Services-Based Data Management in a Global Computing Environment

Dieter Pfoser Nectaria Tryfona Vassilis Verykios
Research Academic Computer Technology Institute
Athens, Hellas
{pfoser|tryfona|verykios}@cti.gr

Abstract

One of the main challenges in today's world of vastly distributed sources of information is to re-combine information sources to provide uniform access. We propose a distributed data management system that should provide the "glue" for combining data sources. This system advocates services as a means to access data. New services are defined on demand and their creation is supported by a behaviorist approach that incorporates new service ideas provided by the user. Services can be based on data and/or on the output of existing services. To increase the usability of services in the system we utilize two ontologies to denote relevant metadata. Service ontology structures existing services and helps in discovering new ones. Parameter ontology structures the parameters used in services and supports the creation of new services. Our proposal of a services-based data management system exhibits similarities to the newsgroup approach in that both "systems" are examples of semantic search engines based on user interaction. By exploring these similarities and by looking at some statistics of newsgroup user/posting behavior, we validate our services-based approach.

1. Introduction

"Databases were large aggregations of programs, cathedrals of software engineering, requiring vast system resources that supported efficient centralized data handling and storage in a cumbersome and rather inflexible way." This or similar could be an entry in the 2010+ encyclopedia of computer science presenting historic views on information technology related concepts. Now, the question is, assuming this will once be a historic view, what will be this future concept of databases that makes this rather current view look like the past? Global computing could be one answer, and in this work, we present a *data-centric view* of such an environment that relies on data distributed over a large number of mobile clients. This work is part of an initiative towards the development of such an environment, termed DBGlobe [17].

Our effort represents a paradigm shift from storing data in monolithic data management systems towards seeing it distributed over a (large) number of small-scale, mobile, data carrying devices, the *Primary Mobile Objects* (PMO). Examples of such devices could be Personal Digital Assistants (PDAs), or palmtop computers. The main objective of such an environment is then to provide the means that one can locate and invoke services or pose queries to a set of devices, i.e., to provide the "glue" for the PMOs to act as a single virtual database. Moreover, our demands are such that varying combinations of subsets of these devices form larger databases on the fly.

In this work, we introduce an approach whose cornerstones can be quoted as "service-centered," "demand based," and "minimally centralized". In a nutshell, we provide data access through services. The identification of existing and the construction of new services is aided by appropriate data structures (e.g., ontologies). Additionally, we employ behaviorist models to facilitate the creation of new services on demand; this includes stating ideas for new services in an unstructured, prose-like way. We argue that our service-based approach bears similarities to newsgroups, in that not all the information resources exist from the beginning but are constructed based on requests and a reply in the form of a new service. A way of looking at newsgroups is to view them as a semantics-based search engine that relies heavily on user involvement. We will explore the similarities between the two approaches to establish whether the reasoning about newsgroup usage can be applied in a straightforward way to our approach.

The remainder of this paper is organized as follows. Section 2 gives a brief overview of the DBGlobe architecture and introduces a metadata scenario the following sections rely on. Section 3 elaborates on using services to encapsulate data and data access as well as on service creation and discovery. Section 4 discusses feasibility considerations by using newsgroups and their relationship to the service-based approach. Section 5 discusses related work. Section 6 gives conclusions and directions for future work.

Proceedings of the Fourth International Conference on Web Information Systems Engineering Workshops (WISEW'03)
0-7695-2103-7/04 $20.00 © 2004 **IEEE**

IEEE COMPUTER SOCIETY

2. DBGlobe

A system such as DB-Globe can be seen as the natural evolution of the Internet. Empirically, this view of the future computing world is verified by observing the user of such devices, the ordinary human. *People move*, whether it is locally from their home to their workplace or long-distance on a business trip or even on vacation. *People carry information* with/on them; this can be addresses, travel directions, work reports, customer data, etc. Also, people not only view this information but they also collect it. Such data may range from addresses to sensor data.

A basis for discussing a service-oriented proposal for data management is its architectural setting. DBGlobe represents a paradigm shift from storing data in monolithic data management systems towards seeing it distributed over a (large) number of small-scale, mobile, data carrying devices, the *Primary Mobile Objects* (PMO). Examples of such devices could be Personal Digital Assistants (PDAs), or palmtop computers. The main objective of such a computing environment as DBGlobe is then to provide the means that one can pose queries to a set of devices, i.e., to provide the "glue" for the PMOs to act as a single database. Moreover, our demands are such that varying combinations of subsets of these devices form larger databases in an ad-hoc manner.

2.1 Architecture

Besides the PMOs, devices exist that comprise the stationary part or infrastructure of such a system. Figure 1 illustrates the principal system architecture. PMOs are spatially grouped into cells. Although the concept of clustering devices into cells stems from cellular and wireless networks, we do not assert a specific technology for connecting the devices to a network. Each cell is administered by a Cell Administration Server (CAS) [23]. CASes are interconnected and form a backbone network. They comprise the DBGlobe infrastructure, the stationary part of the system. A PMO is connected via a CAS to the DBGlobe system. Taking the mobility nature of a PMO into account, the connection is typically realized through a wireless link in the form of, e.g., third or fourth generation mobile phone network or IEEE 820.11 wireless network [7].

Besides the *administrative grouping* of PMOs, a *semantic grouping* exists based on the type of information they carry. Groups of PMOs can form ad-hoc databases to combine their information. To form such databases, we have to identify a set of PMOs that carry information related to a request, a query. Primarily the defining criterion for such an ad-hoc database is the query. However, keeping in mind that we are dealing with

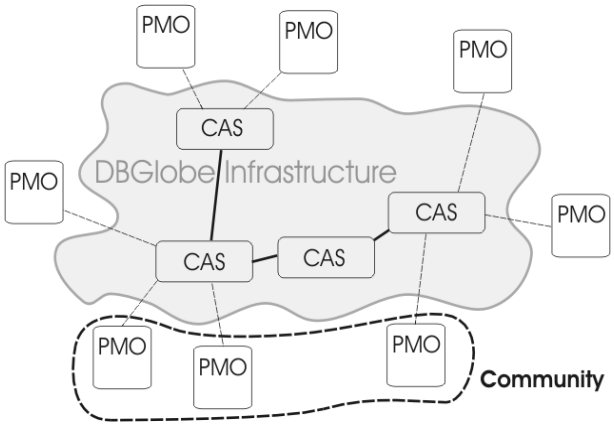

Figure 1. DB-Globe system architecture [17]

mobile entities, their location, as well as their temporal aspects, play an important role to this definition. In the mobile world, devices are accessible when and wherever "it pleases them." Thus, given a query about the history of the Acropolis being asked in the proximity of this site might take into account that neighboring PMOs carry this information. This additional criterion can be similar to a constraint in classical database theory. In more general terms, we term the set of PMOs forming an ad-hoc database a *community* (cf. Figure 1), and the defining characteristic the aspect of a community, which can be any combination of spatial, temporal, or thematic characteristics (depending on the query and the constraints). For example, an ad-hoc database is formed by all PMOs belonging to the "friends of Acropolis" community. Relating queries to communities allows us to pose a query first to the ad-hoc database existing for this community. Being unsatisfied with the query results, the query can be passed on to other portions of the global computing environment.

2.2 Metadata

Metadata stands for extracted structure and meaning of data. In the DBGlobe context, we encounter *profile data* and *content metadata*. The former describes the user and the device itself, whereas the latter describes the data contained in the PMO. A brief description of profile data follows. A more complete view can be found in [15]. Content metadata is described in detail in Section 3.2 in connection with service creation and discovery.

Users do have preferences with respect to what information they usually request, and considering mobility, as to when and to where they do this. Recording these data leads to the creation of a *user profile*. All data that characterizes the PMO will be stored in the *device profile*. We aim at capturing (i) the characteristics of the

IEEE COMPUTER SOCIETY

device itself, e.g., screen size and (ii) the characteristics of the device with respect to the DB-Globe system. The latter data include credentials, after registering with the DB-Globe network, the device obtains *parameters* used in subsequent interactions, a schedule for the *availability* of data, and the *community* a PMO belongs to.

An important property of the device in connection with mobility and related applications is its movement. We provide a mobile ontology that is based on trajectories [14], interpolated samples of the position of the moving object. Given this representation, the ontology captures properties of the trajectory, e.g., speed, relationships to other trajectories, e.g., meet, and to its (spatial) environment, e.g., cross, can be derived [15].

3. Service Orientation

The challenge posed by the architecture proposed in Section 2 is the handling of the data and requests. The term request is used instead of query, since we advocate services to access data instead of query-processing. Considering DBGlobe a distributed database system poses the challenges of distributing and recombining queries and the respective results. This approach is only feasible if a global ontology covering all the data exists and all PMOs possess query-processing capabilities based on a standardized query language. Whereas the latter is more likely to be achieved, the former seems to be an impossibility [15]. In the following, we introduce a rather different approach that is based on services.

3.1 Services and Data Access?

Consider the following question. *What if we knew all the data* and *all the queries* ever posed to the system in advance? In this case, although cumbersome, we could define for each query how to process it. Thus, each query would have a unique identifier by which users can request it. Services are just that. Instead of posing the query "What is the weather in Athens today," one can provide a "Weather service." Services can be seen as anticipated queries. The advantage here is that we precisely know of how to reply (query result) to a service request (query).

Now, this assumption might be too restrictive, but by making the following assumptions, it still holds. First, the user does not always know, or, know precisely what she wants. She relies on ideas, suggestion from others, e.g., the system (*service browsing*). Second, even if a service does not match an exact request, a user might be content with a close match, thus slightly modifying, channeling a user request (*service adoption*). Since in most cases users do not really precisely know what they want, services allow us to be pro-active in suggesting services that closely match the request. Third, in case no service matches (not even close) the request, the request is recorded and as soon as possible adopted resulting in a new service (*service creation*). With this framework, the objective is not to describe all existing data existing in the system but to describe it on demand. Demand is determined by service requests. Thus, the critical task is *to provide fast service development*, i.e., to minimize the time it takes to implement a service after its initial request.

Defining services and implementing them is not a simple task. What we actually ask from PMO owners is to provide services that cover the entire data contained on each device. This seems only be possible if heavily supported by the system, e.g., by means of framework service implementations (code), guides for the definition of service interfaces, etc. (cf. [16]).

On the downside, services narrow the scope of the data handling in such a system. To allow access to a particular data set, one has to define a service (accessing the data set) first. Assuming that we have a total amount of 1 terabyte of data in such a system, but only services accessing 100 megabytes exist, the value of our approach is highly questionable! To partially overcome this problem, we advocate a service hierarchy in that we not only specify services in our framework put also *suggestions for services*. *Service Frameworks* are defined services (e.g., in the form of prose) that await their concrete implementation. Thus, a user who does not find the service she is looking for can specify a service framework if she has a clear idea of how such a service should work. At a more abstract level we introduce *Service Ideas* that advocate services that are needed but lacking concrete ideas for implementation. From a behaviorist point of view, services force the user to focus on describing his data.

Services Ideas ➔ Service Frameworks ➔ Services

Figure 2. Hierarchy of services

In our system, we have to take precautions to support the user in (i) discovering services. This includes the suggestion of new services in case no relevant ones exist and (ii) the construction of new services. The construction of new services is critical in that the user cannot spend an infinite amount of time on creating a new service. As we will see in the next section, services do not always rely only on data contained in one PMO (basic services) but can be based on data stemming from other services (complex services).

3.2 Content Metadata and Services

Should the reader now be convinced that using services is viable in this context and further accept that there might be a critical number of people to supply these

IEEE COMPUTER SOCIETY

services to willing consumers (whether this is really the case will be examined in Section 4), there still remains the question of how difficult it is to author services!

A service can be based on (i) only data, (ii) other services, or (iii) services and data. The service example of the previous section is based on data only. By incorporating a service call, e.g., to obtain a query parameter, the service would be based on services as well.

Content data are the actual data registered by the user on the PMO, which can be spatially-referenced and/or temporally-referenced information, indicating where and when the actual information was seen, or recorded. *Content metadata* describes these data. As these data are not directly exposed but accessed through services, content metadata will be in the form of ontologies relating to services.

Service discovery and *service creation* are essential tasks in a system such as DBGlobe. We introduce service ontology to support the structuring of the services and to aid service discovery. Further, the composition of services is a complex task and has to be supported by the system as well. Thus, *parameter ontology* is introduced to cover all the parameters used in the various services. To discover services, e.g., either to access them or to incorporate them in new services, we propose *service ontology*.

3.2.1. Service Composition and Parameter Ontology. Services exhibit an analogy to programs in that services have an interface consisting of a set of types (described using e.g., WSDL [24]). Now, by describing the types of services in terms of an ontology, we support the construction of new services.

Assuming a Service A is based on Services B and C as shown in Figure 3 (the arrows illustrate this relationship), when constructing the service, how can we easily identify B and C as the constituting service?

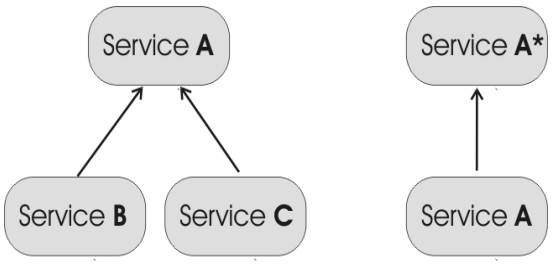

Figure 3. Service hierarchies

For example, we want to construct a service A that returns "restaurants where football players eat in Athens (location)," and a service B that returns "football players per location" and a service C "restaurants where famous people eat" exist, we can base the construction of A on B and C. These services can be described in terms of their in- and output parameters as follows (input ➔ output).
(A) *Restaurants_frequented_by_football_players*: (location ➔ {restaurant})
(B) *Football_players_in_places*: (location ➔ {person})
(C) *Restaurants_frequented_by_people*: (person ➔ {restaurant})

Another example is a weather service that provides for places weather forecasts.
(A) *Weather*: (location, time ➔ weather)
Assume we want to extend this service in that it provides weather information along a route.
(A*) *Weather_en_route*: (route ➔ {weather}).
Assuming that a service *Route*((start, end) ➔ route) exists that computes a route given a start and end point (for simplification we omit other parameters such as shortest path, shortest time, etc.), is it trivial to combine it with *Weather* to obtain *Weather_en_route*? Not if the parameters of the services match, e.g., *Route* returns a type route, which can be used as input parameter for *Weather* if we know how to decompose a route into locations at times.

Figure 4 shows a simple ontology containing classes (types) that might help us in constructing the *Restaurants_frequented_by_football_players* and the *Weather_en_route* service. The ontology generally introduces a set of classes (data types) and their relationships.

The ontology was devised using the ontology editor Protégé 2000 [18]. It comprises classes (the boxes) slots (or relationships, shown as labeled edges, e.g., *has_spatial_position*), and class hierarchies (edges labeled "isa"). All classes in the ontology relate to the superclass *Object*. At the next level, we have three basic classes *Spatial Object*, *Temporal Object*, and *Spatiotemporal Object*. Together, these four classes are abstract classes. Concrete classes should either be derived from Spatial, Temporal, or Spatiotemporal Object if they exhibit a spatial and/or temporal reference, or otherwise should directly be derived from Object. The classes *Timestamp*, *Position*, and *Spatiotemporal Position* are used to position the respective *Object* classes within their respective domain. E.g., spatial objects have a reference to a position object that contains their x- and y-coordinates.

The class *Restaurant* is defined as a sub-class of *Spatial Object*. The class *Football Player* is a sub-class of *Person*. *Person* has a reference to *Restaurant* (*eats_at*). *Football Player* has a reference to *Football Club*, which, in turn, is a Spatial Object.

Coming back to the parameter types used in the service descriptions, we realize that by checking the class hierarchy and relationships of the involved types in the

IEEE COMPUTER SOCIETY

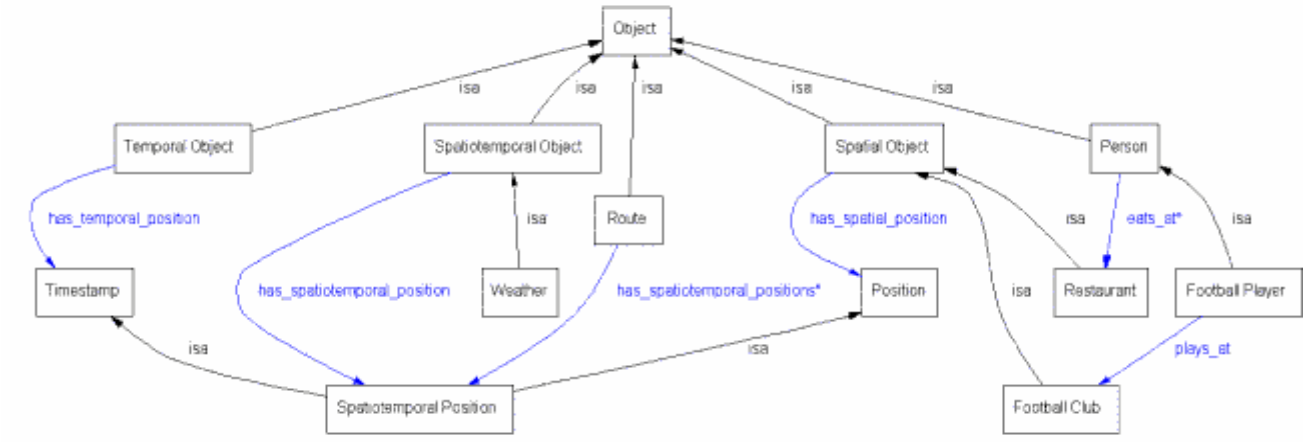

Figure 4. A simple ontology incorporating temporal, spatial, and spatiotemporal objects

ontology, we can verify the syntactic correctness of services, or, conversely, construct new services.

One could argue that using ontologies to denote metadata for service parameters is nothing other than creating metadata for the data itself. However, the service parameters do most likely but not necessarily belong to the domain of the data they encapsulate. Also, the return value of a service can be outside the scope of the data it is based on (closure property).

Services are not defined all at once but evolutionary, thus their semantics are denoted in sync. One could see services as an aid to gradually denote data semantics.

3.2.2. Service Discovery and Service Ontology. Matching requests with service descriptions is a quite common problem in literature. The most common approach is matching user requests expressed in keywords with service parameters (i.e., name, location, business, binding or tModel [22]) (UDDI-type discovery). However, this kind of discovery mechanism does not include semantics and is limited by the representation capabilities of the WSDL language [24], the SOAP protocol [20], and the UDDI registry [22]. These standards are designed to provide descriptions of message transport mechanisms (SOAP), and for describing the interface used by each service (WSDL).

In our context, once a service is defined, semantically it is more than the sum of its parts, i.e., service parameters. Knowing the semantics of the parameters of the service is not sufficient to reason about the semantics of the service and to locate a service that fits user requests. Thus, what is needed besides an ontology relating the parameters is a means to locate or discover and relate services, *a service tree*. The construct to realize such a structure can be of the form of another ontology (cf. also [13]). The service tree is an essential means for service discovery. In the previous section, new services were composed based on existing ones. We used an ontology relating parameters to verify the correctness of the resulting service. However, a precursor to this step is to identify the constituting services, i.e., to discover them.

Service ontology is hierarchically organized in the form of a tree, so that an information type (class) has a number of subordinate information types and at most one superior information type. Semantically similar services are connected to one node in the tree, e.g., to the topic "research." This organization allows an ontology to be structured according to a specialization relationship, e.g., a child to a node termed "research" could be "medical research" (cf. Figure 5). Each class in the service tree has a set of attributes that describes the domain of the referenced services. As those attributes are more abstract, they do not correspond directly to the types of the service interfaces (cf. parameter ontology).

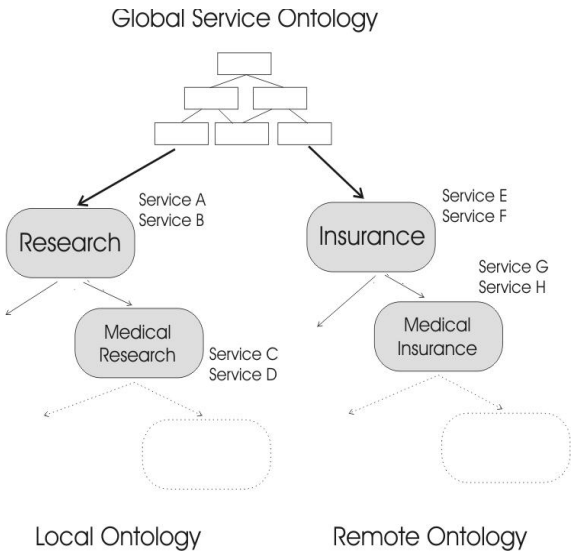

Figure 5: Example of Service Ontology

The classes of information types joined in the service tree support each other in answering requests directed to them. If a user request conforms better to the information type of a given subclass, then the query will be forwarded to this subclass. If no classes are found in the ontology tree while handling a request, then either the user simplifies the request or request is forwarded to other service ontologies through a global service ontology (cf. Figure 5).

Having metadata in the form of ontologies leaves us with the problem of where to place this information within the system. In the following section, the focus is on where to place the respective ontologies.

3.3 Communities, or Limiting the Scope of Ontologies

Section 2 defines communities as a set of PMOs that have data pertaining to the same interest. In the following, we present an alternate and more restrictive view in that we define communities to be islands of ontologies.

A problem when defining ontologies is their scope. Although it would be desirable to have a global parameter ontology, it would be difficult in establishing it in that for any given term one has to specify its unique meaning in a global context. However, when limiting the scope of such an ontology to a community it might be easier to establish it since users in this context have a similar background and are more likely to find/agree on a common vocabulary (parameter ontology). Limiting the scope of the parameter ontology to a community consequently means that its support of creating a new service is limited too, i.e., although we might not explicitly forbid the creation of services based on services stemming from different communities, it is not support by an ontology.

Figure 6 shows three ontologies that ideally are disjoint (in terms of the data they cover). However, it is safe to assume that have exhibit considerable overlap (gray shaded areas). Each community has its local parameter and service ontology and a global service ontology exists linking the local ontologies together.

Service ontology groups services of similar thematic interest in the classes of its specialization hierarchy. By its definition the classes of this ontology form communities. Again, if we assume that each community creates its local service ontology without having knowledge of other communities, the problem that remains to be solved is of how to compose the global service ontology as introduced in Section 3.2. The various communities can either be linked together manually by users belonging to more than one community, thus we enroll users that have diverse domain knowledge to create the links between ontologies, or we create this global ontology in an automatic, domain specific way, e.g., by using conceptual clustering [5] [16].

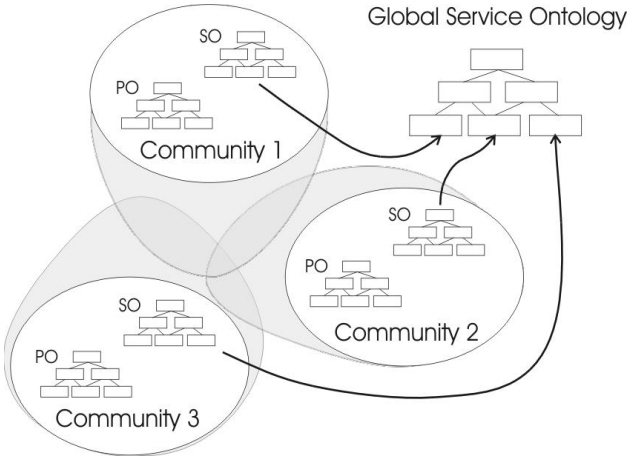

Figure 6. Communities and the various ontologies

4. Feasibility Considerations

When thinking about today's Web, Newsgroups are not the first item that comes to one's mind. Although dealing with a smaller amount of information when compared to the vast number of Web pages, they provide a high-quality search forum based on the fact that humans interact and introduce their expertise. Newsgroups and our service-oriented approach exhibit many commonalities. A newsgroup can be seen similar to a community in that in both cases an interest is shared. Many newsgroups have FAQs, whose purpose it is to summarize posts but also to establish a common ground, terminology for the group. In the service context we have parameter and service ontologies to support the two integral tasks of service creation and service discovery.

In a newsgroup the task of finding a post, this includes searching FAQs, is similar to service discovery. In case, this search is unsuccessful, one makes a post to a specific newsgroup (or more than one leading to cross postings). In the service context this is similar to providing a service idea. A reply to post, which constitutes a simple email, is similar to providing a new service based on a service idea. Table 1 summarizes these equivalences.

Although at the semantic level similarities exist, at the implementation level both approaches are fundamentally different. The viability of the service-oriented approach depends on how much service creation can be simplified, i.e., it should be that simple as it is to write an email. However, if we assume the simplicity assumption holds, we can use newsgroup statistics on posting behavior [25] to verify the applicability of the service-oriented approach, i.e., will people "post" services? For a broader discussion cf. [16].

IEEE COMPUTER SOCIETY

Table 1. Services and Newsgroup equivalents

Services		Newsgroups
discovering devices	←→	finding a post (incl. FAQ)
the suggestion (service ideas) of new services in case no relevant ones exist	←→	posting a question
the construction of new services	←→	posting a reply

It was found that *few users post many messages and many post few messages*. In the service context, this translates to that few users need to provide most of the services, thus need to possess most of the data. P2P networks put technical means to force users to provide data. Similar measures need to be taken in our approach. However, many newsgroups seem to thrive, in that they attract a critical mass of people to sustain their existence.

Many posts find no reply. This might prove even more problematic in the service context, where the reply to a service is not simply constructed by using one's knowledge but by providing one's data.

FAQs can be seen similar to a construct providing the terms, concepts used in a particular newsgroup. Thus, the fact that they *do not help establishing a common ground*, can be interpreted towards that such a construct will not be used for defining services, i.e., do define a new service one also provides his own parameters instead of using the parameter ontology. However, *moderation* had a positive effect on establishing a common ground. Thus, a "moderated" list of concepts, terms could be used instead.

One can conclude that the volunteer give-and-take principle of newsgroups can be successfully applied to our service-oriented data management approach under certain conditions. One, the requirements towards authoring services have to be minimal, i.e., as easy as writing an email. Two, the scope of a community has to be chosen carefully in that a narrow scope is prohibitive towards participation, whereas a wide scope is troublesome towards establishing a common ground, i.e., to define parameter and service ontologies. Third, moderation is helpful in defining the ontologies. A domain expert could act as the moderator of a community. Fourth, a critical number of services is needed to attract a minimal number of users that guarantee the continued existence of a community.

5. Related Work

There is a large body of literature dedicated to research issues related to data integration, multi-databases, information brokering systems, peer-oriented computing and peer-to-peer databases.

The TSIMMIS project [6] is a system for integrating heterogeneous information sources that may include both structured and semi-structured data. It is based on a mediation architecture and proposes a new data model, the Object Exchange Model (OEM), to achieve the integration.

The InfoSleuth [1] project presents an approach for information retrieval and processing in a dynamic environment such as the Web. InfoSleuth integrates agent technology, domain ontologies, and information brokering to handle the interoperation of data and services over information networks.

OBSERVER [9] is an architecture for information brokering in global information systems. One of the addressed issues is the vocabulary differences across the component systems. OBSERVER features the use of pre-existing domain specific ontologies to define the terms in each data repository.

WebFINDIT [2] aims to achieve scalability through the incremental data-driven discovery and formation of interrelationships between information repositories. Clusters of information repositories are established through the sharing of high-level meta-information, and individual sites join and leave these clusters at their own discretion.

The Piazza system [3] mainly focuses on the problem of dynamic data placement. The goals of the system are scalability, even with a large number of nodes, and support for moderately frequent updates.

MOCHA (Middleware Based On a Code SHipping Architecture) [8], is a novel database middleware system designed to interconnect hundreds of data sources.

A flexible personalization architecture for wireless internet based on mobile agents is described in [19]. User interests are matched onto internet site descriptions to reduce the data transmitted through wireless, low-bandwidth connections. Ontologies are used denote, both, user profiles and site descriptions.

AXML [10], a declarative framework for harnessing XML and web services for data integration, has also been proposed recently and can be put to work in a peer-to-peer architecture. The AXML framework is centered on AXML documents which are XML documents that may contain calls to web services. When calls, included in a document, are fired the latter is enriched by the corresponding results. In some sense, an AXML document can be seen as a materialized view integrating plain XML data and dynamic data obtained from service calls.

IEEE COMPUTER SOCIETY

6. Conclusions and Future Work

Global computing is an approach that relies on the distribution of data over a large amount of data sources, the PMOs. In this work, we described a service-oriented approach to provide access to individual as well as to sets of data sources. The two basic tasks in such a framework are service creation and discovery. We introduced two basic concepts, the parameter ontology and the service ontology to support these tasks. Further, we assume that services are created on-the-fly. A behaviorist approach should facilitate the creation of new services. Here, we try to denote the demand for services by recording service ideas and service frameworks in our system. These concepts denote abstract services that yet lack their concrete implementation. Although technologically different, newsgroups show similarities to our service framework in that providing services in response to requests is based on user interaction (post and replies). Collected statistics over posting behavior are used to assess the feasibility of our service-oriented approach.

The directions for future work are as follows. The system provides only limited querying capabilities to discover services. However, to handle large numbers of services, automated service discovery needs to be supported. This can be done by using, e.g., taxonomy dictionaries and distance measures based on service metadata [3] [9] [11] [12] [21]. Constructing the global service ontology is a critical task in that the quality of this ontology highly determines the results of service discovery. Thus, we have to empirically evaluate the conceptual clustering approach in terms of the quality of its result. Although the parameter ontology is a local construct in that its scope is limited to a community, these ontologies should incorporate references to outside ontologies already defining concepts, e.g., the Dublin Core metadata initiative [4]. An authority, e.g., a moderator, facilitates the construction of ontologies in our approach. With a growing number of participants it becomes important to examine less centralized approaches, e.g., as pursued by the Wikipedia project on constructing an encyclopedia [26]. Within the DBGlobe project a prototype is currently under development. A challenge will be to fully implement the ontologies and features proposed in this work in this prototype.

Acknowledgements

The authors wish to thank the partners of the DBGlobe project for their helpful discussions and their insight on this topic. This work was supported by the IST-FET project DBGlobe under contract number IST-2001-32645.

References

[1] Bayardo, B., Bohrer, W., Brice, R., Cichocki, A., Fowler, G., Helal, A., Kashyap, V., Ksiezyk, T., Martin, G., Nodine, M., Rashid, M., Rusinkiewicz, M., Shea, R., Unnikrishnan, C., Unruh, A., and Woelk, D.: InfoSleuth: Agent-Based Semantic Information of Information in Open and Dynamic Environments. In *Proceedings of ACM SIGMOD*, 1997.

[2] Bouguettaya, A., Benatallah, B., Hendra, L., Ouzzani, M. and Beard, J.: Supporting Dynamic Interactions among Web-Based Information Sources. *IEEE Transactions on Knowledge and Data Engineering*, 12 (5):779-801, 2000.

[3] Castro, P., Greenstein, B., Kermani, P., Muntz, R., Papadopouli, M., and Bisdikian, C.: Locating Application Data Across Service Discovery Domains. In *Proc. of the International Conference on Mobile Networking and Computing*, 2001.

[4] Dublin Core Metadata Initiative: Dublin Core Metadata Element Set, Version 1.1: Reference Description, *Web page*, http://dublincore.org/documents/1999/07/02/dces/.

[5] Fisher, D.: Improving Inference Through Conceptual Clustering. In *Proceedings of 1987 AAAI Conference*, pp. 461-465, 1987.

[6] Garcia-Molina, H., Papakonstantinou, Y., Quass, D., Rajaraman, A., Sagiv, Y., Ullman, J.D., Vassalos, V. and Widom, J.: The TSIMMIS Approach to Mediation: Data Models and Languages. *J. Intelligent Information Systems*, 8(2):117-132, 1997.

[7] IEEE: IEEE Wireless Standards Zone, *Web page*, http://standards.ieee.org/wireless/.

[8] MOCHA Web page, http://www.cs.umd.edu/projects/mocha/

[9] Mena, E., Illarramendi, A., Kashyap, V., and Sheth, A.: OBSERVER: An Approach for Query Processing in Global Information Systems Based on Interoperation Across Pre-Existing Ontologies. *Distributed and Parallel Databases Journal*, 9:223-271, 2000.

[10] Milo, T., Abiteboul, S., Amman, B., Benjelloun, O. and Ngoc, F. D.: Exchanging Intentional XML Data. In *Proc. ACM SIGMOD conference*, 2003.

[11] Nguyen, B., Vazirgiannis, M., Varlamis, I., and Halkidi, M.: Organizing Web Documents into Thematic Subsets using an Ontology. *VLDB Journal*, special issue on "Semantic Web," to appear.

[12] Ouksel, A. M. and Sheth, A.: Semantic Interoperability in Global Information Systems. *SIGMOD Record* 28(1):5-12, 1999.

[13] Ouzzani, M., Benatallah, B., and Bouguettaya, A.: Ontological Approach for Information Discovery in Internet Databases. *Distributed and Parallel Databases Journal*, 8:367-392, 2000.

[14] Pfoser, D. and Jensen, C.: Capturing the Uncertainty of Moving-Object Representations. In *Proc. of the 6th Int'l Symposium on Spatial Databases*, pp.111-132, 1999.

IEEE COMPUTER SOCIETY

[15] Pfoser, D., Pitoura, E., and Tryfona, N.: Metadata Modeling in a Global Computing Environment. In *Proc. of ACM GIS 02*, pp. 68-73, 2002.

[16] Pfoser, D., Tryfona, N., and Verykios, V.: Services-Based Data Management – the Newsgroup Way. Technical Report, Data and Knowledge Engineering Group, Research Academic Computer Technology Institute, Hellas. http://dke.cti.gr/pubs/tr/ptv2003.pdf, 2003.

[17] Pitoura, E., Abiteboul, S., Pfoser, D., Samaras, G., and Vazirgiannis, M.: DB-Globe, A Service-Oriented P2P System for Global Computing. *SIGMOD Record*, 32 (3), 2003.

[18] Protégé 2000 Project Homepage, *Web page*, http://protege.stanford.edu/.

[19] Samaras, G. and Panayiotou, C.: A Flexible Personalization Architecture for Wireless Internet Based on Mobile Agents", In *Proc. 6th East-European Conference on Advances in Databases and Information Systems*, pp. 120-134, 2002.

[20] Simple Object Access Protocol (SOAP) 1.1, *Web page*, http://www.w3.org/TR/SOAP.

[21] Sycara, K., Klusch, M., Widoff, S., and Lu, J.: Dynamic Service Matchmaking Among Agents in Open Information Environments. *SIGMOD Record* 28(1), pp. 47-53, 1999.

[22] Universal Description, Discovery, and Integration of Business for the Web, *Web page*, http://www.uddi.org.

[23] Ververidis, C., Valavanis, S., Vazirgiannis, M., and Polyzos, G.C.: An Architecture for Sharing, Discovering and Accessing Mobile Data and Services: Location and Mobility Issues", *Lobster Workshop, LBS for accelerating the European-wide deployment of Services for the Mobile User and worker,* Mykonos, Greece, 2002.

[24] Web Services Definition Language (WSDL), *Web page*, http://www.w3.org/TR/WSDL.

[25] Whittaker, S., Terveen, L., Hill, W., and Cherny L.: The Dynamics of Mass Interaction. In *Proc. of CSCW 98*, pp. 257-264, 1998.

[26] Wikipedia, the Free Encyclopedia, *Web page*, http://www.wikipedia.com.

IEEE COMPUTER SOCIETY