Metadata Modeling in a Global Computing Environment

Dieter Pfoser
Computer Technology Institute
Akteou 11 & Poulopoulou Str.
Athens, Hellas
+30-103416220
pfoser@cti.gr

Evaggelia Pitoura
Department of Computer Science
University of Ioannina
Ioannina, Hellas
+30-651098811
pitoura@cs.uoi.gr

Nectaria Tryfona
Computer Technology Institute
Akteou 11 & Poulopoulou Str.
Athens, Hellas
+30-103416220
tryfona@cti.gr

ABSTRACT
Emerging computational paradigms such as global and ubiquitous computing require some rethinking and innovative research ideas in many computer science areas. In this work, we aim at studying a mobile computing scenario from the database perspective. Given a global computing environment in which data is kept in a number of small-scale, data-charged, mobile devices that use, e.g., wireless networks, for communication, we want to assess the overall data scenario. We use an example to abstract the requirements to such a computing environment and to outline the various existing types of data. Included here are some metadata proposals related to the actual data stored in the device (content data) as well as to the data necessary to the functioning of the device within the computing environment (profile data). The metadata proposals are based on well-known languages and tools such as XML, RDF, UML, and ontologies.

Categories and Subject Descriptors
H.3.5 [**Information Storage and Retrieval**]: Online Information Services – *Data Sharing*

General Terms
Theory

Keywords
global computing, mobile computing, metadata, mobile ontologies, mobile devices, spatiotemporal databases.

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 DB-Globe [13]. In particular, we focus on the overall data scenario and the issue of handling the (meta)data in such systems. Metadata, i.e., data about data, is used to facilitate data discovery as well as to speed up data processing. These issues become especially important when dealing with a highly decentralized system exhibiting a large number of data storages that spatially migrate and the number of online devices varying greatly. Having a device in such an environment, how does one discover, e.g., information about historic sites in Greece? Our proposal is that each device communicates metadata about the data it contains, e.g., information about Acropolis, to its environment where it can be "discovered" by other devices posing queries.

In a global computing environment, the mobile devices contain various types of data. In this work, we identify three types. *Content data* refers to the bulk of the data contained in the devices. These data can cover a similar spectrum to what exists on the Internet. Metadata can be denoted with the help of ontologies in connection with XML and RDF. The mobile devices and their users themselves create, or incur additional data, termed *profile data*. As part of a proposal to capture these data, we introduce metadata proposals for user profiles, device profiles and movement data. Registering a mobile device requires the communication of its essence in terms of data. Here, we introduce *essential metadata* to contain an abstract view of the content data and profile data.

The outline of the paper is as follows. Section 2 introduces our approach to global computing by giving the overall system architecture. Section 3 outlines the whole data scenario as it exists for mobile devices. Sections 4, 5, and 6 elaborate further on content data, profile data, and essential metadata. Finally, Section 7 gives conclusions and directions for future work.

2. A NEW DATABASE PARADIGM
Before proceeding with our proposals regarding data and metadata, we have to answer one basic question. How could such a global computing environment look like? DB-Globe 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*

GIS'02, November 8-9, 2002, McLean, Virginia, USA.

Objects (PMO). Examples of such devices could be Personal Digital Assistants (PDAs), or palmtop computers. The main objective of such a computing environment as DB-Globe 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 ad-hoc.

Besides the PMOs, devices exist that comprise the stationary part or infrastructure of such a system. Figure 1 illustrates the principal system architecture. In this example, we have two ad-hoc databases composed of PMOs that connect to proxies. Proxies deal with disconnections and provide network interfaces. Taking the mobility nature of a PMO into account, the connection to a proxy 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 [8]. *DataStores*, are dispersed throughout the stationary network, to keep metadata and services related to the mobile entities, such as priorities, constraints, rules, categories and descriptions. *DataHandlers* (DH) execute rules in response to events, manage the flow of the data from one device to another and *execute queries*.

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. Data that can be collected may range from addresses to sensor data.

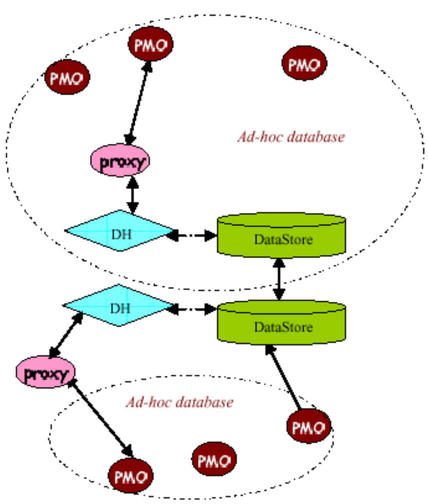

Figure 1: DB-Globe system architecture [10], [13]

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. This set of PMOs does not necessarily have to be static during the lifetime of a query be can either discover new "relevant" PMOs or drop "irrelevant" ones. For the query this means that either the query execution time is longer, or the notion of a query result has to be relaxed, i.e., towards a continuous query evaluation and partial results, e.g., [4]. Primarily the defining criterion for such an ad-hoc database is the query. However, keeping in mind that we are dealing with *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*, 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). E.g., 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.

3. ON DATA

The emphasis in our approach to a global computing environment is on scattered information that needs to be recombined for a satisfactory query response. To perform such operations metadata about the respective schemata has to be communicated a-priori. To come up with a metadata proposal, we have to first be aware of what data exists. In principal, arbitrary data such as it exists in the Web can be stored on PMOs. What makes our approach special is mobility, which not only affects the data but also produces new data. Overall, the data stored on PMOs can be grouped into the following three major categories.

Content data.

- *Descriptive data* and information, which is the actual data registered by the user on the PMO, e.g., the history of Acropolis.
- Content data can be *spatially and/or temporally referenced*, indicating where and/or when the actual information was seen, or recorded.

Profile data, characterizing the user and the device he/she is carrying.

- *User profile* captures the user and its preferences. For example a user visiting the Acropolis site may be a tourist or a scientist on an excavation site and this indicates different needs and interests.
- *Device profile* constitutes information about, e.g., the type of PMO (virtual, physical), availability of information (always, what time spans), ad-hoc database (community) it belongs to, PMO behavior (read only, read/write) as well as device characteristics.
- *Movement data*, indicates the current as well as the historical locations, i.e., x, y coordinates of the device/user in time. Movement data can be part of the user profile and/or the device profile.

Essential metadata characterizing the data kept on a PMO.

- Excerpts of the profile data, i.e., data that has to do with the condition, and properties of each PMO.

- Abstractions of the content data, i.e., data about the content data. For example, PMO X contains historical information about Acropolis.

In the following sections, we define the above three categories of data in more detail. A more in-depth treatment of these categories can be found in the technical report to this work [16].

4. CONTENT DATA

Content data will comprise the bulk of the data stored on a PMO. Since we do not make any restrictions in terms of its type and size, determining the kind of data that is actually stored on various PMOs becomes a challenging task.

4.1 Languages and Tools for Metadata

Our content data scenario faces the same challenges as outlined in the Semantic Web proposal of the W3Consortium [1]. This allows us to use partially the same constructs to denote the metadata.

The constructs to "add" meaning to content data (analogously to Web information) are as follows. *XML* lets everyone create their own tags—hidden labels such as <zip code> that annotate textual information. XML allows users to add arbitrary structure to their documents but says nothing about the structures' meaning [2]. Meaning is expressed by using *RDF*. It is encoded in sets of triples ("subject, verb and object"). In RDF, a document makes assertions that particular entities (people, Web pages or whatever) have properties (such as "is a sister of," "is the author of") with certain values (another person, another Web page). Such triplets are denoted using XML tags. In RDF, subject (entities) and object (values) are each identified by a Universal Resource Identifier (URI) (e.g., a link to a Web page). Using URIs allows for the definition of collections of concepts somewhere on the Web (name spaces). See [6] for a comprehensive RDF example. To complement RDF, RDF Schema (RDFS) denotes templates of RDF documents, i.e., an RDF document is an instance of an RDFS document. An *ontology* formally defines the relations among terms, which, following the specific terminology, are referred to as classes of objects [1],[5],[7]. A means to denote ontologies is RDFS [17].

The above constructs are only exemplary for the number of approaches that exist to denote metadata. In the next section, we will give a simple example ontology for content data.

4.2 An Example Ontology

The following example ontology contains some basic classes that might be used in a DB-Globe content data ontology. The ontology as shown in Figure 2 was devised using the ontology editor Protégé 2000 [17]. It comprises classes (marked with a "C"), instances (marked with an "I"), 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*. All these 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.

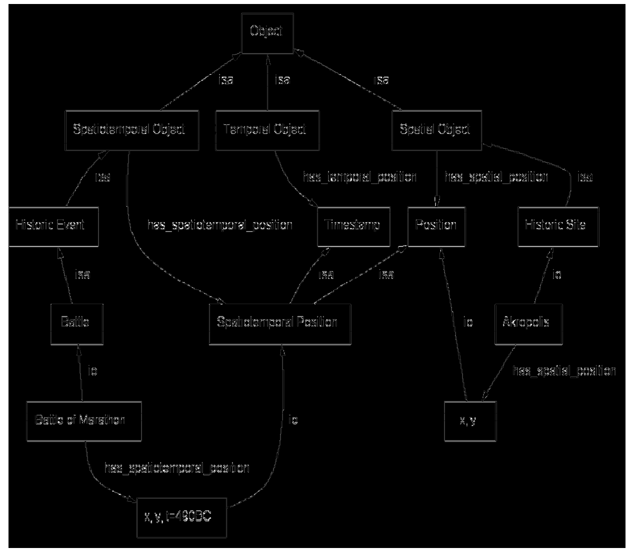

Figure 2: An example DB-Globe content data ontology

Historic Event and *Historic Site* are subclasses of *Spatiotemporal Object* and *Spatial Object*, respectively. Examples of instances are the object *Battle of Marathon* (of the subclass *Battle*) and *Akropolis*. The former has a reference to a spatiotemporal coordinate (x, y, 490 BC; x and y are used instead of real coordinates) whereas the latter references only a spatial coordinate. Introducing further classes allows us to build a more comprehensive ontology covering more and more of the semantic spectrum of content data.

Figure 3 gives the RDF Schema (denoted in XML, with important labels and keywords in bold face) for denoting the *Historic Site* part of the above ontology. In the document several namespaces are used, referencing RDF and RDFS language constructs as well as the newly defined classes and instances subsumed under *DBG* (for DB-Globe). The document defines three classes, *Historic Site*, its superclass *Spatial Object*, the class *Position* and the slot *has_spatial_position* tying the latter two classes together. These RDFS statements formalize the relationships among the respective classes shown in the diagram of Figure 2. Figure 4 shows the definition of instances denoted in RDF and using XML. *Akropolis* is an instance of the class *Historic Site* (instance label 00056). The position of Akropolis is denoted in the instance 00089 and referenced by the slot (relationship) *has_spatial_position* (cf. also Figure 2). Instead of actual coordinates, we use the abstract values "x" and "y."

5. PROFILE DATA

The second important data component of a PMO describes the device itself as well as its user. Its essential parts are the user and the device profile. Further, in the DB-Globe context, movement data is an essential part of each of these profiles. Thus, we will treat it separately in the following.

5.1 User Profiles

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 creating a *user profile*. It represents the choices and the needs of each individual user so that (i) the mobile device behaves in a way desired by the user and (ii)

```
<rdfs:Class rdf:about="&DBG;Historic Site"
     rdfs:label="Historic Site">
    <rdfs:subClassOf rdf:resource= "&DBG;Spatial
Object"/>
</rdfs:Class>
<rdfs:Class rdf:about="&DBG;Spatial Object"
     rdfs:label="Spatial Object">
    <rdfs:subClassOf rdf:resource="&DBG;Object"/>
</rdfs:Class>
<rdfs:Class rdf:about="&DBG;Position"
     rdfs:label="Position">
    <rdfs:subClassOf rdf:resource="&rdfs;
Resource"/>
</rdfs:Class>
<rdf:Property
rdf:about="&DBG;has_spatial_position"
     a:maxCardinality="1"
     rdfs:label="has_spatial_position">
    <rdfs:range rdf:resource="&DBG;Position"/>
    <rdfs:domain rdf:resource="&DBG;Spatial
Object"/>
</rdf:Property>
```

Figure 3: RDF Schema denoting the *Historic Site* part of the ontology

```
<rdf:Description rdf:about="&DBG;00056"
     DB_Globe:Site_ID="Akropolis"
     rdfs:label="Akropolis">
    <DB_Globe:has_spatial_position
rdf:resource="&DBG;00089"/>
    <rdf:type rdf:resource="&DBG;Historic Site"/>
</rdf:Description>

<DB_Globe:Position rdf:about="&DBG;00089"
     DBG:spatial_coordinate="x, y"
     rdfs:label="x, y"/>
```

Figure 4: RDF statement denoting the *Akropolis* instance

information of interest is *forwarded* to the user in both *synchronous* (pull) and *asynchronous* (push) modes. In both cases the *location* of the user and the *time* are essential features and are taken into account.

The user profile can be: (i) explicitly defined by the user and (ii) implicitly be modified by a data mining module that takes the demographic data of the user and his/her behavior patterns into account, where behavior patterns can be categorized into (a) spatiotemporal behavior (i.e., the user motion patterns in space through time) and (b) previous choices that the user has made regarding information access.

5.2 Device Profile

All data that characterizes the PMO will be stored in the device profile. We aim at capturing (i) the characteristics of the device itself, e.g., screen size and (ii) the characteristics of the device with respect to the DB-Globe system, e.g., availability.

The downside of mobile devices when compared with wired desktop machines is in terms of small screen size, small memory, limited keyboard, low processing capability and so on. Hence considerations for content developed for mobile devices are very different as compared to desktop machines. The same content might look and behave differently on different devices. A solution to this is to have the content adapted for the target device. A framework like CC/PP [3] for the management of device profiles is intended to provide the information necessary to adapt the content and the content delivery mechanism to best fit the capabilities of device and preferences of the user.

Moreover, devices have to store data necessary to interact with the DB-Globe system. Such 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 (cf. Section 2). Overall, the number of device parameters will increase as we gain a more in-depth technical knowledge of the project, i.e., by means of a prototype implementation.

5.3 Movement

An important property of the device in connection with mobility and related applications is its movement. In the example ontology of Section 4, we modeled spatiotemporal information. However, spatiotemporal data stemming from *moving objects* is more complex and deserves further attention. The following sections give an account of the structure of these data as well as a metadata proposal in the form of a mobile ontology.

5.3.1 Movement Data

To record the movement of, e.g., a mobile terminal and/or its user, we need to know its position at all times, i.e., on a continuous basis. Practically, we can only obtain the position at discrete instances of time such as every few seconds. By later on interpolating these samples, we can construct the movement of the object. The simplest approach is to use linear interpolation, as opposed to other methods such as polynomial splines [14]. The sampled positions then become the end points of line segments of polylines, and the movement of an object is represented by an entire polyline in three-dimensional space (two spatial and one temporal dimension). In geometrical terms, the movement of an object is termed a *trajectory* [14].

Given this representation, properties of the trajectory and thus movement can be derived, e.g., speed, heading, covered area, etc. We initially distinguish five basic relationships of the trajectory with its environment; *stay within*, when the trajectory is all the time in the range of interest, *bypass*, when the trajectory passes by the area of interest, *leave*, when the trajectory leaves the area of interest, *enter*, when the trajectory enters the area of interest, and *cross*, when the trajectory crosses the area of interest.

Further, the trajectory exhibits relationships to other trajectories and/or to its (spatial) environment. Additionally, relevant positions among trajectories need to be registered at time points. The most common ones based on topological reasoning [15] are *intersect*, indicating that two trajectories intersect, *meet*, showing that two trajectories touch at more than one point, *equal*, when two trajectories coincide, *near*, when two trajectories are close to each other, based on definitions on what "close" means, and *far*, when two trajectories are away from each other.

The next section describes the way trajectory data and relationships can be combined to form a mobile ontology.

5.3.2 A Basic Mobile Ontology

Figure 5 captures the basic mobile ontology by using UML [18]. This is by no means an exhaustive description of an ontology for the mobile world. However, it serves as a basic ontology that is based on the trajectory concept [14],[15]. It includes the relationships of trajectories to their environment as well as their relationships with respect to each other. This basic ontology can be

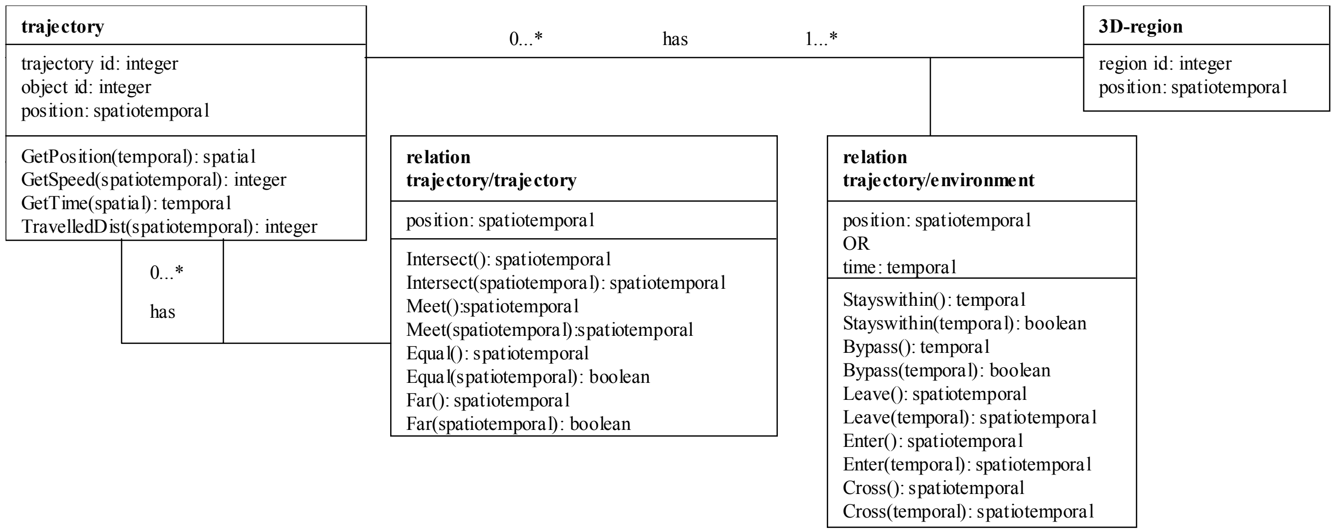

Figure 5: A basic mobile ontology denoted in UML

used to define more specialized classes, e.g., for particular types of devices such as palmtops, mobile phones, etc.

The diagram shows classes and their relationships. Operations denote derived (and frequently updated) values. Figure 5 exhibits four major classes, trajectory, 3D-framework, relation trajectory/trajectory, and relation trajectory/3D-framework. To describe a *trajectory,* we need an identification of the mobile device (indicated by "object id"), the actual trajectory ("trajectory id") as well as the position of the trajectory itself. In other words, "position" describes the trace of the moving object. The data types used are abstract since they only should indicate the dimensionality of the parameter. More concrete instances of data types can be found in, e.g., [9]. A set of operations, e.g., GetSpeed(spatiotemporal), GetTime(spatial), and Travelled Distance(spatiotemporal), are prototypical to show what type of information can be derived from the trajectory data, e.g., to compute the traveled distance of a trajectory, we apply an operation that uses a spatiotemporal range as a parameter.

The *3D-region* class is prototypical to denote the environment of the trajectory. Real-world examples of 3D-region instances are lakes, buildings, borders, etc. The "position" property indicates the extent of the region over time, e.g, a building existed from 1957 until 1999. Trajectories "have" (one or more) relationship either with other trajectories, or their environment, in our case the 3D-region class. Figure 5 contains in the respective classes functions to compute such relationships. E.g., *Leave* without parameter computes the spatiotemporal positions at which a trajectory left a given instance of a 3D-region class. To restrict the operation, we can use an argument to the function. In the case of Leave it is a temporal argument, i.e., it restricts the search to a given time interval. In the class *relation trajectory/environment* the parameter "position" or "time" capture the result of the function. Equally, so does "position" in *relation trajectory/trajectory*.

Figure 5 depicts only a basic ontology comprising trajectories and 3D-regions and related relationship classes. Further classes could be 3D-lines (e.g., road-networks in time) with relations such as, e.g., moving along, etc.

6. ESSENTIAL METADATA

Essential metadata is used in the global computing environment to create an image of the PMO in the DataStore. We have to be content with communicating an abstraction of the data, since a perfect image would require the duplication of the PMO data. Essential metadata contains the data about the content data as well as excerpts of the profile data. Figure 6 illustrates the principle behind this information abstraction. On top of this information pyramid is the data with the highest abstraction level, the essential metadata. This information is based on the semantics of the content data, the device profile, and the user profile. In raising the abstraction level, we reduce the detail of information contained in as well as the size of the data. The latter is important to minimize the communications overhead when a device logs on to the system.

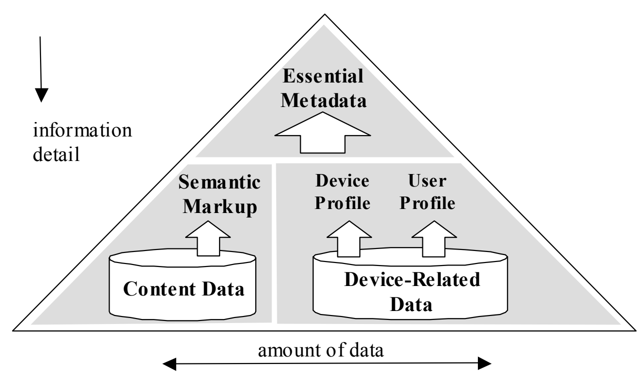

Figure 6: Information abstraction

The question we have to answer is how to practically compose the essential metadata? Is it comparatively easy to decide what parameters to include from the device profile, so is it more difficult to make this choice for the user profile. It consists of semantically richer data, besides personal information about the user, his preferences are stored here as well. Thus, choosing the right abstractions is not that straightforward anymore. However, the semantics of the data, although richer, are still known in advance and thus allow for a conscious decision of what part of the data to include in the essential metadata. The case of content data is more complicated. E.g., assuming we are dealing with a PMO of a history buff, it is not enough to only select dates of events to be included in the essential metadata. What is needed here is an abstraction of the data in the form of, e.g., "historic information of the years 356-322 B.C. relating to the areas comprising the current countries of, among others, Greece, FYROM, Turkey, Egypt, Syria, Iran, Iraq, and India," or alternatively "historic information about Alexander the Great." As one can see, abstractions are neither unique nor accurate. It is desirable to automate finding this abstraction. Assuming, the semantic markup for content data has to be supplied by the user as it is/will be common for existing data sources, we have to be able to automatically abstract the essential metadata. Alternative and more simplistic approaches are for the user to provide *keywords* for the data, or to select predefined *categories* characterizing it. However, because of the importance of "registering" the data with its environment, active user involvement should be minimized.

7. CONCLUSIONS AND FUTURE WORK

Global computing is an approach that relies on a distribution of data over a large amount of data sources, the PMOs. The description of data in terms of metadata is the enabling concept for data discovery and efficient query processing. In the concrete setting, we identify three basic types of data and define related metadata approaches. First, content data represents the bulk of the data stored on PMOs and can be seen similar to the type of data that exist in the current Web context. Second, profile data is introduced to capture the mobility aspect of the device, the user profiles, and the device profile. Third, essential metadata provides an abstract view of the data contained on a PMO. It is used to convey the essential information to the global computing environment.

Directions for future research, if not already indicated previously, are as follows. Essential metadata is the critical piece of information that allows us to register a PMO with the global computing environment. The challenge is of how to generate these data automatically assuming semantic markups exist for all the data that has to be considered. Further, we introduced the community concept in relation to a group of PMOs forming ad-hoc databases. It is important to investigate the possible advantages and disadvantages of this approach, e.g., with respect to query processing. We stated that spatial and temporal data form two special communities. The question is whether they always exist regardless of what other communities exist, i.e., are they orthogonal? We suggested that the mobility aspect changes the metadata approach as it exists for the Semantic Web. A case study, or prototyping is needed to verify this assumption. The notion of a user profile is only introduced in this work. We have to further study its implications with respect to information retrieval and querying.

ACKNOWLEDGEMENTS

This work was supported by the IST-FET project DB-Globe under contract number IST-2001-32645. The authors would like to thank the members of the DB-Globe project for their discussions on the subject.

REFERENCES

[1] Berners-Lee, T., Hendler, J., and Lassila, O.: The Semantic Web, *Scientific American*, May 2001.

[2] Bosak, J. and Bray. T.: XML and the Second-Generation, *Scientific American*, May 1999.

[3] CC/PP Working Group Homepage, *Web page*, http://www.w3.org/Mobile/CCPP/, current as of August 2002.

[4] Chen, J., DeWitt, D., Tian, F., and Wang, Y.: NiagaraCQ: A Scalable Continuous Query System for Internet Databases. In *Proc. of the 1992 ACM SIGMOD International Conference on Management of Data*, pp. 379-390, 2000.

[5] Dornfest, R. and Brickely, D.: The Power of Metadata. The O'Reilly Network. *Web page*, http://www.openp2p.com/pub/a/p2p/2001/01/18/metadata.html, current as of August, 2002.

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

[7] Gruber, Th.: Toward Principles for the Design of Ontologies Used for Knowledge Sharing. In *Formal Ontologies in Conceptual Analysis and Knowledge Representation*. Kluwer Academics, 1993.

[8] IEEE: IEEE Wireless Standards Zone, *Web page*, http://standards.ieee.org/wireless/, current as of August 2002.

[9] Güting, R., Böhlen, M., Erwig, M., Jensen, C. S., Lorentzos, N., Schneider, M., and Vazirgiannis, M.: A Foundation for Representing and Querying Moving Objects. *ACM Transactions on Database Systems* 25(1), pp. 1-42, 2000.

[10] Karakasidis, A. and Pitoura, E.: DBGlobe: A Data-Centric Approach to Global Computing. *DB-Globe Technical Report*, http://softsys.cs.uoi.gr/dbglobe/, 2002.

[11] Lockheed Martin: UML Based Ontology Toolset (UBOT) project, *Web page*, http://ubot.lockheedmartin.com/, current as of August 2002.

[12] Manjoo, F.: P2P Goes in Search of 'Doogle.' *Wired News*, http://www.wired.com/news/technology/0,1282,41850,00.html, 2001.

[13] Pitoura, E.: DB-Globe, A data centric approach to global computing, IST 2001-32645, *Web page*, http://softsys.cs.uoi.gr/dbglobe/, current as of August, 2002.

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

[15] Pfoser, D. and Tryfona, N.: Designing Ontologies for Moving Objects Applications. In *Proc. of the Int'l Workshop on Complex Reasoning on Geographic Data.* Paphos, Cyprus, 2001.

[16] Pfoser, D., Pitoura, E., and Tryfona, N.: Metadata Modeling in a Global Computing Environment. *Technical Report*, DKE-TR002, http://dke.cti.gr/Reports/DKE-TR002.pdf. Data and Knowledge Engineering Group, Computer Technology Institute, Patras, Hellas, 2002.

[17] Protégé 2000 Project Homepage, *Web page*, http://protege.stanford.edu/, current as of August 2002.

[18] Rumbaugh, J., Blaha, M., Premerlani, W., Eddu, F., and Lorensen, W.: *Object-Oriented Modeling and Design*. Prentice-Hall, Englewood Cliffs, NJ, 1991.