Querying Real World Services Through the Semantic Web[*]

Kaoru Hiramatsu, Jun-ichi Akahani, and Tetsuji Satoh

NTT Communication Science Laboratories
Nippon Telegraph and Telephone Corporation
2-4, Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-0237 Japan
{hiramatu,akahani}@cslab.kecl.ntt.co.jp, satoh.tetsuji@lab.ntt.co.jp

Abstract. We propose a framework for querying real world services using meta data of Web pages and ontologies and implement a prototype system using standardized technologies developed by the Semantic Web community. The meta data and the ontologies enable systems to infer related classes while modifying the queries and retrieving approximate result and help them smooth obstacles in interaction while querying. The standardized technologies help us to describe the meta data and the ontologies in RDF (Resource Description Framework) and OWL (Web Ontology Language) formally and enable us to share them on the network. In this paper, we illustrate details of the framework and the prototype system, demonstrate how it works using practical data in Kyoto, Japan, and discuss requirements for actualizing this framework on the network.

1 Introduction

Some of the most popular and frequently used content on the Internet is information on real world services, such as that containing location information (e.g. store locations and hours of business) and services (e.g. timetables and route maps of public transportation systems). Much of this information is accessible as Web pages, and users can search for them with keywords using major search engines as they can with other kinds of contents by trial and error. In searching for real world services, locality is one of the most important factors; therefore it is necessary for users to specify a region geographically in order to find the real world services efficiently. However, because of the ambiguity of location names, users have to enter an exact address or use some heuristics (e.g. use a zip code as a keyword instead of an exact address) that are often noted on help pages for local searches.

Such inflexibility and heuristics on the current Internet are mainly caused by the ambiguity of terms extracted from Web pages by the search engines and queries submitted by users. To handle this ambiguity, much research has been done on meta data of the Web pages with respect to information retrieval [1], natural language processing [2][3], and databases [4][5]. Although these academic approaches show steady improvement and some of them mention sharing and reusing the meta data, no standardized frameworks and languages have yet come into wide use.

[*] Previous version of this paper was presented at the demonstration session of the 2nd International Semantic Web Conference

S.A. McIlraith et al. (Eds.): ISWC 2004, LNCS 3298, pp. 741–751, 2004.
© Springer-Verlag Berlin Heidelberg 2004

The Semantic Web [6] is expected to play an important role in improving the situation on the Internet. Thanks to Semantic Web activity at the World Wide Web Consortium (W3C), Resource Description Framework (RDF)[1] and Web Ontology Language (OWL)[2] have been standardized as frameworks of the Semantic Web. They enable us to describe meta data of the Web pages and the ontologies formally. The formalized meta data and the ontologies are regarded as being machine-readable semi-structured knowledge and as sharable on the Internet without ambiguity. Such knowledge will not only enable the search engines to handle terms extracted from the Web pages and the queries more accurately, but also help the systems provide users with various related information using the taxonomic knowledge in the ontologies.

In this paper, we propose a framework for querying information on real world services and implement a prototype system based on the framework utilizing standardized technologies developed by the Semantic Web community. Taxonomic knowledge and database schemata of the real world services are integrated into the ontologies that are described in OWL. Based on these ontologies, the meta data of the Web pages concerning real world services such as location, hours of business, and service types, is extracted from the Web pages by a scraping program, converted into RDF data, and stored in service provider agents separately according to their area of responsibility and service categories. The meta data and the ontologies help systems avoid ambiguity of terms extracted from Web pages and queries, and enable them to infer related classes for modifying the queries and searching for approximate results.

The prototype system accepts queries in natural language sentences and displays search results in several styles (e.g. map view and tree-structure view) according to meta data of the search results. As in the upper example of Fig. 1, the prototype system accepts a query “Find traditional Japanese restaurants near a bus stop that will be open around 2 p.m. tomorrow[3],” then returns three search result sets that include a restaurant and a bus stop, respectively. The prototype system utilizes the meta data and the ontologies to smooth interactive querying. For example, if no result satisfies the initial query, the prototype system will extend its search range using the meta data and the ontologies (e.g. find a category “Restaurant” that is a superordinate category of “Japanese Restaurant.”)

The accepted queries are translated into queries in extended SQL [7] and submitted to a mediator agent, which coordinates the optimal combination of service provider agents according to advertisements for the service provider agents in DAML-S [8]. Then, the mediator agent submits the query to the coordinated service provider agents and a service integrator agent integrates the search results returned from them. As the lower example in Fig. 1 shows, when a user submits the query “find a route to Kyoto station and a bank on the way to Kyoto station,” the prototype system forwards the query to two service provider agents, a route finding service and a location information service, and outputs integrated results on a result viewer (lower right of Fig. 1.) Moreover, to enable the users to digress from the search results, the prototype system extracts meta data of selected search results as default values of future queries.

[1] RDF: http://www.w3c.org/RDF/

[2] OWL: http://www.w3c.org/2001/sw/WebOnt/

[3] As in the screenshot (Fig. 1), this version of the prototype system only accepts queries in Japanese.

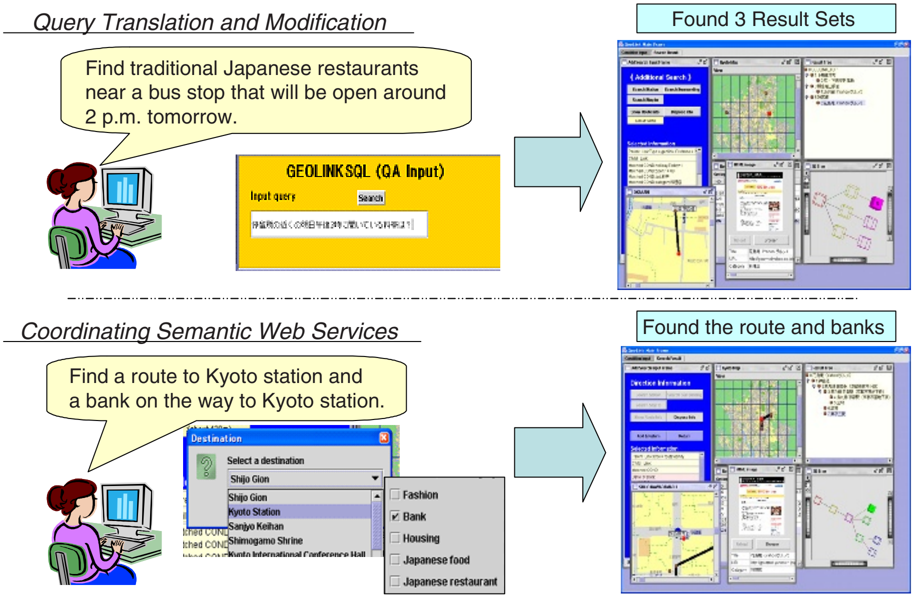

Query Translation and Modification
Find traditional Japanese restaurants near a bus stop that will be open around 2 p.m. tomorrow.
GEOLINKSQL (QA Input)
Input query
Search
Found 3 Result Sets
Coordinating Semantic Web Services
Find a route to Kyoto station and a bank on the way to Kyoto station.
Destination
Select a destination
Shijo Gion
Shijo Gion
Kyoto Station
Sanjyo Keihan
Shimogamo Shrine
Kyoto International Conference Hall
□ Fashion
☑ Bank
□ Housing
□ Japanese food
□ Japanese restaurant
Found the route and banks

Fig. 1. Example of the prototype system

In the remainder of this paper, we illustrate details of the framework and the prototype system, and demonstrate how it works using practical data in Kyoto, Japan. We also discuss requirements for actualizing this framework on the network through comparing it with related work.

2 Framework and Prototype System

2.1 Overview

Information on real world services is provided by many directory services (e.g. Yahoo![4] and Zagat survey[5].) and rendered in various styles. Almost all of entries in them include names, addresses, hours of business, and telephone numbers. These fields are regarded as basic and common data. The other fields such as ratings, reputations, and atmospheres of places are also familiar and useful for users to make their choice. All these data are important and meaningful for sharing as meta data of information on real world services. However, in contrast to the former fields that are objective, the latter fields are filled in by reporters subjectively; consequently, it is necessary for inference engines to take account of mutual trust between reporters and users. To simplify the prototype system, we mainly utilize the former fields as the meta data of the information on real world services.

In order to collect meta data of information on real world services, we implemented a scraping program based on Web robots. It accesses Web pages according to a URL

[4] Yahoo!: http://www.yahoo.com/

[5] Zagat survey: http://www.zagat.com/

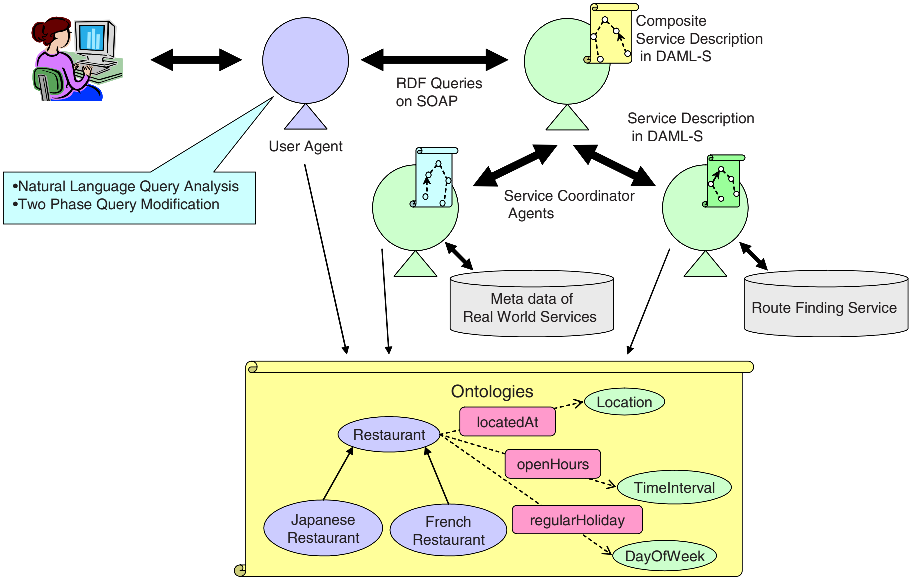


Fig. 2. System overview

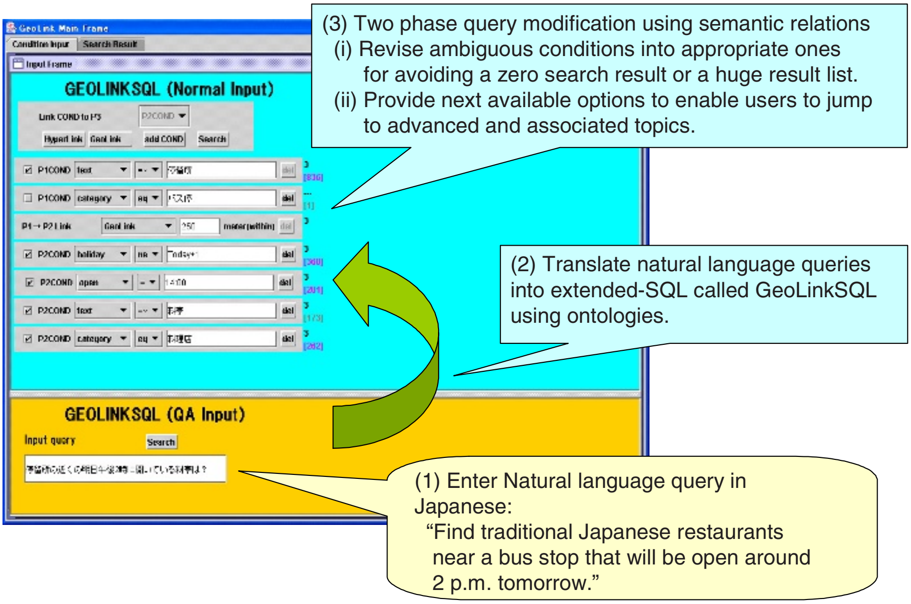
list and downloads them recursively. It then analyzes the downloaded Web pages using morphological analysis and pattern matching and extracts fields of names, addresses, and business hours of the real world services. In this analysis, we utilize digital map data as dictionaries for extracting the names and the addresses and assigning geographic coordinates to the downloaded Web pages belonging to the real world services. The extracted fields are converted to RDF data and stored in service provider agents separately according to their area of responsibility and service category. However, it is difficult and unreasonable for the scraping program to extract meta data of all the information residing in route planners and dynamic databases in the same manner. Therefore, we utilize a service description in DAML-S instead of employing their meta data.

Ontologies of the meta data are described in OWL. The ontologies are utilized to determine field extraction by the scraping program and to infer related and superordinate categories by user agents. As shown in Fig. 2, the ontologies include data schemata of the extracted fields and related knowledge such as hierarchies of service categories and addresses.

The prototype system based on this framework consists of a user agent and service coordinator agents (Fig. 2). In this prototype system, one service coordinator agent performs one or more roles of a service provider agent, a mediator agent, a service wrapper agent, and a service integrator agent. We will describe the details of these agents in Section 2.3. These modules are implemented using Java[6], Jena[7] for manipulating

[6] http://java.sun.com/

[7] a Java API for manipulating RDF models, http://www.hp.hp.com/semweb/



☑ P1COND text =~ 停留所 [836]

□ P1COND category eq バス停 [1]

P1→P2 Link Geo link 250 meter(within)

☑ P2COND holiday ne Today+1 [360]

☑ P2COND open = 14:00 [201]

☑ P2COND text =~ 料亭 [173]

☑ P2COND category eq 料理店 [262]

Input query: 停留所の近くの明日午後2時に開いている料亭は？

(3) Two phase query modification using semantic relations
(i) Revise ambiguous conditions into appropriate ones for avoiding a zero search result or a huge result list.
(ii) Provide next available options to enable users to jump to advanced and associated topics.

(2) Translate natural language queries into extended-SQL called GeoLinkSQL using ontologies.

(1) Enter Natural language query in Japanese:
"Find traditional Japanese restaurants near a bus stop that will be open around 2 p.m. tomorrow."

Fig. 3. Query translation

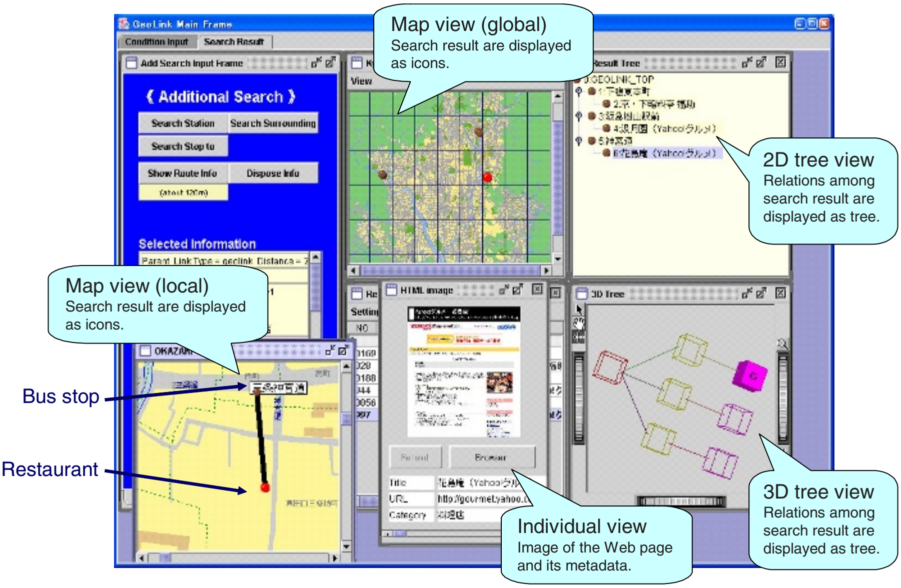
RDF models, PostgresSQL[8] for storing RDF data, and Jun for Java[9] for displaying tree structures of search results in the result viewer. These modules utilize SOAP (Simple Object Access Protocol) for message passing.

The user agent behaves as a front-end of the prototype system. As Fig. 3 shows, the user agent can accept queries either in natural language style or in form style, which is equivalent to queries in extended SQL. The query entered as the natural sentence is classified as a query type using SAIQA[10] and translated into a query in extended SQL using a template that is derived from the query type.

The query in extended SQL is submitted to the mediator agent. According to advertisements of the service provider agents in DAML-S, the mediator agent coordinates optimal combination of the service provider agents, divides the query according to the combination, and submits the divided query to the organized service provider agents. The service provider agents accept the query and returns search results that satisfy the queries based on the meta data of the real world services that the service provider agents store according to their area of responsibility and service category. The service integrator agent then integrates search results which are returned from the service provider agents into result sets based on relations that are specified in the query in extended SQL.

While processing a query, the prototype system applies two-phase query modification [9] to it. We will explain the details of this modification in Section 2.2.

[8] an open source Object-Relational DBMS, http://www.postgresql.org/
[9] a 3D graphics class library, http://www.sra.co.jp/people/nisinaka/Jun4Java/
[10] a question answering system developed by NTT Communication Science Laboratories



Map view (global)
Search result are displayed as icons.

2D tree view
Relations among search result are displayed as tree.

Map view (local)
Search result are displayed as icons.

Individual view
Image of the Web page and its metadata.

3D tree view
Relations among search result are displayed as tree.

Fig. 4. Screenshot of the result viewer

After the query processing, the user agent displays search results using a result viewing frame. As shown in Fig. 4, the result viewing frame includes a map view, a 2D tree view, a 3D tree view, an individual view, and an add-search-input frame. The map view displays search results as icons on a digital map, while the 2D and 3D tree views show the search results (e.g. three sets of a restaurant and a bus stop) using the tree style. In both tree views, a node in the tree view expresses a Web page that includes information on a real world service, where an edge stands for a relation between real world services (e.g. hyperlinks between Web pages and geographical relationships.) The individual view enables the user to browse images of the Web page and its meta data by clicking the icons on the map view or the nodes in the 2D and 3D views. The add-search-input frame prompts the user to conduct the next search by showing available search options that are coordinated from meta data of one or all of the initial search results and preset user profiles. To conform with popular design guidelines for user interfaces, the available search options are displayed as enabled buttons, while the others are displayed as disabled buttons.

This query processing of the prototype system is based on an augmented Web space [7] that enables the system to search for a set of Web pages with relations by specifying conditions of the Web page attributes, hyperlinks, and implicit geographic relations between the Web pages. In the augmented Web space, the implicit geographic relations are calculated as geographical generic links, which are created dynamically according to implicit geographic relations, such as distance and directions, between objects that are described in the Web pages. Owing to this function, the prototype system can find sets of search results that satisfy the queries. Although the original augmented Web space is designed to utilize only geographic relations, we extended it to handle the meta data for creating semantic links.

2.2 Two-Phase Query Modification

We employ two-phase query modification [9] to the prototype system for driving users to evolve interactive queries. This query modification is divided into two phases:

1. Revising ambiguous conditions into appropriate ones to obtain an adequate number of search results.
2. Providing next available options that enable users to jump to advanced and associated topics.

Both phases are processed tightly coupled with query processing in accordance with the meta data and the ontologies.

In the first phase, the user agent prefetches a number of search results of each term condition in the query and identifies ambiguous term conditions in accordance with the number of search results. It then revises them into appropriate ones automatically using the ontologies. For example, if a term extracted from the query is too ambiguous and an enormous amount of search results are matched, the term will be replaced with a more specific term along with word relations in the taxonomic knowledge. On the other hand, if a keyword results in a zero search result, then the keyword will be replaced with a hypernym or a superordinate category name. After that, while joining the retrieved Web pages, if a geographic condition between Web pages is too narrow, in that an adequate result cannot be produced, the condition will be expanded into a wider area in accordance with the characteristics of distribution of the meta data. In contrast, if a geographic condition is too wide, the condition will be shortened. The user agent applies this query modification repeatedly until a adequate search result is obtained.

In the second phase, the prototype system provides the users with the next available options, which are advanced queries from the initial one and associated topics with the original target. Since the search result has a tree structure, the next available options are developed from root and leaf nodes of the search result and from characteristics of links between Web pages, according to the meta data and the ontologies. After the initial search result is returned to the user, he/she is prompted to choose one of the next available options, which are explicitly displayed in the add-search-input frame of the result viewer. In the second phase, all the next options are displayed on the result viewer so that the user can select and browse them repeatedly through trial and error. We suppose this helps users deepen and broaden their interests, even if they do not have a clear search intention when they start querying.

2.3 Coordinating Real-World Services

Various pieces of information on real world services is available on networks and each of them is responsible for a type of information such as restaurant directories or route guides. It is thus necessary to find adequate services to coordinate search results. We therefore need to introduce service coordinator agents into our framework. In our framework, one service coordinator agent performs one or both of the following roles.

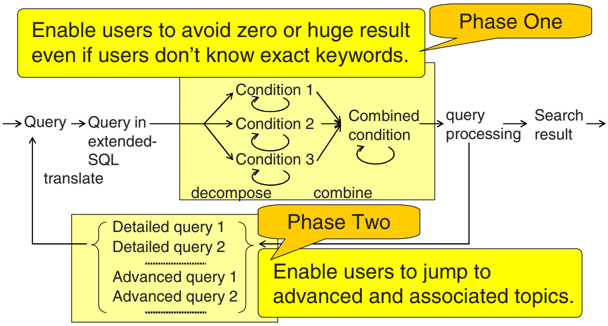


Fig. 5. Two-phase query modification

1. Service provider agents that provide services. Each service provider agent advertises its service description in DAML-S.
2. Mediator agents that forward queries to suitable service provider agents based on the service descriptions of the service provider agents.

Moreover, the service provider agents are categorized into the following two types.

1. Service wrapper agent that wraps Web services.
2. Service integrator agent that integrates services provided by other service provider agents. Each service integrator agent advertises a composite service description.

In the prototype system, we utilize two types of Web services: service provider agents and a route finding service. These Web services are wrapped by the service wrapper agents and are accessible in accordance with service descriptions in DAML-S. We have also implemented a service integrator agent that can integrate these two services.

3 Typical Example

For this prototype system, we prepared a test data set by extending the data originally created for the Digital City Kyoto prototype [10]. We constructed a URL list that includes about 5,000 Web pages of real world services in Kyoto, Japan, and extracted meta data from them using the scraping program. The meta data, such as names, hours of business, and addresses, are stored in service provider agents separately according to service area for which they were responsible, and service category. We collected service descriptions of the service provider agents and the real world services that coordinate search results dynamically.

To illustrate how the prototype system works, we employ the two example queries shown in Figure 1. The first is "find traditional Japanese restaurants near a bus stop that will be open around 2 p.m. tomorrow," and the second is "find a route to Kyoto station and a bank on the way to Kyoto station."

In the first example, a user inputs the example query into the field for a natural sentence. Then, the user agent analyzes the query and translates it into a query in extended

```
SELECT p2.url
FROM p1,link,p2
  p1.keyword =˜ ’traditional’
  p1.keyword =˜ ’Japanese restaurant’
  is_open(p1,’14:00’)
  link AS diatance(p1,p2) < 250m
  p2.keyword =˜ ’bus stop’
```

Fig. 6. Translated query in extended SQL

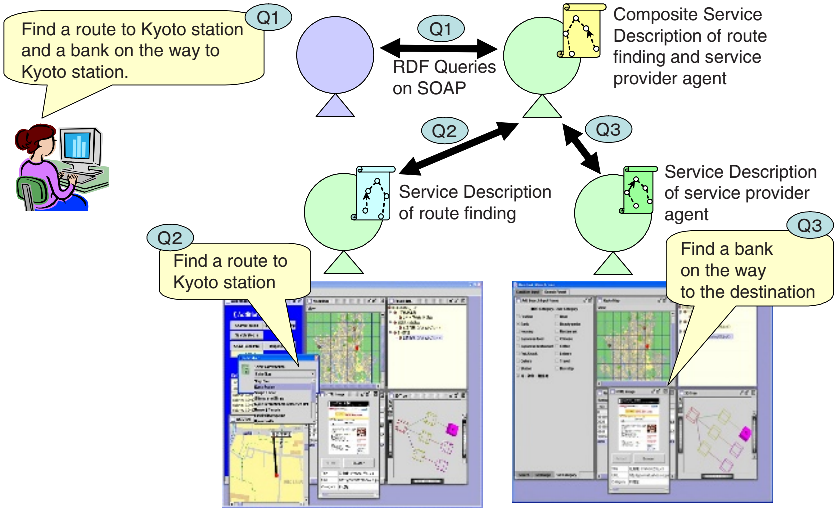

Q1
Find a route to Kyoto station and a bank on the way to Kyoto station.
Q1
RDF Queries on SOAP
Composite Service Description of route finding and service provider agent
Q2
Q3
Service Description of route finding
Service Description of service provider agent
Q2
Find a route to Kyoto station
Q3
Find a bank on the way to the destination

Fig. 7. Coordinating the agents of the real world services

SQL (Fig. 6). After translation, the user agent divides the query into term conditions and relational conditions, optimizes a processing order for them based on statistics of the test data set, and transmits them to a service coordinator agent. The service coordinator agent distributes the term conditions and the relational conditions to service provider agents based in a service area and service category for which they are responsible, and ask them to process each condition in the optimized order. While this processing occurs, the first phase of the query modification process is applied. The number of matching results for each term condition in the translated query is estimated by the query processing modules of the service provider agents. In this example, there is no term condition where the estimated result is zero or beyond the preset thresholds; if there is such a condition, then it will be replaced with more adequate term conditions using semantic relations in the ontologies.

Next, matching results are retrieved from the service provider agents and integrated into the search result in accordance with the link conditions in the query. As shown in the upper right of Fig. 1, the example query succeeded in retrieving three sets of a restaurant and a bus stop as the search result; however, if a link condition based on geographic distance is inadequate, the system extends or shortens the distance in order to obtain an adequate result. Also, if the combined result becomes invalid even though each condition is successfully processed, the system backtracks to the term replacement process and recombines the results until an adequate result is obtained.

The second example is conjunctive, meaning that the prototype system divides and processes it as in Fig. 7. The user agent translates a query and submits it to a mediator agent. In this phase, the mediator agent divides query Q1 into two: one is "find a route to Kyoto station" (Q2), and the other is "find a bank on the way to the destination" (Q3) according to the service descriptions. The mediator agent first submits query Q2 to a service wrapper agent, which provides a route finding service, and the service integrator agent receives route information as search results. The service integrator agent then asks the mediator agent to add the route information in the search results to query Q3 and submit it to the service provider agent to find a bank along the route. The search results of query Q3 are also returned to the service integrator agent and integrated with the search results of query Q2. The result viewer displays the integrated results and next available options as shown in Fig. 7.

4 Discussion and Conclusion

In this paper, we illustrated the details of the framework and the prototype system, and demonstrated how the prototype system works using practical data in Kyoto, Japan. We assume that enlargement of the Semantic Web will lead to a closer relationship between the Internet and real world services. To accelerate such evolution, we are planning to refine the framework and the prototype system along with meta data and ontologies.

As described in Section 1, many heuristics are diffused for searching for real world services using current search engines on the Internet. In general, the search engines accept keywords and return results using their indices created from terms scraped from the Web pages. This simple framework contributes to scalability and performance of the search engines, but it restricts users to conducting related information and cascading the result flexibly. In contrast to these search engines, users can utilize meta data of search results (e.g. names and locations of real world services) as default values of future searches using other information sources in our framework. However, extracting such meta data while processing queries causes a reduction in system performance. To avoid this tradeoff would be difficult; therefore it is necessary to estimate the quantity and quality of meta data.

As described in Section 2.1, we utilized common and objective meta data of information on real world services in the prototype system. Other meta data such as atmosphere, reputations, and ratings of real world services, are also meaningful for comparison, though such subjective meta data requires systems and users to confirm author information and date of creation and update. Such confirmation, which is regarded as a component of mutual trust is a difficult but worthwhile challenge in the Semantic Web.

The scraping program extracts meta data of the real world services based on the ontologies automatically using morphological processing and pattern matching. However, due to incomprehensive descriptions in Web pages and imperfect performance of natural language processing, the scraping program cannot always work correctly. To minimize the effect of errors, we need to investigate robust query processing and error checking by inferring engines that use ontologies.

Acknowledgment. Thanks are due to content holders for the permission to use their Web pages concerned with Kyoto, and to Yuji Nagato and Yoshikazu Furukawa of NTT Comware Corporation for their great contributions to demonstrating the system.

References

1. Jones, C.B., Purves, R., Ruas, A., Sanderson, M., Sester, M., van Kreveld, M., Weibel, R.: Spatial Information Retrieval and Geographic Ontologies An Overview of the SPRIT Project. In: Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR02). (2002) 387–388
2. Smith, D.A., Crane, G.: Disambiguating Geographic Names in a Historical Digital Library. In: Proceedings of the 5th European Conference on Research and Advanced Technology for Digital Libraries. (2001)
3. Krovetz, R., Croft, W.B.: Lexical ambiguity and information retrieval. ACM Transactions on Information Systems (TOIS) **10** (1992) 115–141
4. Knoblock, C.: Agents for Gathering, Integrating, and Monitoring Information for Travel Planning. IEEE Intelligent Systems **17** (2002) 63–66
5. Chen, C.C., Thakkar, S., Knoblock, C.A., Shahabi, C.: Automatically Annotating and Integrating Spatial Datasets. In: Proceedings of the Eighth International Symposium on Spatial and Temporal Databases (SSTD 2003). (2003)
6. Berners-Lee, T., Hendler, J., Lassila, O.: The Semantic Web. Scientific America (2001)
7. Hiramatsu, K., Ishida, T.: An Augmented Web Space for Digital Cities. In: The 2001 Symposium on Application and the Internet (SAINT2001). (2001) 105–112
8. Coaliation, T.D.S.: DAML-S: Web Service Description for the Semantic Web. In: The First International Semantic Web Conference (ISWC). (2002)
9. Hiramatsu, K., Akahani, J., Satoh, T.: Two-phase Query Modification using Semantic Relations based on Ontologies. In: Proceedings of IJCAI-03 Workshop on Information Integration on the Web (IIWeb-03). (2003) 155–158
10. Ishida, T., Akahani, J., Hiramatsu, K., Isbister, K., Lisowski, S., Nakanishi, H., Okamoto, M., Miyazaki, Y., Tsutsuguchi, K.: Digital City Kyoto: Towards A Social Information Infrastructure. In: Cooperative Information Agents III. Volume 1652 of Lecture Notes in Computer Science. (1999) 34–46