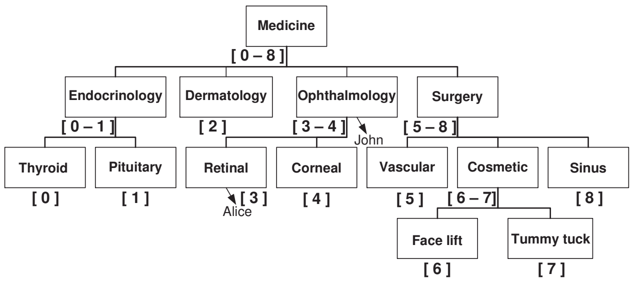

Figure 1: Assigning dimensions to concepts

cept are considered matches for the given service request. To enable effective service selection, we need an approach by which we can compare among the possible matches. We have identified certain key properties that characterize service selection. We list four of the properties below.

Property 1. *A target most strongly matches itself.*

Property 2. *If the choices are mutually ordered subconcepts of the target, the higher choice is preferred.* As an example, when looking for a surgeon, a slightly over-specialized vascular surgeon is a closer match than a cardiovascular surgeon with deeper expertise.

Property 3. *A descendant of the target at i steps away is preferable to an ancestor at i steps away.* When $i = 1$, a child of the target is a better match than its parent. Intuitively, selecting a more specialized service we would expect to obtain the desired expertise in the topics of interest, whereas a generalist would have "wasted" some expertise on topics that are not of high interest.

Property 4. *A descendant of the target is preferred to a choice that is incomparable to the target.* Intuitively, a descendant adds expertise in the topics related to the target and hence would be a better match than an unrelated choice, whose strength would be in topics farther from the target. As a special case, a child is a better match than a sibling.

3. MAPPING TAXONOMY TO VECTORS

The essential idea behind expressing a taxonomy in a multidimensional structure such as a vector is to ensure we have sufficiently many dimensions to capture the structure of the taxonomy. A simple way to generate a suitable mapping is to associate each leaf of the taxonomy with a set containing a single dimension (for convenience, a unique natural number). This is merely the "home" dimension for the leaf. A service instance associated with that leaf would be assigned a vector consisting of all the dimensions being considered, but the highest value of the vector would occur along the given dimension. Nonleaf concepts would be associated with a set of contiguous "home" dimensions that unions the dimensions with which their children are associated. Consequently, a bottom-up labeling of the concepts would identify the dimensions of each concept. Figure 1 illustrates the result for the medical taxonomy described earlier. Here the integers refer to the index of the dimensions beginning at zero.

Let's consider a service instance associated with a particular concept in a taxonomy. What vector should we assign it to capture the knowledge associated with a specific concept? This concept could occur anywhere in the taxonomy, not just as a leaf. Thus, the given service instance should obtain the highest values for the home dimensions of that concept. Further, we would not expect there to be any discrimination among the home dimensions, because we have no reason to believe that they are different for the service instance under consideration. The following are candidate approaches for assigning vectors to service instances based on their associated concept. First, a *boolean approach*, where a vector has 1 for each of the home dimensions and 0 elsewhere. Second, a *weighted approach*, which is a simple recursive approach is used to create a vector whose elements are positive real numbers. Consider two real numbers α and β such that $0 < \beta \leq 1 < \alpha$, which capture the strengthening and weakening of expertise along particular dimensions, respectively. Associate the root of the taxonomy with a vector of the home dimensions of the root, with each element set to 1. For each other concept, generate a vector based on its parent's vector by setting each of this concept's home dimensions to α times its value in the parent vector and setting each of its other dimensions to β times its value in the parent vector. Intuitively, a specialist would know more about his specialty than anyone else. As someone becomes more effective at his specialty, in general, he would become less effective at other specialties. Third, in a *ordinal approach*, we simplify the weighted approach to consider not absolute weights but ordinals. A weight vector would be mapped to vector of natural numbers where the lowest weights are replaced by 1, the next lowest by 2, and so on. Specifically, set $\beta = 1$. Then, the vector for a concept captures information about the path from the root to that concept.

Matching Service Vectors. Effective matching presumes the specification of a distance (or, inversely, similarity) metric between vectors. Since here, the vectors' values along different dimensions correspond to their placement in a taxonomy, we require a distance metric that has an element-wise way of functioning. We consider the *Manhattan distance* ($d_{ij} = \sum_{t=1}^{n} |e_{i_t} - e_{j_t}|$) between two vectors which equals the sum of the element-wise distance between the given vectors. A better match corresponds to a smaller distance.

4. DISCUSSION

We have arrived at some formal results proving whether the vector representations satisfy the desirable mathematical properties described for taxonomies. We have established that vector-based representation can yield more effective service matching than a traditional static approach. This is because the vectors can capture ratings of services, whether kept individually by the agents or collected at a reputation agency. Further, when suitably constructed and matched, the vectors reflect the taxonomic structure that provides an intellectual basis to service matching. Based on the long history of using vectors for matching in connection with information retrieval [3], and the techniques developed for handling extremely high dimensions, we are confident that vector representations can support large-scale service selections.

5. REFERENCES

[1] M. N. Huhns, U. Mukhopadhyay, L. M. Stephens, and R. D. Bonnell. DAI for document retrieval: The MINDS project. In M. N. Huhns, editor, *Distributed Artificial Intelligence*, pages 249–283. Pitman/Morgan Kaufmann, London, 1987.

[2] C. A. Knoblock, Y. Arens, and C.-N. Hsu. Cooperating agents for information retrieval. In *Proceedings of the 2nd International Conference on Cooperative Information Systems*, pages 122–133, 1994.

[3] G. Salton and M. J. McGill. *An Introduction to Modern Information Retrieval*. McGraw-Hill, New York, 1983.