DBGlobe
IST-2001-32645
DBGlobe: A Data-Centric Approach to Global Computing

Deliverable D17
Final Report

Covering period 1.1.2002-28.4.2005

Report Version: 2.0
Report Preparation Date: March 17, 2005 – *Revised: June 10, 2005*
Classification: IST
Contract Start Date: 1/1/2002 **Duration**: 40 months
Project Coordinator: University of Ioannina
Estimated Person-Months: 6 **Total Number of Pages**: 168
Partners: Institute National de Recherché en Informatique et en Automatique, Computer Technology Institute, Research Center of the Athens University of Economics and Business, University of Cyprus, University of California, Riverside

Information Society

Project funded by the European Community under the "Information Society Technologies" Programme (1998-2002)

Abstract

The DBGlobe project aims at developing novel data management techniques to deal with the challenges of global computing. This deliverable constitutes the final report of the DBGlobe project. Its overall goal is to present the major achievements of the project. It is structured as follows. Section 1 contains an executive summary, while Section 2 describes the project objectives. In Section 3, we present the methodologies employed to achieve these objectives. The project results and achievements are described in Section 4. Section 4 also includes both an assessment of the work carried out within each individual workpackage and within the project as a whole. Section 5 briefly describes project deliverables, while Section 6 highlights the potential impact of the project results (Section 6.1-6.6) including the project's achievements fiche (Section 6.7). Finally, Section 7 provides a future outlook.

CONTENTS

1. Executive Summary

In this section, we present an overview of the project, highlight its goals and present the partners involved.

1.1 Introduction

An increasing number of devices will soon be able to exhibit sophisticated behavior and interactions with other devices. There will be substantial computing engines in the cell phones and PDAs that we carry. Smart buildings will put computers in light switches, vending machines, and home appliances. Software objects, in the form of mobile software agents, will roam the Internet. This will create a massive infrastructure composed of highly diverse interconnected mobile entities. Novel computing paradigms are needed to model and manage this dynamic environment, often refer to as *global* or *ubiquitous* computing.

Global computing refers to this new form of distributed computing that involves a large number of autonomous computing nodes that cooperate to share resources and services. Nodes normally correspond to computers located at the fringe of the network.

In our research, we view the conglomeration of interconnected nodes carrying data as a virtual super-database with a dynamic schema. In the context of the DBGlobe project, our goal is to manage this super-database. In particular:

> *we develop data management techniques for modeling, indexing and querying data hosted by open-ended networks of massively distributed, highly-dynamic and often mobile nodes*.

To resolve heterogeneity and semantic mismatch problems, we employ a *service-oriented approach*. In DBGlobe, the data of each node are made publicly available through services, that is, access to data stored locally at each node is achieved by invoking an appropriate service. Direct querying of the structure and content of data is also supported by defining services that employ an XML-based query language.

At a second level, metadata, i.e., data describing nodes and their services, are maintained to capture their behavior and state. A uniform XML-based representation for services and metadata is used to facilitate information exchange and sharing. Thus, our focus is on semantic matching of hierarchical service descriptions and data.

The collection of data on devices that exist around a specific context (e.g., location or user) forms a data sharing community that we call an ad-hoc database or simply community. The configuration of such communities varies over time. Communities are designed to support context-specific behaviors such as location-aware queries.

DBGlobe covers all levels of a data management system for data and services hosted by massively distributed autonomous and possibly mobile nodes. In the rest of this report, we present our results along the following topics (correspondingly roughly to the project workpackages):

- infrastructure support provided by our prototype including mobile nodes and the management of communities

- metadata for services and nodes, including location-dependent data
- efficient routing of path queries on XML data, and
- querying that incorporates service calls inside XML documents.

1.2 Overview of Objectives and Main Achievements

DBGlobe proposes a *data-centric approach* to the *design* and *analysis* of dynamic environments of autonomous and mobile entities by considering each mobile entity as a primary data store and a mini-server that protects and encapsulates access to its data.

1.2.1 System Architecture and Prototype Implementation

DBGlobe is a global data and service management system. It connects a number of autonomous, mobile devices and provides support for describing, indexing and querying their data and services (Figure 1.1).

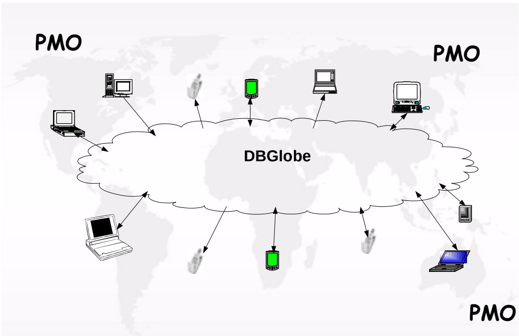

Fig 1.1: The DBGlobe Layer

PMOs connect to the DBGlobe system (and possibly directly to each other) in order to exchange data and perform computations. They register by providing appropriate metadata information. Their number and location may change over time, as new PMOs join or leave the system and existing PMOs relocate.

DBGlobe components are grouped into infrastructure components and application-middleware components. Infrastructure components provide the basic functionality of the system, such as ubiquitous connectivity to devices, while middleware components support semantic and application related system services, such as the creation of communities**.**

Metadata information about PMO devices, users and their data is used to form communities, locate the appropriate services and build complex services. To discover the PMOs that offer an appropriate service, we use distributed indexes based on Bloom filters to route queries to matching PMOs. Information exchanged between PMOs is expressed in the Active XML language (AXML for short). An AXML document is an XML document that includes calls to services.

DBGlobe Infrastructure Components. Besides PMOs, the basic infrastructure components of DBGlobe are the Cell Administration Servers (CASs). Each CAS offers ubiquitous connectivity to devices, captures and stores contextual information and provides basic service publishing and semantic discovery features.

The overlay network of CASs is an essential layer of DBGlobe. The topology of CASs reflects the geographic layout of the system. In particular, we adopt a hybrid (partially ad-hoc) architecture where geographical 2-D space is divided into adjacent administrative areas (as in cellular communication systems) each managed by a Cell Administration Server. The overlay network of CASs constitutes the backbone that makes it possible for the PMOs to communicate and share data and services with each other. In our current design, each cell represents the coverage area of one WLAN access point. We assume that every PMO (including stationary devices) is associated with at most one cell at any given time (e.g., by keeping a live connection to the cell's defining network access point).

Each CAS can independently manage the PMOs which enter its area of authority. It keeps track of the PMOs that enter or leave the cell's boundaries. It stores metadata describing each PMO, the context and the resources offered and assists the user to locate services by semantically matching requests with existing service descriptions. It also provides basic services to visiting PMOs such as network addressing, session management and positioning. Each cell can support large numbers of PMOs moving inside its area and acting as sources or requestors of information.

PMO Components. A PMO is any autonomous, electronic device capable of communicating independently with the CAS via some communication channel. We assume that every PMO has built-in a globally unique identity (like Ethernet adapter addresses or IMEIs in GSM phones) and possibly incorporates components that can capture context (e.g., GPS receiver, digital compass, temperature sensor, etc.). In addition, it may host an application server (e.g., web server) for executing services. When a PMO is a device with limited resources that cannot host a service but still contains resources to be shared, CASs may act as proxies. The corresponding CAS may host the service execution by retrieving the necessary data from the PMO and offering the service to requestors in lieu of the PMO that actually hosts the resources.

Cell Administration Server Components. CASs are interconnected through a network, e.g. the Internet. Although they can function autonomously, they are also aware of their neighbors that manage geographically adjacent cells). The CAS module consists of (Figure 1.2):

- A service ontology that has a hierarchical structure (starting from a universal concept).
- A service directory that lists all the services offered by PMOs in the cell.
- A service description repository of the local services.
- A CAS directory, containing addresses of other CASs.

- A device type and a PMO repository containing the list of device types and PMOs available in the cell and their profiles, and

- A temporal profile manager for storing the connection times of devices, discovering patterns and estimating probabilities of next appearance. A server can also keep historical data and computes statistics about their mobility habits to assist proactive behavior.

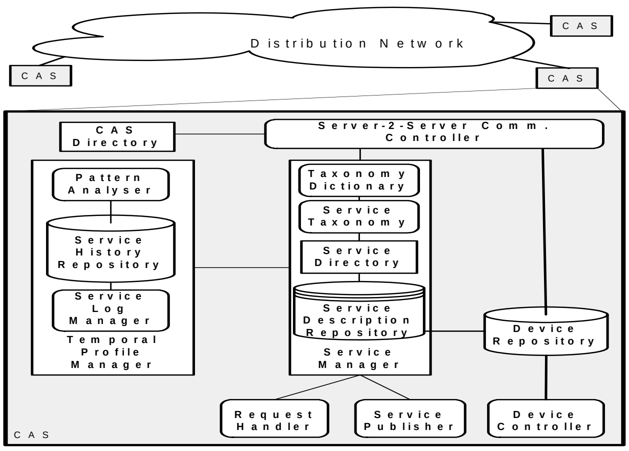

Fig 1.2: Cell Administration Server (CAS) components

Application Middleware and Communities. The basic responsibilities of the middleware are the management of communities and application support for disconnected operation. The major components of the middleware are the: (i) Communities Administrator Servers (CoAS) (ii) UserAgents, and (iii) dynamic query results database.

A community evolves around a semantic concept and may include any combination of spatial, temporal, or thematic characteristics that relate to that concept. The collection of PMOs with data or services related to the specific concept form the community. For example, all PMOs that provide weather services/data may form a "Weather" community. If a PMO needs to know whether it is raining in Athens, it will use the "Weather" community to find PMOs with appropriate services.

Each community is managed by a Community Administration Server (CoAS). A CoAS is the analogue of the CAS for communities. Similar to a CAS, a CoAS keeps track of the PMOs in its community and their related to the community services. It extracts relevant metadata information from the CAS servers and assists the user to locate services by semantically matching requests with existing service descriptions inside the community. Each CoAS keeps also track of other communities in the system.

Communities can be created proactively by users and administrators or in a dynamic manner by gathering (and grouping) metadata information on PMOs, services and users that have matching profiles and goals. CoASs are informed for changes on available services and metadata from the CASs. When a new service is available or a new PMO registers, the CASs propagates, for example, the appropriate metadata to all CoASs.

The dynamic query result database provides caching at the application layer. Intermediate results are stored in the form of views in this database. The goal of UserAgents module is to support network disconnections by cooperating with the dynamic query result database [7]. When a PMO gets disconnected, the UserAgent acts as its intermediary. During disconnection, the UserAgent remains active on the fixed infrastructure and works on the user's behalf (e.g. executing user requested queries). It stores query results at the dynamic query result database to be retrieved by the user upon reconnection. Currently, the dynamic query result database is used only for storing intermediate results during disconnections.

1.2.2 Metadata and Context

Metadata stands for extracted structure and meaning of data. In the DBGlobe context, we encounter content metadata and profile data. The former describes the data contained in the PMOs, whereas the latter describes the user and the device itself.

Content Metadata. 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 are 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 the parameters used in the various services.

Parameter Ontology. Services have signatures (e.g., described in WSDL) that provide the type of the arguments and results. Matching requests with service descriptions is a quite common problem in the literature. The most common approach is matching user requests expressed in keywords with service parameters (i.e., name, location, business, binding or tModel). However, this kind of discovery mechanism does not include semantics. The parameter ontology aims at describing these types of all services in terms of a set of ontologies.

The example of a weather service illustrates our approach. Given a service that returns the weather for a given place and time,
(A) Weather: (location, time ➔ weather)
we want to extend this service to provide weather information along a route,
(A*) Weather_en_route: (route ➔ {location, time, weather}).

Assuming that a service Route(start, end ➔ route) exists that computes a route given a start and an end point, one can combine Weather to obtain Weather_en_route, if the parameters of the respective services match, e.g., Route returns a type route, which can be used as an input parameter for Weather if we know how to decompose a route into locations at times. These relationships can be exposed in parameter ontology.

Service Ontology. Once a service is defined, semantically it is more than the sum of its parts. Knowing the semantics of the parameters of the service is not sufficient to locate a service that fits user queries and reason about the semantics of the service (service discovery). Thus, what is needed besides an ontology relating the parameters is a means to locate or discover and relate services, a service tree (including cross links). The construct to realize such a structure can be in the form of ontology.

We propose a hierarchical structure in which semantically similar services are related to the same node in the tree, e.g., to the topic research. The tree forms a specialization relationship in that a child node contains more specialized services, e.g., a child of a "Research" node could be "Medical Research". Each node in the service tree has a set of attributes (keywords) that describes the domain of the referenced services. As these attributes are more abstract, they do not correspond directly to the types of the service interfaces (cf. parameter ontology). Searching is facilitated by a dictionary that contains lexicographically ordered keywords linked to the respective nodes in the service ontology.

Profile Data. Besides content data we have profile data characterizing the user and the PMO itself. Users 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 device itself, e.g., screen size and (ii) the characteristics of the device with respect to the DBGlobe system.

In connection with mobility and related applications, an important property of the device and the user (and thus part of the respective profiles) is their movement. It allows us to analyze spatial migration patterns, which leads to a better understanding of the PMO distribution in time and to the provision of better location-based services.

We provide a mobile ontology that is based on trajectories, 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 the structural elements introduced by XML and ontologies to denote hierarchies and relationships.

1.2.3 Query Routing

For a service request originating from a PMO, there may exist many sites (PMOs, CASs or CoASs) with matching services or documents. Thus, we need a mechanism to locate which sites contain relevant information efficiently. We adopt a decentralized approach. Each site maintains a data structure, called a filter, that summarizes all documents and services that exist locally. This is called a local filter. Besides its local filter, each site also maintains one or more merged filters that summarize the services and documents that exist in a set of its neighboring sites. These filters facilitate the routing of a query. When a query reaches a site, the site first checks its local filter and then uses the merged filters to direct the query only to those neighboring sites that may contain relevant documents. As our filters, we use Multi-level Bloom filters.

Multi-level Bloom Filters. Bloom filters are compact data structures for probabilistic representation of a set that supports membership queries. Consider a set A = {a_1, a_2, …, a_n} of n elements. The idea is to allocate a vector v of m bits, initially all set to 0, and then choose k independent hash functions, h_1, h_2, …, h_k, each with range 1 to m. For each element a ∈ A, the bits at positions $h_1(a)$, $h_2(a)$, ..., $h_k(a)$ in v are set to 1. A particular bit may be set to 1 many times. Given a query for b, we check the bits at positions $h_1(b)$, $h_2(b)$, ..., $h_k(b)$. If any of them is 0, then certainly b is not in the set A. Otherwise we conjecture that b is in the set although there is a certain probability that we are wrong. This is called a "false positive" and it is the payoff for Bloom filters' compactness. The parameter k and m should be chosen such that the probability of a false positive is acceptable.

Traditional Bloom filters cannot efficiently summarize hierarchically structured data (such as XML documents and XML-based service descriptions). We have proposed an extension of traditional Bloom filter, called multi-level Blooms appropriate for hierarchical documents [5]. We consider two types of multi-level Blooms based on the way XML documents are hashed: Breadth and Depth Bloom Filters.

Let an XML tree T with j levels, and let the level of the root be level 1. The Breadth Bloom Filter (BBF) for an XML tree T with j levels is a set of Bloom filters {BBF_0, BBF_1, BBF_2, …, BBF_i}, $i \leq j$. There is one Bloom filter, denoted BBF_i, for each level i of the tree. In each BBF_i, we insert the elements of all nodes at level i. Depth Bloom filters provide an alternative way to summarize XML trees. We use different Bloom filters to hash paths of different lengths. The Depth Bloom Filter (DBF) for an XML tree T with j levels is a set of Bloom filters {DBF_0, DBF_1, DBF_2, …, DBF_{i-1}}, $i \leq j$. There is one Bloom filter, denoted DBF_i, for each path of the tree with length i (i.e., of i + 1 nodes), where we insert all paths of length i.

Our experimental results show that our multi-level Blooms outperform a same size traditional Bloom filter for processing path queries over hierarchical data

Semantic and Topological Distribution. Multi-level Bloom Filters are used for routing queries at the appropriate sites. Each site (PMO, CAS or CoAS) maintains a local Bloom filter summarizing the local services or documents. In addition, each site maintains a merged filter summarizing the documents for a set of its neighbors.

We consider two ways for determining the set of neighboring sites for which we maintain summaries. One approach is based on locality, and the other on filters similarity. The locality based approach organizes the sites based on their proximity in the physical network (i.e., at the infrastructure level). The motivation behind this organization is to satisfy queries locally and minimize response time. The second approach organizes the sites based on their filter similarity, so as to group relevant sites together, thus supporting communities. The motivation for this organization is to minimize the number of irrelevant sites that process a query, by grouping together sites with relevant content. To achieve this we have defined a metric for deciding how "similar" two multi-level bloom filters are.

1.2.4 Query Processing

The information exchanged between PMOs is expressed in the Active XML (AXML in short) language. An AXML document is an XML document with embedded calls to services. The service calls are represented as particular XML elements. When calls included in an AXML document are fired, the document is enriched by the corresponding results. An AXML PMO contains AXML documents and offers (AXML) services, which can be used to enrich the AXML documents of the same or of other components. While documents with embedded calls have been used before, there is a fundamental difference between AXML and dynamic (XML) Web pages, since services may be invoked from anywhere on the network, calls embedded in an AXML page do not have to be evaluated before sending it.

In some sense, an AXML document can be seen as a (partially) materialized view integrating plain XML data and dynamic data obtained from service calls. As a simple example, consider an AXML document for the home-page of a local newspaper. It may contain some plain XML data, such as general information about the newspaper, and some dynamic fragments, e.g. one for the current temperature in the city, obtained from a weather forecast Web service, and a listing of current art exhibits, obtained from the local TimeOut guide service.

Controlling Service Call Activation. The service calls inside AXML documents are represented by special XML elements carrying the tag <sc> (for service call) which are interpreted as calls to services. An <sc> element encodes enough information to invoke the service on a fixed site. In case of a PMO that does not have a fixed URL, the <sc> element encodes an identification of the PMO that is needed to locate it on the network. Particular attributes of the <sc> element allow to specify when a service call should be activated (e.g. when needed, every hour, etc.), and for how long its result should be considered valid.

Simple services are similar to remote procedure calls. The service is called with some arguments, and eventually returns an answer. For the interesting class of continuous services, the interaction is more complex. Once a service call has been registered, a stream of answers is returned for this single service call. Streams of answers are encountered in many real-life applications: the stream of source updates for the maintenance of a data warehouse, the readings of a temperature sensor or surveillance system, answers returned by continuous queries or publish-subscribe systems. In the case of such continuous services, the activation of the service call encapsulates a service subscription, and all the results received subsequently from the service are integrated in the caller document.

Intensional Parameters and Results. The parameters and result of a service call may themselves be AXML documents. A system component receiving a call with parameter containing service calls may have to activate the calls it includes before actually performing the service. Similarly, a service result may contain further service calls, which have to be activated by the site receiving the result. Thus, service call activations entail exchanging intensional data, and lead to a form of distributed computation.

The use of intensional parameters/results involves security issues. For instance, a malicious user may force a site to perform a dangerous action by calling one of the site's services with an intensional parameter that contains a call to a dangerous service. Besides security, the node capabilities need also to be taken into consideration. We formalized the problem and developed algorithms to solve it.

Defining Services. The AXML framework allows to use arbitrary existing services as well as to define new services on top of the enriched AXML documents. The definition of AXML services relies on parameterized XML queries expressed in XQuery, the standard language for querying XML data, extended with updates. AXML services may query and update AXML or regular XML documents. The AXML service specification allows, in particular, the definition of continuous services, and of services with intensional input/output. One should note that an AXML service, when called by a non-AXML client, will detect the limitations of its caller and in such case will return only extensional answers.

Distribution and Replication. By the sole presence of PMO- and DBGlobe- services, AXML documents already include inherently some form of distributed computation. A higher level of distribution that also allows (fragments of) AXML documents and services to be distributed and/or replicated over several sites is highly desirable in a dynamic mobile architecture: a single XML document may need to be distributed on more than one PMO if the PMO does not have enough storage. Furthermore, replication of services, as well as the data they utilize, is needed, since a PMO will tend to use services "nearby". To address this, we extended the AXML data model with distribution and replication, and provided a location-aware extension to XQuery to handle distributed/replicated data and services.

1.3 Composition of the Consortium

The vision of global computing encompasses nearly all areas in computer science. DBGlobe addresses the related data management issues, thereby representing the database community’s view and contribution to this emerging area. The teams participating in the project are all well-established database research groups. All teams have a common objective: view global computing from a data-centric perspective and define appropriate models, languages and data delivery and co-ordination mechanisms.

Within the field of data management research, each DBGlobe team has complementary expertise.

The *Computer Science Department of the University of Ioannina (UoI)* has a strong research group in data management for mobile and ubiquitous computing. It has extended expertise on co-ordination among activities and on data delivery mechanisms. It brings this knowledge into the projects through its participation in WP2. In addition, the group has a strong background on data replication and has contributed to WP1.

INRIA brings into the consortium expertise in query management. Thus, INRIA leads the work in WP4. INRIA's group has also extensive work in XML and semistructured data models and utilizes this in metadata handling in WP1. INRIA is the main developer of AXML.

The *Computer Technology Institute (CTI)* in co-operation with the *Computer Science Department of the University of Aalborg (AAU)* brings into the consortium expertise on data modelling. The group will participate mainly in WP1. The group has also extensive work in spatio-temporal databases and querying. They provided input in the development of the simulation environment for dynamic mobile entities and on the spatio-temporal aspects of the query language (WP4).

The *Department of Informatics of the Athens University of Economics and Business (AUEB)* has a strong group on spatio-temporal databases and led the work on the development of the mobile object simulation environment (WP2). The group has also expertise in event handling and participated in the development of the co-ordination schema in WP3.

The *Computer Science Department of the University of Cyprus (UCY)* has a strong group on mobile computing and mobile agents. It led the research in the design and development of the proof-of-concept prototype (WP5). It has also participated in WP3 with its background on co-ordination.

The *Data Mining Lab of the University of California, Riverside (UCR)* conducts research in the areas of Data Mining, Databases and Algorithms. More specifically, problems include efficient indexing techniques for moving points, approximating range queries, similarity searching in sequence databases, finding frequent sequences or sets, approximate set indexing, local adaptive classification techniques. The group brought in the project its expertise by participating in WP4 in the development of appropriate indexing techniques for query management.

Figure 2.3 and Table 3.1 summarize the interaction between the project partners.

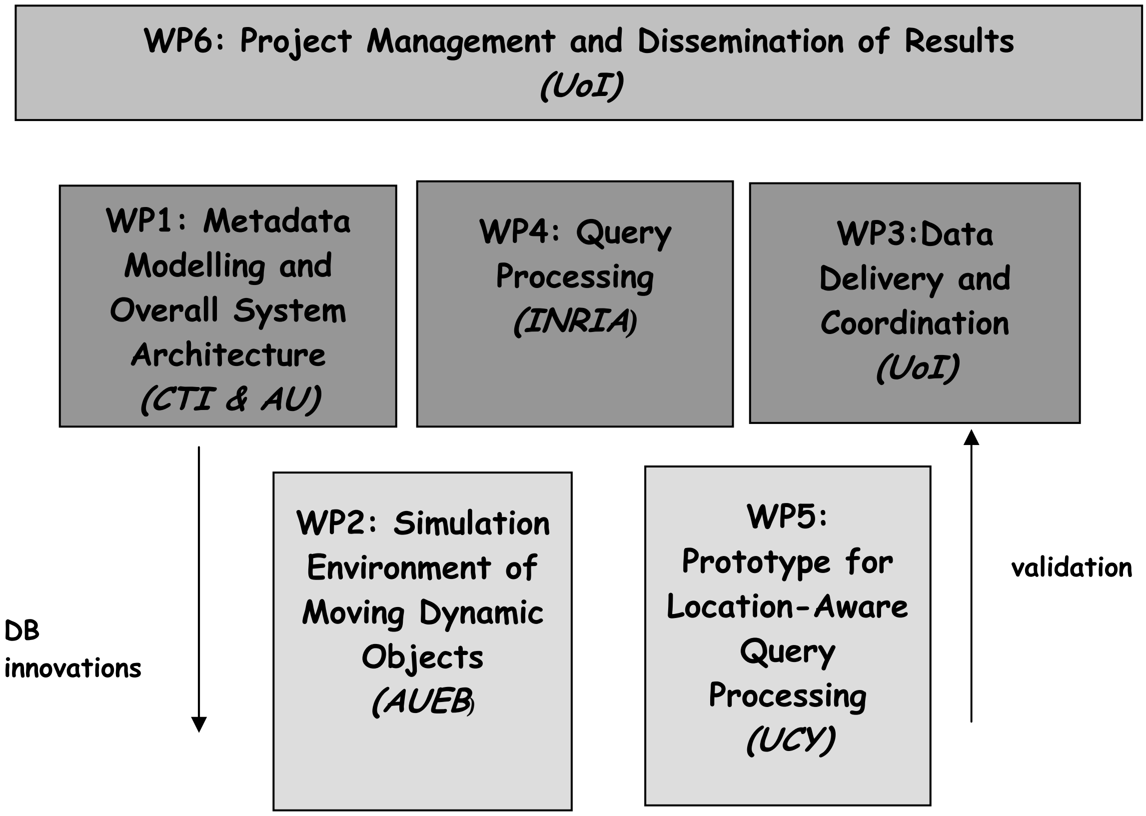

Figure 1.3 Interaction among the project partners and workpackages

Table 1.1 Summary of the European Partners research expertise

Partner	Technical Expertise	Workpackage
UoI	Data Delivery Mechanisms and Activities Co-ordination Data Replication and Distribution	WP3 (leader) WP1
INRIA	Query Processing Semistructured Data Models	WP4 (leader) WP1
CTI & AAU	Conceptual Modelling Spatio-temporal Databases	WP1 (leader) WP2 & WP4
AUEB	Moving object databases Event Handling	WP2 (leader) WP3
UCY	Mobile agent software systems Transaction management	WP5 (leader) WP3

2. Project Objectives

In this section, we present the objectives of the project per workpackage. The following sections include an assessment of the extent of how the objectives were met.

System Architecture (WP1)

The main target in the first workpackage (WP1) of DBGlobe was to derive appropriate architectures for ad-hoc databases of mobile entities. Such architectures should be metadata driven in order to facilitate transparent interoperability among the involved entities, and to enhance location management, which is a critical aspect of a global computing system that relies mostly on mobile distributed entities.

Simulation Enviroment (WP2)

The aim of workpackage 2 (WP2) was to design and implement a virtual testbed of the proposed DBGlobe architecture (WP1), which simulates a real-world dynamic environment of mobile entities able to co-operate with each other. This simulator was subsequently to be extended to a proof-of-concept prototype of DBGlobe (WP5). A detailed description of the simulator is provided in deliverable D6: “Initial simulator prototype and web page”.

Data Delivery and Coordination (WP3)

One important aspect in DBGlobe is the means of communicating data among the components that participate in the system: (a) the mobile entities, (b) the DBGlobe servers and (c) the users. Current database servers support a pull-based mechanism of data delivery: data are sent to clients after having been explicitly requested. This request-response time of data delivery is not appropriate for all interactions in the DBGlobe system. Some of the mobile entities deliver information in a push-based fashion, that is, without an explicit request. Both push and pull data delivery may be periodic or aperiodic. For instance, for push delivery, data may be sent, when a condition is met (e.g., the refrigerator notifies the user’s PDA, that there is no milk) or periodically (a sensor reports measurements of the ozone level every hour). Finally, data may be received either by a single component or by a group of components. DBGlobe’s data delivery mechanism must incorporate:

- Push and pull data
- Periodic and aperiodic
- Multicast and unicast.

In global computing, data are delivered from sources to clients in a variety of modes. In Deliverable D8 we have studied the properties of the various data delivery mechanisms and how they can be deployed in the overall DBGlobe framework.

We classified the various data delivery mechanisms based on the following four basic characteristics: (1) how is data delivery initiated (push vs. pull), (2) the frequency of the delivery (periodic vs. aperiodic), (3) the receivers of the data delivery (unicast vs. broadcast) and (4) the unit of interaction (data vs. query shipping). Various combinations along these four dimensions are possible. An interesting such combination is data diffusion in which an interest (query) for a data item is periodically injected (pulled) in the network and then broadcasted to the appropriate nodes for evaluation.

A central issue in data delivery is ensuring that the clients receive consistent data. We developed a formal framework for expressing the correctness properties of data delivery and present protocols that ensure both the temporal and the semantic coherency of data for the various data delivery mechanisms. By temporal coherency, we refer to the currency related properties of the data sets read by a client, e.g., when were the data items read by a client current at the server? Semantic coherency refers to transaction-based consistency properties, e.g., do the data items read by a client correspond to a "serializable" execution of the server transactions? Without both semantic and temporal coherency, clients may be seeing either outdated data sets or data sets that correspond to inconsistent database states.

We showed how such mechanisms fit into the DBGlobe architecture as defined in DBGlobe's "Deliverable D3: Overall System Architecture" and present directions regarding an adaptive deployment of each mechanism.

Information Discovery and Querying (WP4)

The basic objective of workpackage 4 (WP4) was to derive methods and techniques for querying and discovering knowledge in environments consisting of autonomous, dynamic, and mobile entities. The first fundamental challenge is choosing an appropriate data model to describe the information available to and offered by, these dynamic entities, and designing a query language and a query execution paradigm for this context. Towards this end, we have designed and implemented AXML, a language that incorporates service calls within XML documents.

Proof–of-Concept Prototype (WP5)

The aim of workpackage 5 (WP5) was to develop a proof of content prototype that verifies the various concepts developed within the DBGlobe project. In summary, the prototype is used:

- To demonstrate through a concrete example the idea of ad-hoc databases and the various forms of querying
- To provide a concrete example of the workflow interaction model and the different data delivery mechanisms
- To capture, design and implement a location-aware system

Regarding the functional specifications of prototype, the prototype should provide an interface that will allow users to:

1. Register, connect and disconnect to the system. Each user must first connect to the DBGlobe system in order to be able to use its services.

2. Define shared resources directory. The DBGlobe system will use any kind of stored information that exists on user devices. The DBGlobe system must allow each user to define a shared resource directory on the mobile device. All information the user gets from the system must be stored there in order for other user to be able to get it.

3. Create and login in a community. All users will join the DBGlobe system via their mobile devices. They will be able to join a community in order to retrieve and query data.

4. Define criteria for user's profile. Information discovery and query capability will use user's profile (or a similar concept) in order to get information or services. Each user has a profile in the system in order to serve that purpose. Moreover the user will be able to define his/her own criteria about the type, content and characteristics of information he/she wants to get.

5. Make a search for available services. Since user changes his location the available services may be different from point to point. The prototype must include an interface in order to inform or help users about the available services.

6. Broadcast or multicast information. The example prototype will broadcast or multicast information to the users which are in a location, belong to a "community" or define the exact information to be broadcast in their device independently of their location.

7. Co-ordinate and co-operate with other mobile devices. Each mobile device with storage capability is a mobile entity. A group of mobile entities can create a virtual or physical "community". In the DBGlobe system those mobile entities must be able to co-ordinate and co-operate in order to discover and query their data. The example prototype will use this system capability.

8. The prototype should provide all types and formats of information the system supports. For example the data that each mobile entity stores will vary from images to stream data.

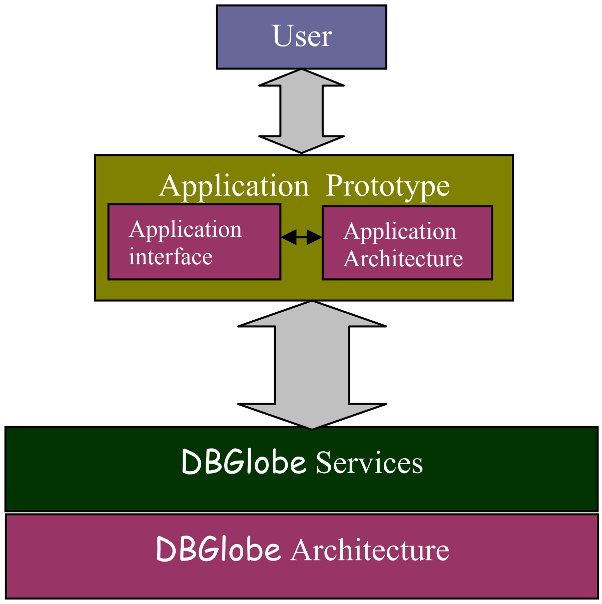

Fig. 2.1 DBGlobe System

As we progressed in designing and developing our application we have further enhanced and understand the needs of the DBGlobe system and architecture as reported in the specific deliverables. Figure 2.1 positions the prototype with regards to the DBGlobe architectrure.

Project Management and Dissemination of Results (WP6)

This work package was responsible for the co-ordination of the project in both administrative and technical terms aiming towards achieving effective operation of the project, as well as timely delivery of the results.

The objectives were:

- To produce quality deliverables
- To comply with the reporting requirements outlined in the Contractual Agreement entered into the European Community,
- To organize a number of meetings and form working groups
- To distribute the research results of the project by publishing at competitive journals and conferences and creating a web page for the project,
- To make the results of the project visible in the research (and industry) community by presenting our work in many scientific and industrial forums,
- To disseminate the simulator
- To continously observe and assess the areas of impact

- To describe the activities related to the integration of DBGlobe project with the other project of the Global Computing

3. Methodologies

3.1 State of the Art

A detailed comparison of the DBGlobe project specific technical contributions (e.g. in metadata modeling, querying, service discovery) with related current research can be found at the associated deliverables and publications.

In addition, Section 3.2 provides a survey that compares the overall DBGlobe approach with related current research. Section 3.3 provides a comparison of the methodologies used in DBGlobe with other projects within the Global Computing initiative.

3.2 System Architectures for Global Computing: A Comparative Survey

3.2.1. Introduction

In this section, we survey issues and systems related to global computing. *Global computing* refers to a new form of distributing computing which involves numerous, casually accessible and often invisible computing devices interconnected through a possibly heterogeneous network structure. These devices are frequently mobile, or embedded in the environment.

Recently global computing has attracted considerable attention, since it is expected that in the near future, an increasing number of devices will be able to exhibit sophisticated behavior and interactions with other devices. There will be substantial computing engines in the cell phones and PDAs that we carry. Smart buildings will put computers in light switches, vending machines, and home appliances. Software objects, in the form of mobile software agents, will roam the Internet. This will create a massive infrastructure composed of highly diverse interconnected mobile entities. *Novel system architectures are needed to model and manage this dynamic environment.*

The distinction between global computing and a number of other recently emerging forms of distributed computing, such as peer-to-peer or grid computing, is often vague. Table 3.1 emphasizes the distinctive characteristic of each of these areas.

The systems and frameworks surveyed in this report are classified into two main categories: (i) distributed and (ii) data-centric systems. In the *distributed systems* category, we include systems that aim at providing the basic framework that is required for supporting the functionality for the services offered by a generic global computing system. Whereas, *data-centric systems* are designed to deal with the manipulation of information which is disseminated among the various data sources; their main focus is on information retrieval and data processing. Furthermore, while the programming model in distributed systems is procedural, data centric systems support declarative query languages.

3.3.2. Distributed Systems

One of the main issues addressed by distributed systems research in global computing is system and network heterogeneity. Another important aspect is mobility, since many of the devices participating in such systems, such as wearable devices, mobile phones, or PDAs, are mobile. This fact creates the need for systems whose services will support mobility of system components or clients. Furthermore, distributed systems for global computing must support by definition a large number of appliances. Thus scalability is central in such systems.

We further classify the distributed systems presented in this report in the following three subcategories:

- *programming frameworks* that are infrastructures that provide the means, tools and protocols, for building distributed applications on top of them.

- *distributed applications* that are global computing systems with a very specific functionality; these systems are in use today and are gaining increasing popularity, and

- *grid computing systems* that follow a computational grid approach and are included here for comparison.

Table 3.1: Areas Related to Global Computing

System Type	Description	Characteristics
Peer-to-peer	A communications model in which each party has the same capabilities and either party can initiate a communication session [36, 37, 38, 39, 40, 41, 42]	• Usually simple file sharing • Centralized authorities maintain indexing information (hybrid peer -to -peer systems), or no centralized authorities exist (pure peer -to -peer systems)
Grid computing (or the use of a *computational grid*)	Applying the resources of many computers in a network to a single problem at the same time - usually to a scientific or technical problem that requires a great number of computer processing cycles or access to large amounts of data [43, 44, 45, 46, 47]	• Solution of a specific problem • Large amounts of data and processing power • No need for mobility • Powerful machines and interconnections • Large scale distributed system
Pervasive or Ubiquitous Computing (*calm technology*)	The strongly emerging trend toward: Numerous, casually accessible, often invisible computing devices Frequently mobile or imbedded in the environment Connected to an increasingly ubiquitous network structure.	• Mobility • Small appliances • Data & services everywhere • Uniform access • Integration with human users

	[48, 49, 50, 51, 52]	• Large scale distributed system
Metacomputing	A collection of computers held together by state-of-the-art technology and "balanced" so that, to the individual user, it looks and acts like a single computer. The constituent parts of the resulting "metacomputer" could be housed locally, or distributed between buildings, even continents. [53, 54]	• Uniform access • May use **Grid Computing** to accomplish specific tasks • Large scale distributed system • Heterogeneity • Multiple administrative domains
Semantic web	An intelligent version of the World Wide Web where context-understanding programs will help users find what they want, by using self-descriptions and other techniques. [55, 56, 57]	• Uniform metadata representation for resources • Extension of the current World Wide Web
Invisible Computing	Enhances existing everyday devices and makes them smarter without requiring extra human interaction. In this domain the computer is not the main focus but rather the device itself or the specialized function it performs. [58, 59, 60]	• Integration in everyday appliances • No large scale distributed system
Stream Computing	Focuses on real – time processing of continuous and sometimes endless flows of data, for analysis and interpretation, without having to receive the entire block. [61, 62, 63, 64]	• Sensors • Backbone for processing data streams

3.2.2.1 Programming Frameworks

Distributed programming frameworks are frameworks designed to assist the process of building global computing systems. They provide the necessary infrastructure, and a programming model for building a distributed system. They offer protocols and methods so that a specific application can be built on-top of them; thus, more emphasis can be put on developing sophisticated applications without the need to design the framework below. These frameworks are easily deployed and extensible to fit the requirements of each specific application.

We present four programming frameworks: One.World, Globe, Jini and VIA.

One.World

One.World [1, 2] provides an integrated and comprehensive framework for building pervasive applications. Its programming model separates data and functionality. Tuples represent data and components implement functionality. It includes a set of services such as service discovery, checkpointing and migration.

System Components

One.World programming model relies on separate abstractions for application data and for functionality. Applications store and communicate data in the form of tuples and are composed from components. *Tuples* are records with named fields and are self-describing in that an application can dynamically determine a tuple's fields and their types. *Components* implement functionality. *Environments* serve as containers for tuples, components, and other environments, providing the system with structure and control.

System Functionality

Each device typically runs a single instance of one.world. Each such *node* is independent of other nodes and may be administered separately. Applications run within one.world, and all applications running on the same node share the same instance of the architecture. The architecture provides the same basic abstractions and core services across all nodes and uses mobile code to provide a uniform and safe execution platform.

Components implement functionality and interact by importing and exporting *event handlers*. They statically declare which event handlers they import and export but are dynamically linked and unlinked. *Environments* provide structure and control. They serve as containers for tuples, components, and other environments. Each application has at least one environment, in which it stores tuples and in which its components are instantiated. Applications are not limited to a single environment and may span several, nested environments. The root environment of each node hosts one.world's kernel. Environments are also an important mechanism for dynamic composition: an environment controls all nested environments and can interpose on their interactions with the kernel and the outside world. Environments thus represent a combination of the roles served by file system directories and nested processes in more traditional operating systems. Figure 3.1 shows an example environment hierarchy. *Leases* control access to both local and remote resources. They limit the time applications can access resources, such as an environment's tuple storage or a communication channel, and force applications to periodically renew their interest in the resources. As a result, leases make time visible throughout the system and cleanly expose change to applications.

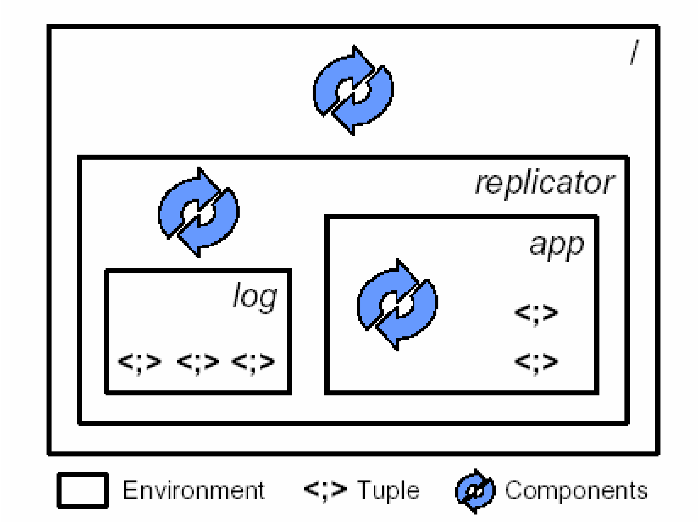

Fig. 3.1: An example of one.world hierarchy

One.world provides a set of services that serve as common building blocks and directly help developers in making their applications adaptable. *Migration* provides the ability to move or copy an environment and its contents to another node. It affects an entire application, because both components and stored tuples are moved or copied. *Remote event passing* (REP) provides the ability to send events to remote receivers, including receivers located by service discovery. *Replication* makes stored tuples accessible on several nodes at the same time, even if the nodes are not currently connected. Finally, *check pointing* captures the execution state of an environment tree and saves it as a tuple, thus making it possible to later revert the environment tree's execution state. The *Remote Event Passing* mechanism (REP) provides the ability to send events to remote services and is one.world's basic mechanism for communication across the network. *Discovery* allows the detection of the location of a service.

One.world's discovery mechanism provides a set of options for performing resource discovery tasks. The discovery service routes events to services with unknown location. It supports early and late binding as well as multicast, and is integrated with REP. One.world's discovery mechanism relies on a central server.

Globe

Globe [3, 4] is a wide-area distributed system. Its major objectives are: (i) to provide a uniform model for distributed computing, (ii) support a flexible implementation framework, and (iii) ensure worldwide scalability. Its architecture is based on distributed shared objects.

System Components

Globe is a system with an object-oriented architecture. The main entity is *a distributed shared object* which is a collection of local objects. *Local objects* are located at a single address and implement the distributed object's functionality by communicating with each other over the network. A local object is designated as replica, if it stores part, or all of the distributed object's state. Each local object consists of sub–objects, which implement system functions.

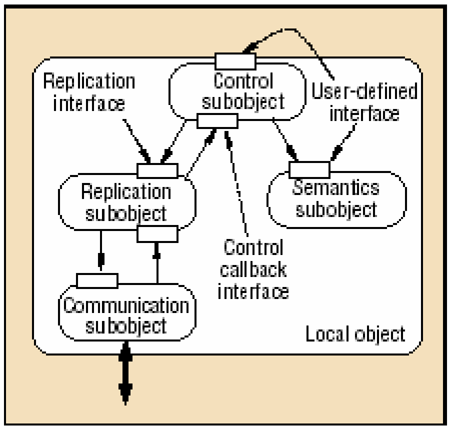

Fig. 3. 2: A Local Object

The following four sub-objects are defined, as shown in Figure 3.2:

- A *semantics sub-object,* containing the methods that implement the distributed shared object's functionality. It is the only sub-object that a developer has to implement.
- A *communication sub-object,* which handles the communication among local objects
- A *replication sub-object,* which is responsible for keeping the distributed object's state and designating a specific replication policy.
- A *control sub-object,* handling the control flow within the local object.

These four sub-objects are designed to build scalable distributed shared objects.

System Functionality

In Globe, processes interact and communicate through distributed shared objects. A Globe object is physically distributed, meaning that its state might be partitioned and replicated across multiple machines at the same time. A process wishing to invoke an object's method should initially use Globe's naming service to get the object's unique identifier. Having this identifier, it can locate the object using Globe's location service. The location service returns the available contact addresses for the process to communicate with the distributed object. A contact address consists of a communication protocol and a network address of the distributed object's access point. The process selects one of the available access points and binds to the distributed object. Binding results in an interface belonging to the object being placed in the client's address space, along with an implementation of that interface. Such an implementation is called a local object.

A distributed object, as shown in Figure 3.3, is built by local objects, which reside in a single address space and communicate with local objects in other address spaces.

They form a particular implementation of a distributed object's interface. For example, a local object might implement an interface by forwarding all method invocations, as in RPC client stubs. A local object in another address space could implement the same interface through operations on a replica of the object's state. Local objects are constructed in a modular way, to separate issues such as replication and communication from what the object actually does i.e. its semantics.

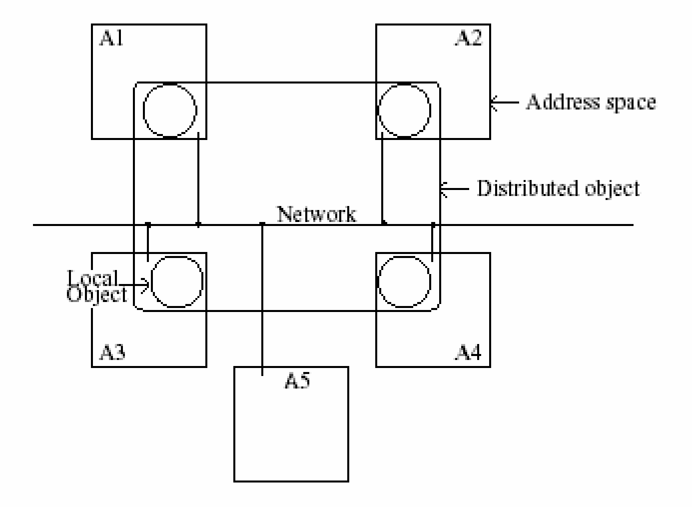

Fig. 3.3: A Distributed Object

Jini

A Jini system [5] is a distributed system based on the idea of federating groups of users and the resources required by those users. The overall goal is to turn the network into a flexible, easily administered tool on which resources can be found by human and computational clients. Resources can be implemented as either hardware devices, software programs, or a combination of the two. The focus of the system is to make the network a more dynamic entity that better reflects the dynamic nature of the workgroup by enabling the ability to add and delete services flexibly.

System Components

A Jini system consists of the following parts:

- A set of components that provides an infrastructure for federating services in a distributed system.
- A programming model that supports and encourages the production of reliable distributed services.
- Services that can be made part of a federated Jini system and which offer functionality to any other member of the federation.

While these pieces are separable and distinct, they are interrelated, which can blur the distinction in practice. The components that make up the Jini technology infrastructure make use of the Jini programming model; services that reside within the infrastructure also use that model; and the programming model is well supported by components in the infrastructure.

The most important concept within the Jini architecture is that of a *service*, which is an entity that can be used by a person, a program, or another service. A service may be a computation, storage, a communication channel to another user, a software filter, a hardware device, or another user. Members of a Jini system federate in order to share access to services. A Jini system consists of services that can be collected together for the performance of a particular task. Jini systems provide mechanisms for service construction, lookup, communication, and use in a distributed system. Services communicate with each other by using a *service protocol*, which is a set of interfaces written in the Java programming language. The base Jini system defines a small number of such protocols, which define critical service interactions.

Services are found and resolved by a *lookup service*. The lookup service is the central bootstrapping mechanism for the system and provides the major point of contact between the system and users of the system. In precise terms, a lookup service maps interfaces indicating the functionality provided by a service to sets of objects that implement the service. In addition, descriptive entries associated with a service allow more fine-grained selection of services based on properties understandable to people. Objects in a lookup service may include other lookup services; this provides hierarchical lookup.

The Jini technology infrastructure defines the minimal Jini technology core. The infrastructure includes the following:

- A distributed security system, integrated into RMI, which extends the Java platform's security model to the world of distributed systems.
- The discovery/join protocol, a service protocol that allows services (both hardware and software) to discover, become part of, and advertise supplied services to the other members of the federation.
- The lookup service, which serves as a repository of services. Entries in the lookup service are objects in the Java programming language; these objects can be downloaded as part of a lookup operation and act as local proxies to the service that placed the code into the lookup service.

System Functionality

The heart of the Jini system is a trio of protocols called *discovery*, *join*, and *lookup*. A pair of these protocols (discovery/join) occurs when a device is plugged in. Discovery/Join is the process of adding a service to a Jini system. A service provider is the originator of the service. First, the service provider locates a lookup service by multicasting a request on the local network for any lookup services to identify themselves. Then, a service object for the service is loaded into the lookup service (join). This service object contains the Java programming language interface for the service including the methods that users and applications will invoke to execute the service, along with any other descriptive attributes.

A client locates an appropriate service by its type along with descriptive attributes which are used in a user interface for the lookup service. The service object is loaded into the client. The service object's methods may implement a private protocol between itself and the original service provider.

Some other important features of a Jini system are:

- *Java Remote Method Invocation:* Communication between services can be accomplished using *Java Remote Method Invocation* (RMI[tm]). RMI provides mechanisms to find, activate, and garbage collect object groups. Fundamentally, RMI is a Java-programming-language-enabled extension to traditional remote procedure call mechanisms. RMI allows not only data to be passed from object to object around the network but full objects, including code.
- *Security:* The design of the security model for Jini technology is built on the twin notions of a *principal* and an *access control list*. Jini services are accessed on behalf of some entity. Services may request access to other services based on the identity of the object that implements the service. Whether access to a service is allowed depends on the contents of an access control list that is associated with the object.
- *Leasing:* Access to many of the services in the Jini system environment is *lease* based. A lease is a grant of guaranteed access over a time period. Each lease is negotiated between the user of the service and the provider of the service as part of the service protocol.
- *Transactions:* A series of operations, either within a single service or spanning multiple services, can be wrapped in a *transaction*. The Jini Transaction interfaces supply a service protocol needed to co-ordinate a *two-phase commit*.
- *Events:* The Jini architecture supports distributed *events*. An object may allow other objects to register interest in events in the object and receive a notification of the occurrence of such an event. These enables distributed event-based programs to be written with a variety of reliability and scalability guarantees.

The end goals of the system span a number of different audiences; these goals include the following:

- Enabling users to share services and resources over a network
- Providing users easy access to resources anywhere on the network while allowing the network location of the user to change
- Simplifying the task of building, maintaining, and altering a network of devices, software, and users.

VIA/VIA*

VIA (Verified Information Access) [6, 7] is an overlay network maintained by an application-level protocol based on the principles of robustness, self-organization, scalability, simplicity, and local autonomy.

System Components

According to the VIA architecture the master directory service resides on a “gateway” that connects a local discovery domain to a “main channel” allowing communication between gateways. Gateways are organized into clusters that have similar data. Clusters aggregate as needed to create a cluster hierarchy.

Applications describe their data through metadata tags, an ordered list of attributes with a finite set of values. A query is also a list of attributes with a finite set of values.

System Functionality

The design of VIA was guided by the following observations about sharing data across service discovery domains:

- An index should be tailored to the pattern of queries and the content of the underlying data sources. Application specific indices may be more appropriate and have the advantage of being practical to implement.
- Rather than break apart and distribute the index by content, we can distribute the index so that each server only manages "local" information. To reconstruct the fragmented index, servers are connected by links to each other in a peer-to-peer overlay network.

Gateways act as mediators between devices in a discovery domain and resources available in other domains. Whenever a client enters a domain it registers the data it wishes to share with the local gateway. If it leaves the domains its registration is purged after a period of time. The main channel acts as a broadcast mechanism between gateways and can be implemented in many different ways (e.g. an IP-multicast channel or a set of cooperating servers.

A question is propagated from the gateway to the main channel for all other gateways to hear. Gateways that require filtering join a cluster, through linking operations and thus the amount of irrelevant queries they have to handle minimizes, while other gateways remain on the main channel, thus listening to all the queries. The cluster hierarchy represents a logical filtering hierarchy where nodes higher in the hierarchy reduce the workload of nodes lower in the hierarchy by filtering out irrelevant queries.

Gateways have only partial knowledge about other gateways on the network. A gateway monitors the irrelevant question it processes and if these are many it decides to join a cluster. It then generalizes its metadata tags and looks for a cluster it can join. To join a cluster a gateway locates the members of the cluster and links to its closest member. In VIA, the metadata tags encode information about the expected query distribution through the ordering of attributes. From this ordering, a gateway can find the best cluster to join for a particular metadata tag purely through generalizing that tag.

VIA* allows the dynamic adaptation of the gateways organization in order for them to be able to adapt to the changes of the workload in the system. A gateway monitors the query workload and the benefits it has because of the filtering the root of the hierarchy does for it and thus dynamically decides whether to leave this cluster and join another one.

3.2.2.2 Distributed Applications

Many distributed applications are in use today and experience wide recognition. They mainly focus on providing simple services to their users mainly file manipulation and sharing. They mainly focus on scalability, the support of large volumes of data and users.

We present three such applications: Roma, OceanStore and OpenNap.

Roma

Roma [8, 9] is designed to find solutions for the following needs of mobile users: the need for a tool a user can rely on to locate files stored on any of his devices or storage repositories; the difficulty of keeping track of multiple versions of a file across different devices; and the unfortunate dependence on filenames and directories for identifying and grouping files together.

Roma provides an available, centralized repository of metadata, or information about a single user's files. The metadata format includes sufficient information to enable tracking each file across multiple file stores. A user's metadata repository may reside on a device that the user carries along with him, thus ensuring that metadata are available to the user's local devices even when wide area network connectivity is intermittent. To maintain compatibility with existing applications, synchronization agents periodically scan data stores for changes made by legacy applications and propagate them to the metadata repository. Roma metadata include fully extensible attributes that can be used as a platform for supporting other methods of organizing and locating files.

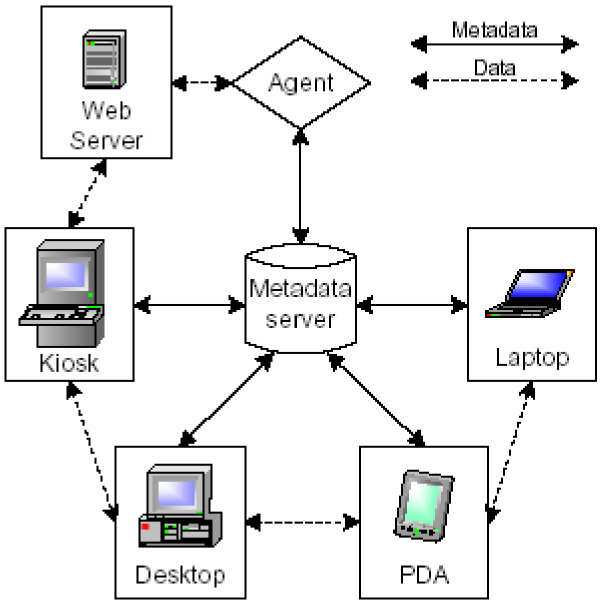

Fig. 3. 4: Architecture of Roma

System Components

At the core of the Roma architecture (illustrated in Figure 3.4) is the *metadata server*, a centralized, potentially portable service that stores information about a user's personal files. It is a logically centralized entity that keeps metadata information about all copies of a user's data.

The files themselves are stored on autonomous *data repositories*, such as traditional file systems, web servers and any other device with storage capability.
Roma's architecture supports:

- data stores that are not under the user's control.
- heterogeneous protocols (local file systems, HTTP, FTP, etc.). There are no restrictions on the protocols supported by a data store.
- data stores with naming and hierarchy schemes independent of both the user's personal namespace and other data stores.

A *metadata record*, that is stored at the metadata server, is a tuple composed of the UID, one or more data locations, a version number and optional, domain-specific attributes. The data location specifies the location of a file instance as a Universal Resource Identifier (URI). The version number is a simple counter. Roma-aware applications can supplement metadata records with a set of optional attributes, stored as name/value pairs, including generic attributes such as the size of a file or its type, and domain specific attributes such as keywords, categories, thumbnails, outlines or song titles.

Finally, Roma *agents* are software programs that run on behalf of the user, without requiring the user's attention and have several tasks that are described below.

System Functionality

Roma-aware applications query the metadata server for file information, and send updates to the server when the information changes. Applications obtain file data directly from data repositories. Agents monitor data stores for changes made by Roma-unaware applications, and update file information in the metadata server when appropriate.

Roma supports *a decentralized replication model* where all repositories store "first class" file replicas that is, all copies of a file can be manipulated by the user and by applications. To increase availability and performance, a user can copy a file to local storage from another device, or an application can do so on the user's behalf. Roma helps applications maintain the connection between these logically related copies, or instances, of the file by assigning a unique file identifier (UID) that is common to all of its instances. The file identifier can be read and modified by applications but is not normally exposed to the user. Once the file is copied, the contents and attributes of each instance can diverge. Thus Roma keeps one metadata record for each file instance. Whenever a change is made to a file instance, its version number is incremented. Furthermore the optional attributes a record can have enable application user interfaces to support new modes of interaction with the user's file space, such as query-based interfaces and browsers.

Although the use of a centralized metadata server needs the insurance that it will be always available despite intermittent network connectivity, the advantages it offers are considered more important.

- *Centralization* helps avoid write conflicts, since a single entity has knowledge of all versions of the data in existence. Some potential conflicts can be prevented before they happen. Also it allows easier searching over all of a user's metadata because applications only have to search at a single entity.

- *Separation* of the metadata from the data store allows easier integration of autonomous data stores, including legacy and third-party data stores over which the user has limited control. Storing metadata on a server under the user's control, rather than on the data stores with the data, eliminates the need for data stores to be Roma-aware. This greatly eases the deployability of Roma. It also makes it feasible to impose a personalized namespace over third-party or shared data. A user can organize his data in a manner independent of the organization of the data on the third-party data store. Finally it enables applications to have some knowledge about data that are currently inaccessible, either because the data store is offline or because it speaks a foreign protocol.

One solution to the problem of availability is to host the metadata server on a portable device kept close to the user. Since metadata tend to be significantly smaller than the data they describe, it is feasible for users to take their metadata server along with them when they disconnect from the network.

Roma-aware applications have two primary responsibilities.

- The first is to take advantage of metadata information already in the repository, either by explicitly presenting relevant metadata to the user or by automatically using metadata to make decisions.
- The second responsibility is to inform the metadata server when changes are made to the data that affect the metadata. At the very least, this means incrementing the version number.

Finally some of the agents' responsibilities are:

- providing background updates of metadata on behalf of the user (for example, updating metadata in response to changes made by non-Roma-aware applications);
- warning a user when he is about to edit an out of date version of a document;
- hoarding files in preparation for disconnected operation;
- making timely backups of information across data stores; and
- tracking third-party updates on autonomous data stores like web servers.

The OceanStore Project

The OceanStore project [10, 11, 12, 13, 14] is an internet-based, distributed storage system that builds a utility infrastructure designed to span the globe and provide continuous access to persistent information.

System Components

OceanStore, in a full-scale deployment would potentially consist of millions of individual servers, each cooperating to provide a service. A group of such servers forms a *pool*.

Objects (every version of every unique entity) are identified by a permanent *globally unique identifier (GUID)*. Certain "OceanStore objects" act as directories, mapping these names to GUIDs and the user maps from names to GUIDs via a hierarchy of such objects. Directories may point to other directories, allowing *arbitrary directory hierarchies*. There exists no centralized root directory; however clients can select several directories as roots.

OceanStore has an *introspection subsystem* consisting of: *Observation modules* which monitor the activity of a running system, keep a historical record of system behaviour and employ sophisticated analyses to extract patterns from these observations. *Optimization modules* which use the resulting analysis to adjust or adapt the computation.

System Functionality

Data in OceanStore flow freely between pools, allowing replicas of data objects to reside at any server, at anytime. Monitoring of usage patterns allows the system to adapt to regional outages and to avoid attacks. Monitoring also enhances performance through pro-active movement of data. Any object may join OceanStore, contributing storage or providing local user access in exchange for compensation. Users subscribe to a single OceanStore service provider but they may consume storage and bandwidth from many different providers. The providers automatically buy and sell storage capacity and network coverage among themselves, transparently to the users.

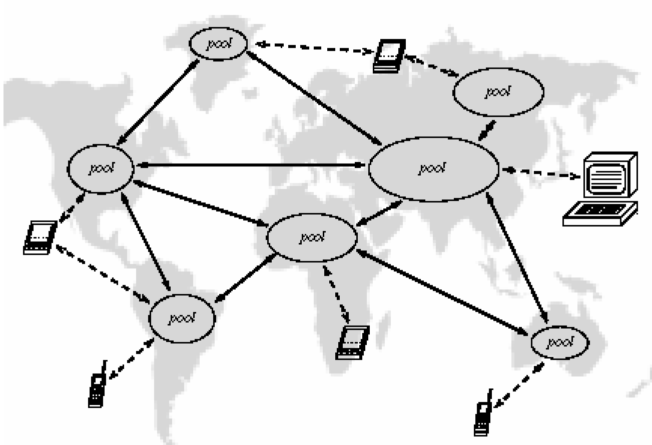

Fig. 3.5: Abstraction of OceanStore's Pools

OceanStore caches data promiscuously; any server may create a local replica of any data object. These local replicas provide faster access and robustness to network partitions. They also reduce network congestion by localizing access traffic. OceanStore uses untrusted servers. Therefore, it is assumed that any server in the infrastructure may crash, leak information, or become compromised.

Objects exist in both *active* (latest version with a handle for update) and *archival* form (permanent read-only version). This highly redundant data encoding is called *deep archival storage*. Oceanstore employs an update model based on conflict resolution in order to implement write sharing (concurrent updates) while avoiding wide-area locking.

In order to locate fragments OceanStore makes use of Tapestry, an overlay network based on the hashed-suffix routing structure by Plaxton. Addressable entities (floating replicas, archival fragments or clients) that are functionally equivalent (i.e. different replicas of the same object) are identified by the same GUID. Clients interact with these entities using a series of protocol messages labelled with a destination GUID, a random number and a small predicate rather than the IP address (*location independent addressing*).

There's a *two-tier mechanism* for routing

i. a fast, *probabilistic algorithm* (using *attenuated bloom filters*) routes entities rapidly if they reside nearby, and if it fails
ii. a slow *reliable global method* (variation of *Plaxton algorithm*) using a large-scale hierarchical data structure locates objects.

OpenNap

The OpenNap system's goal [15, 16] is to enable peer to peer file transfer using certain hosts as directories.

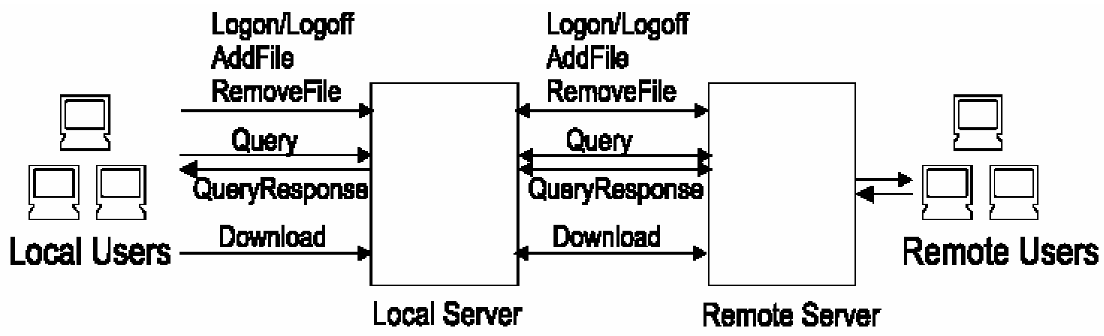

Fig. 3.6: OpenNap Components and Messages

System Components

Two are the main components of the OpenNap system architecture: the clients in which data are stored and the servers which hold indexes for the clients' data.

System Functionality

There are three basic actions supported by the system as indicated by Figure 3.6: login query and download. A description of these actions follows.

On login, a client logins to a server and provides information concerning its library. This information consists of the files a user can share and its metadata. These metadata can be filenames creation dates, copyright info, etc. Each file has its own metadata which are considered to be a document having as context the filename, the author name etc.

The server maintains a table of connections that contains user connection information. Upon login a user can query this server and allow from other users to download his shared files. A user connects to its local server. The other servers are remote servers. Similarly fir a server, users connected, are local users. Users connected to other servers are remote users.

A query consists of a list of desired words. When a server receives a query, a search is carried out in the server's index. If a maximum number of results, which is set by the server, is satisfied, then the query is referred to as satisfied.

There are various ways of processing the queries. However, the main method is the retrieval of the inverted lists for all the words. A list intersection follows that the identifiers of the matching documents are obtained.

A download occurs when a user selects one of the returned results. The user's client is directly connected to the host that has the desired file. Upon download completion, the server is notified of the addition of the new file. This notification also occurs even if the addition of new elements occurs by other means. If a file is removed from a host (if deleted for example) the server is also notified.

When a client logs off, the index of the local server is updates, indicating that particular files are unavailable.

There two policies followed in order to deal with user logins. In the first, called batch policy, only the libraries of connected or active users are in the index. Since the index is updated upon logon and logoff, its size remains as small as possible. This approach while offering query efficiency, it also generates expensive update activity.

In the second policy called incremental, user files are kept in the index at all times. This is an efficiency policy to follow, assuming that only a few files change when a user goes offline, since substantial effort is saved during logon and logoffs. On the other hand query results should be subject to filtering, since the metadata of all users are kept in the index. A constraint rising from this approach is that a user should always connect with the same server, or at least with chained servers, as described below.

The OpenNap system follows chained architecture. Multiple servers are linearly connected forming a chain. A user logs only to its local server, the user's metadata are indexed and a query is submitted. If this query is satisfied, i.e. the desired number of results is returned the rest of the system is not affected. If not, the next server is notified and performs a query. This action continues until the query is satisfied or all servers have been queried.

3.2.2.3 Grid Computing

Grid computing systems mainly focus on using multiple resources scattered around a network of inter-connected users to realize a common task, mostly of scientific nature.

As a representative from this domain, we present Globus.

The Globus Project

The Globus Project [17, 18, 19, 20, 21, 22, 23, 24] is a multi-institutional research effort which aims at developing technologies and infrastructures that will allow the sharing and coordinated use of diverse, globally distributed resources (CPU power, live satellite imagery, mass storage etc.) in dynamic distributed virtual organizations. Globus is a typical example of grid computing.

System Components

The Globus system consists of:

- *Connectivity protocols* concerned with communication and authentication,
- *Resource protocols* concerned with negotiating access to individual resources,
- *Collective protocols and services* concerned with the coordinated use of multiple resources.
- Web services technologies (service description and discovery, automatic generation of client and server code from service descriptions, etc).

It also provides *software tools* (the Globus Toolkit) for building computational grids and grid-based applications. It is a community-based, open architecture, open source set of services and software libraries.

System Functionality

The Grid system focuses on solving the following problems:

- *Security:* The Grid Security Infrastructure (GSI) enables secure authentication and communication over an open network in a decentralized manner using public key cryptography.
- *Identification:* Users and services are identified by a mechanism which is based on unique certificates (containing subject name, a public key, identity of Certificate Authority, digital signature etc).
- *Information Infrastructure* and *Discovery*: The *Globus Metacomputing Directory Service (MDS)* uses the *LDAP protocol* as a uniform means of querying system information (metadata) from a variety of system components, and for constructing a *uniform namespace* for system resources information. The metadata describe current configuration, capabilities and status of the available resources. GRIS (Grid Resource Information Service) provides a means to query resources on a specific grid (runs on the resource's site). The Grid Index Information service (GIIS) maintains an aggregate, hierarchical, LDAP directory of metadata about all the objects of the system, it allows querying and provides indexing and caching to provide a hierarchical overall system image that can be searched by grid applications. Dynamic Registration of resources is done using the Grid Registration Protocol (GRRP), lookup and discovery are done using the Grid Information Protocol (GRIP) that allows hierarchy of resources and directories.

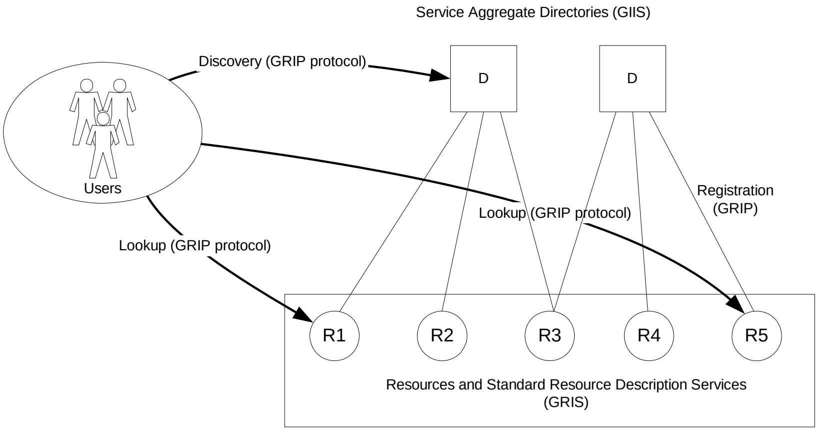

Fig. 3.7: MDS Architecture

- *Resource Management:* Globus uses a layered resource management architecture. High-level global resource management services are layered on top of local resource allocation services. The most important components are: The Resource Specification Language (RSL) that provides a method for exchanging information about resource requirements between all of the components. The Resource Allocation Manager (GRAM) that provides a standard interface to all of the local resource management tools that a site might use. The co-allocator (DUROC) which is a service that coordinates a single request that may initiate multiple GRAMS.

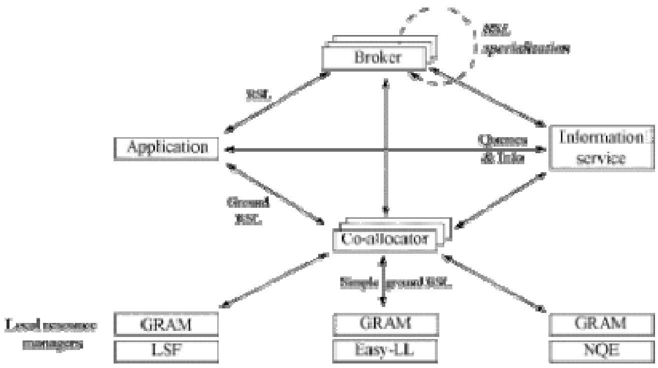

Fig. 3.8: Components of the Globus Resource Management Architecture

- *Data Management:* Globus allows applications to access data stored in any remote file system by specifying a URL. This is done through the Globus Access to Secondary Storage (GASS) module that includes client libraries which allow applications to access remote files, and a server component which allows any node to act as a limited file server.
- *Communication:* The Globus Toolkit contains a communication library (Nexus) designed for the grid environment and supporting multi-method communication. It also contains the Message Passing Interface (MPICH-G).
- *Fault detection:* A monitoring component (Heartbeat Monitor) enables processes to be monitored. Instances (heartbeats) are sent periodically to one or more monitors.
- *Portability:* the Globus Toolkit provides system-level APIs that improve application portability.

3.2.3. Data Centric Systems

Data centric systems mainly focus on data manipulation and try to propose efficient methods for data handling. Apart from issues such as scalability and extensibility, which are also important, the problem of using heterogeneous data sources must be solved. The two sub-categories are data-centric applications which deal with all the issues mentioned above and a more specific category of stream processing systems that focuses on the manipulation of stream data, e.g. data produced from sensors, and the processing of continuous queries. These systems use special techniques to handle and aggregate such data.

3.2.3.1 Data Centric Applications

Data-centric architectures for global computing focus on data manipulation. They advance techniques for describing, storing, indexing and querying data efficiently.

We describe three such systems, Piazza, MOCHA and ObjectGlobe.

Piazza

The Piazza system [25, 26] is built to investigate data placement schemes for peer-to-peer domains with dynamic membership, data and workloads. The data placement problem is how to distribute data and work so the full query workload is answered with lowest cost under the existing resource and bandwidth constraints. Data must be placed in strategic locations and then used to improve query performance. The goal of the system is to achieve scalability even with large number of nodes and moderately frequent updates.

System Components

The peers are the main components of the Piazza system. They play any of the following four roles:

- Data origin: provides original content to the system and is the authoritative source of that data.
- Storage provider: a peer that stores materialized views (consuming disk resources and perhaps replacing pervious materialized views if there is insufficient space).
- Query evaluator: it uses a portion of its CPU resources to evaluate the set of queries forming its workload.
- Query initiator: they act as clients in the system and pose new queries. (A node may initiate new queries on behalf of a query it is attempting to evaluate).

Peers belong to spheres of cooperation, in which they pool their resources and make cooperative decisions. Each sphere of cooperation may in turn be nested within a successively larger sphere, with whom it cooperates to a lesser extent.

The architecture of the system is depicted in Figure 3.9. Data origins 1-2 serve original content; peer nodes, A-E, cooperate to store materialized views and answer queries, but have limited disk and CPU resources; Nodes are connected by bandwidth-constrained links, and advertise their materialized views to their neighbours as shown in Figure 3.9b. Nodes A-C and D-E belong to spheres of cooperation which are both successively nested in a larger sphere.

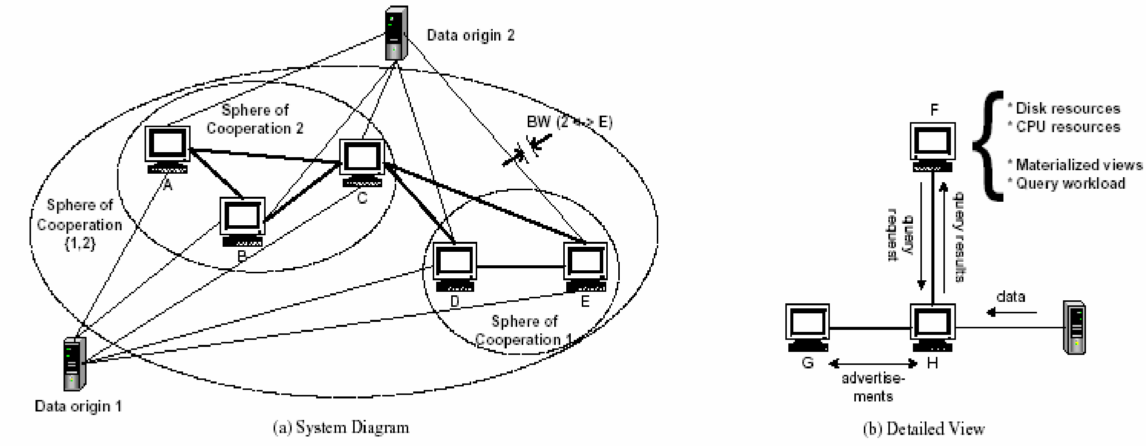

Figure 3.9: The Piazza System

System Functionality

The system models a data origin as an entity distinct from the peers, Piazza can only guarantee availability of data when its origin is a member of the network, and only the origin may update its data. However, a peer could also act as a data origin.

The Piazza system can optimize the system functionality by taking the current query workload, finding commonalities among the queries, exploiting materialized views when cost – effective, distributing work under resource and bandwidth constraints, and determine whether results should be materialized for future use. For reasons of scalability, these decisions are not taken globally, but in the boundaries of a sphere of cooperation. In order this optimization to be performed, two important sub – problems must be addressed:

- *Propagating information about materialized views:* When a query is posed, the ideal approach would be to satisfy this query using the data in nearby storage providers in order to save resources. This requires the query initiator to be aware of existing materialized views and properties such as location and data freshness. One approach to this problem is the use of routing protocols in order to propagate information about materialized views. More particularly, a node advertises its materialized views to its neighbours. Each node gathers these advertisements it receives and propagates them to its own neighbours. If the resources are constrained, a node might reject some of this information, without risking the system correctness – a query can always be posed in terms of data origins. The use of this routing protocol deals with the problems arising by broadcasting every query and view materialization.

- *Consolidating query evaluation and data placement:* If a node poses a query which cannot be satisfied by the data available by known peers, these data must be retrieved by the data origins. However many similar queries that cannot be evaluated might exist, so the system should choose an optimal strategy for evaluating them. For this reason, all un–evaluable queries are broadcast within the cluster. The cluster identifies commonalities among the query set and then assigns roles to specific nodes based on the best expected overall cost.

When original data is updated, materialized views must be refreshed in order to support dynamic data and dynamic workloads. In order to achieve better scalability, expiration times are used on the data items, rather than coherence protocols. This may not achieve the strong semantics of traditional databases, however network traffic is reduced and better guarantees are provided, compared to other P2P systems.

MOCHA

MOCHA (Middleware Based On a Code SHipping Architecture) [27, 28] is a novel database middleware system designed to interconnect hundreds of data sources. MOCHA is built around the notion that the middleware for a large-scale distributed environment should be self-extensible. A self-extensible middleware system is one in which new application-specific functionality needed for query processing is deployed to remote sites in automatic fashion by the middleware system itself. In MOCHA, this is realized by shipping Java code containing new capabilities to the remote sites, where it can be used to manipulate the data of interest.

System Components

Figure 3.10 depicts the organization of the major components in the architecture for MOCHA.

- *Client Application:* MOCHA supports three kinds of client applications: applets, servlets and stand-alone Java applications.
- *Query Processing Coordinator (QPC):* The Query Processing Coordinator (QPC) is the middle-tier component that controls the execution of all the queries and commands received from the client applications. It is responsible for the parsing, optimization and execution of the query and it also handles catalog management and code deployment.
- *Data Access Provider (DAP):* The role of a Data Access Provider (DAP) is to provide the QPC with a uniform access mechanism to a remote data source. Therefore its main functionality is data translation and data filtering.
- *Data Server:* The Data Server is a repository of metadata & data. In particular, it is the server application that stores the data sets for a particular data site. This element can be a full-fledged database server, a Web server or even a file server providing access to flat files.
- *Catalog:* The catalog contains metadata about views defined over the data sources, user-defined data types, user-defined operators, and any other relevant information such as selectivity of various operators. It holds all the necessary information describing the structure and proper use of tables, data types and query operators.

The views, data types and operators are generically referred to as “resources” and are uniquely identified by a Uniform Resource Identifier (URI). The metadata for each resource is specified in a document encoded with the Resource Description Framework (RDF), an XML-based technology used to specify metadata for resources available in networked environments.

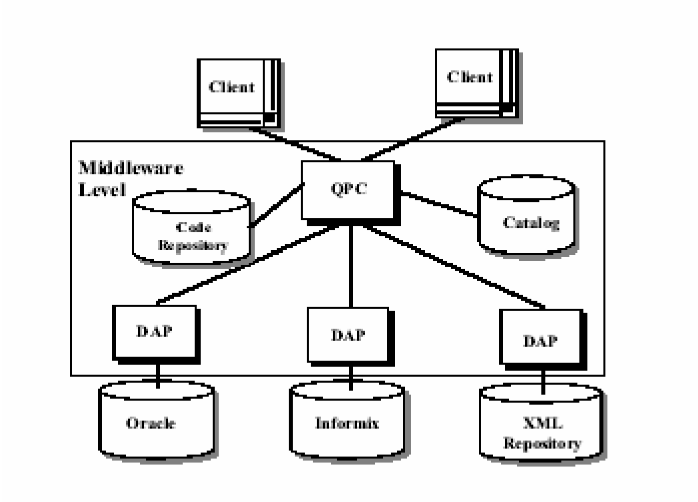

Figure 3.10: MOCHA’s Components

System Functionality

A major goal behind the idea of automatic code deployment, which is the center of MOCHA design, is to fill-in the need for application - specific processing components at remote sites that do not provide them. These components are migrated on demand by MOCHA from site to site and become available for immediate use. This approach sharply contrasts with existing solutions in which administrators need to manually install all the required functionality throughout the system.

The system is built upon Java and XML standards and is platform independent. It provides uniform access to distributed data through an integrated schema for multiple sites. Because of the automatic code deployment its administration is easy and fully extensible with user defined types and methods.

MOCHA capitalizes on its ability to automatically deploy code in order to provide an efficient query processing service. By shipping code for query operators, MOCHA generates efficient plans that place the execution of powerful data-reducing operators (”filters”) on the data sources. On the other hand, data-inflating operators that produce results larger that their arguments are evaluated near the client. Since in many cases, the code being shipped is much smaller than the data sets, automatic code deployment facilitates query optimization based on data movement reduction, which can greatly reduce query execution time.

Query optimization is based on heuristics. The critical factor is usually the network and effort is taken to reduce data movement. Another factor is CPU and I/O costs as well as a quick convergence to a good plan. Many plans are ignored in order to minimize the search space and find a sufficiently good plan fast.

Applets are expected to be the most commonly deployed clients in the system, used as the GUI for the users to pose queries against the data collections and visualize their results. MOCHA provides a set of Java libraries with the APIs necessary for the client to easily interact with the system. These APIs contain all the required infrastructure to load the code containing the application-specific components necessary to manipulate the query results.

Query optimization and automatic code deployment are driven by the metadata in the catalog. Query parsing, query optimization, query operator scheduling, query execution and monitoring of the entire execution process is provided by the QPC. QPC also provides access to the metadata in the system and to the repository containing the Java classes with application-specific functionality needed for query processing. During query execution, the QPC is responsible for deploying all the necessary functionality to the client application and to those remote sites from which data will be extracted. To accomplish that QPC relies on the DAP that acts as a wrapper or a gateway by providing uniform access to remote sites. . However, the DAP has an extensible query execution engine capable of loading and using application-specific code obtained from the network with the help of the QPC. Since a DAP is run at the data source site or in close proximity to it, MOCHA exploits this capability to push down to the DAP the code and computation of certain operators that "filter" the data been queried, and minimize the amount of data sent back to the QPC. This is a feature unique to MOCHA.

ObjectGlobe

The goal of the ObjectGlobe [29, 30] project is to distribute powerful query processing capabilities across the Internet and to perform complex queries on data residing at different sites.

System Components

ObjectGlobe features three kinds of suppliers: *data providers* that supply data, *function providers* that offer query operators to process the data, and *cycle providers* that are contracted to execute query operators. Of course, a single site (even a single machine) can comprise all three services, i.e., act as data-, function- and cycle-provider. Most data and function providers also act as cycle providers.

System Functionality

ObjectGlobe enables applications to execute complex queries, which involve the execution of operators from multiple function providers at different sites (cycle providers) and the retrieval of data and documents from multiple data sources.

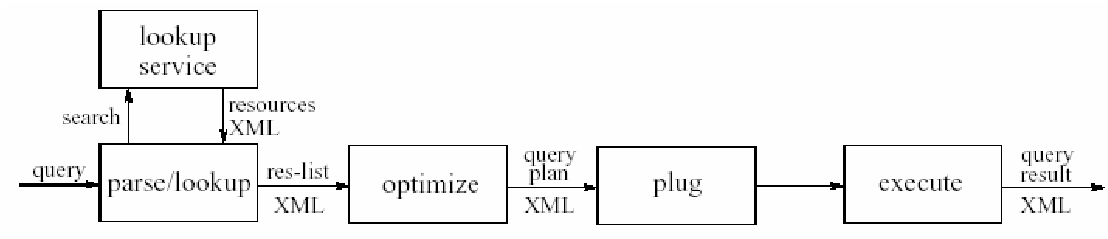

Figure 3.11: Processing a Query in ObjectGlobe

Processing a query in ObjectGlobe involves four major steps:

i. *Lookup:* In this phase, the ObjectGlobe lookup service is queried to find relevant data sources, cycle providers, and query operators that might be useful to execute the query. In addition, the lookup service provides the authorization data—mirrored and integrated from the individual providers—to determine what resources may be accessed by the user who initiates the query and what other restrictions apply for processing the query.
ii. *Optimize:* The information obtained from the lookup service, is used by quality-aware query optimizer to compile a valid (as far as user privileges are concerned) query execution plan, which is believed to fulfill the users' quality constraints. This plan is annotated with site information indicating on which cycle provider each operator is executed and from which function provider the external query operators involved in the plan are loaded.
iii. *Plug:* The generated plan is distributed to the cycle providers and the external query operators are loaded and instantiated at each cycle provider. Furthermore, the communication paths (i.e., sockets) are established.
iv. *Execute:* The plan is executed following an iterator model. In addition to the *external* query operators provided by function providers, ObjectGlobe has *built-in* query operators for selection, projection, join, union, nesting, unnesting, and sending and receiving data. If necessary, communication is encrypted and authenticated. Furthermore, the execution of the plan is monitored in order to detect failures, look for alternatives, and possibly halt the execution of a plan.

External query operators are loaded on demand (from function providers), and they are executed at cycle providers. Just like data and cycle providers, function providers and their external query operators must be registered along with their description in the lookup service before they can be used. ObjectGlobe supports a nested relational data model; this way, relational, object-relational, and XML data sources can easily be integrated. Other data formats (e.g., HTML), however, can be integrated by the use of wrappers that transform the data into the required nested relational format; wrappers are treated by the system as external query operators. XML is used as a data exchange format between the individual Object - Globe components. Part of the ObjectGlobe philosophy is that the individual ObjectGlobe components can be used separately; XML is used so that the output of every component can be easily visualized and modified. For example, users can browse through the lookup service in order to find interesting functions which they might want to use in the query. Furthermore, a user can look at and change the plan generated by the optimizer.

Query execution in ObjectGlobe involves and a monitoring mechanism, which checks for the progress of the execution plan, as generated by the optimizer. It also checks for the liveliness of the query execution, since some queries might time out. Finally, it assures the Quality of the query execution in order to achieve efficient operation.

The Lookup service has an important role in the creation of an execution plan, since it provides the system with the necessary information concerning data, cycle and function providers, as well as with authorization information concerning the execution plan. In order to achieve this, it maintains the appropriate metadata information: For data providers, information is maintained concerning the stored data, which are organized in overlapping themes. Cycle providers are described based on their hardware characteristics, while the information about the available functions is kept for the function providers.

3.2.3.2 Stream Processing Systems

While stream processing systems are also data-centric systems, they focus on dealing with a special kind of data: stream data. Since the use of sensors in many computing systems is very popular and because of the special nature of the data they produce the need for the development of such systems is imperative. Streams data require aggregation and special handling in order to extract useful information from the raw data that sensors produce. Also in stream data, we meet a special kind of queries, continuous queries, which also demand new methods for processing.

MUSE

MUSE [31] is a middleware architecture for sensor smart spaces, which are environments that have some contextual understanding of what is occurring in the environment and perhaps will respond in a reactive way to that.

System Components

A smart space built using the MUSE infrastructure consists of at least a few rudimentary *services*. The term service can take on many different connotations depending on the level of abstraction used to define it. In MUSE, a service is all of the following:

- A software or hardware component that can be "plugged" into a network and dynamically registers itself within a community of other services. Each service provides a set of services to client applications through predefined interfaces. Client applications can also query the community for a list of services they are interested in. The composition of a population of services in a service community is dynamic, and auto-recovery mechanisms exist for dead services to be removed from the community roster after a certain period of time.

- Services are Jini™ services that register themselves with a Jini lookup service and provide proxy objects as an interaction mechanism for a client application to use to interact with the service.

These services belong to a community of services whose population characteristics vary over time. This means that as time goes on some services may disappear and other ones sprout up. A MUSE service community, which represents a *sensor smart space*, consists of the following services:

- A *lookup service* that is used to register what services are available in a community.
- Services called *sensor services* that represent sensors that can be queried for data about the network. This data usually takes the form of some raw format such as measured signal strengths or temperature readings. Because sensor devices are usually quite dumb and will often possess little computing capability, the sensor service is usually a proxy entity that participates in a service community on the behalf of the actual sensor device. The sensor service is viewed as an independent data source that internally acquires its data autonomously. Sensor services will usually monitor the sensor device and update its own internal data records.
- Services called *fusion services* that derive contextual information about what is occurring in the environment. The contextual information usually represents some sort of event such as "Bob opened the door" or "an accident occurred at the corner of Veteran and Sunset". Fusion services are responsible for fusing the data from sensors in an environment in some useful way, and they provide an interpretation of the on goings of a smart space based on the information available from the sensors. In MUSE, fusion services derive the probability of an event occurring (e.g. the person seen is Bob). Probabilistic models are highly attractive because they formally define the likelihood and the level of belief in the conclusions drawn.
- Additional services, such as a *MIRA* service are often used when we want to add information recording to our smart space. The word MIRA is short for "Multimedia Internet Recorder and Archive." Queries on sensor data and derived context data will need to deal with the nondeterministic nature of such information. Work on MIRA is being done to facilitate such a system as well as provide syntax for queries whose results or meanings are nondeterministic in nature. Another design goal of MIRA is that a memory component for a smart space should be pervasive. That is, as you travel from one smart space to another the request to record certain kinds of sensor information should travel with you. This transforms the idea of a centralised database into a distributed one in which recording tasks and queries must be shared between MIRA services. Such services also need to be fairly lightweight so that they can be installed in such a pervasive manner.

System Functionality

Current work in the MUSE project is looking at general purpose inferencing models for fusing different kinds of sensor data into contextual event information. The current approach is to use Bayesian Networks, but another alternative being examined is the use of Hidden Markov Models for certain applications. It is important to realize that derived contextual information is inherently uncertain. There are two main reasons for this: noisiness in the sensor data and quality of the reasoning model.

MUSE services are currently all implemented in Java2 and MUSE's connectivity layer uses Jini™ Connection Technology. Its infrastructure now consists of:

- A skeleton API to build sensor services and allow for consumers to query sensors for information. The API allows for both a polling based model or an event based model to distribute sensor data.

- A skeleton API to build sensing services which use Bayesian Networks as their primary inferencing mechanism. This implementation includes some initial approaches to dealing with the information quality-resource constraint optimization problem.

- A prototype implementation of MIRA.

- An implementation of a sample sensor smart space using the MUSE infrastructure. This sample sensor space is a location system to track a laptop connected to an indoor WaveLAN wireless network

STREAM

STREAM [32, 33] mainly focuses on defining a solid framework for query processing in the presence of continuous data streams. Certain components are defined as show in Figure 3.12: *Input streams*, the *Stream*, a *Continuous Query*, indicated by Q in the Figure, the *Store*, the *Scratch* and the *Throw*. Streams are considered to consist of tuples.

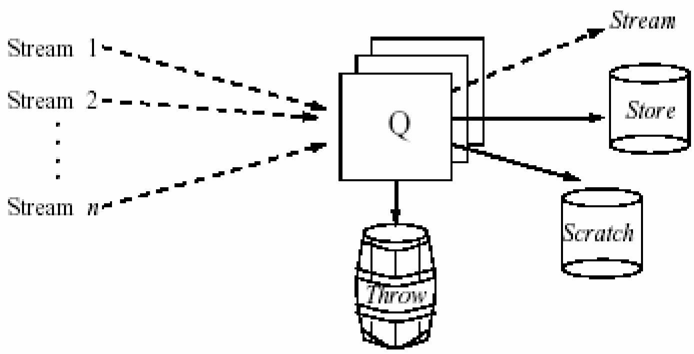

Fig. 3.12: STREAM System Architecture

System Components

Stream is a data stream containing tuples appended to the answer of continuous query Q. Store contains current and previous tuples of Stream Scratch either stores tuples for future use, or derived information from tuples. Tuples which are not the answer for Q are directed to *Throw*. *Throw* also contains tuples that were in *Store* or *Scratch* and are no longer needed.

System Functionality

Let us consider a single continuous query Q with answer A, operating over any number of incoming data streams. Multiple continuous queries can be handled within the architecture implied in Figure 3.12. We also assume that the query is over data streams only, although mixing streams and conventional relations poses no particular problems.

When query Q is notified of a new tuple *t* in a relevant data stream, it can perform a number of actions, which are not mutually exclusive:

i. It can determine that because of *t* there are new tuples in the answer A. If it is known that a new tuple *a* in *A* will remain in A "forever", then *Q* may send tuple *a* to the *Stream* component illustrated in Figure 3.12. In other words, *Stream* is a data stream containing tuples appended to *A*, similar to case (2) discussed previously.
ii. If a new tuple *a* is determined to be in *A*, but may at some time no longer be in *A*, then *a* is added to the *Store* component illustrated in Figure 3.12. In other words, together Stream and Store define the current query answer *A*. If our goal is to minimize storage for the query result, then we want to make sure that tuples are sent to Stream rather than Store whenever possible.
iii. The new stream tuple *t* may cause the update or deletion of answer tuples in *Store*. Answer tuples might also be moved from *Store* to *Stream*.
iv. We may need to save *t*, or save data derived from *t*, so that in the future we are assured of being able to compute our query result. In this case, t (or the data derived from it), is sent to the *Scratch* component of Figure 3.12. Combined with action (iii), data might also be moved from Store to Scratch.
v. We may not need *t* now or later, in which case *t* is sent to the Throw component of Figure 3.12. Note that Throw does not require any actual storage.
vi. As a result of the new stream tuple *t*, we may take data previously saved in Scratch (or Store) and send it to Throw instead. If our goal is to minimize storage, we want to make sure that unneeded data is sent to Throw whenever possible, rather than Scratch.

STREAM does not deal with pervasive applications and environments, so there was no focus on indexing and metadata specification.

Fjords

In the Fjords system [34, 35], the focus is on a specific type of sensor processing environment in which there are a large number of fairly simple sensors over which users want to pose queries.

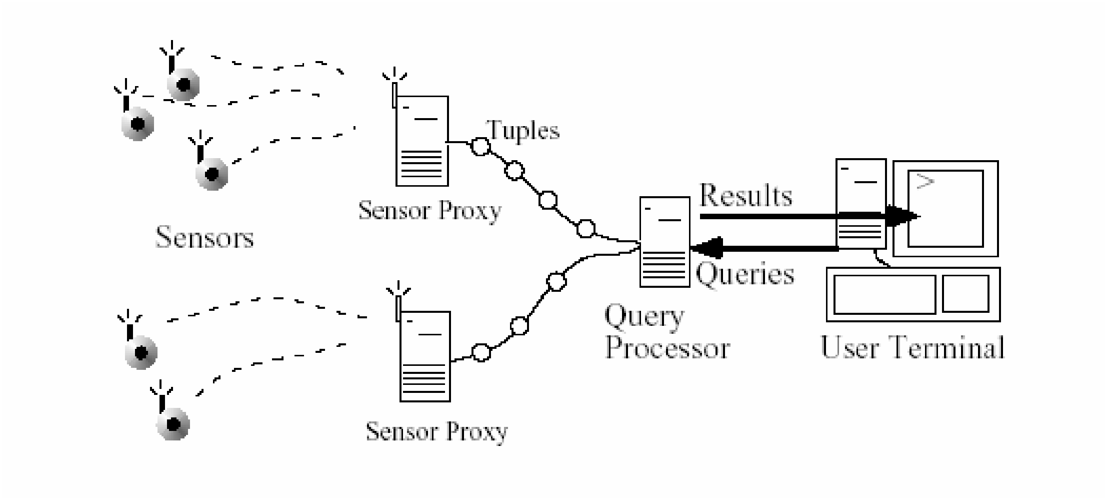

Fig. 3.13: Fjords System Architecture

System Components

A sensor is a remote measurement device that provides data at regular intervals. Sensor data are delivered to each sensor's proxy. Query processors receive user queries and return the results using information stored in form of tuples in the proxies.

System Functionality

This sensor may have some limited processing ability or configurability, or may simply output a raw stream of measurements. Because sensors have at best limited capabilities, they are not expected to parse queries or keep track of which clients need to receive data from them: they simply sample data, aggregate that data into larger packets, and route those packets to a data processing node. This data processing node is a fixed, powered, and well-connected server or workstation with abundant disk and memory resources, such as would be used in any conventional database system. The node that receives sensor data is called the sensor's *proxy*, since it serves as that sensor's interface into the rest of the query processor. Typically, one machine is the proxy for many sensors. The proxy is responsible for packaging samples as tuples and routing those tuples to user queries as needed.

The general query environment is shown in Figure 3.13. Users issue queries to the server; the server processes the query, instantiates operators and locates sensor proxies, and starts the flow of tuples. Although sensors do not directly participate in query processing, their proxy can adjust their sample rate or ask them to perform simple aggregation before relaying data, which, as we will show, is an important aspect of efficiently running queries over many sensors. New Fjords may be merged into already running Fjords with similar structures over the same sensors, or may run independently. Queries run continuously because streams never terminate; queries are removed from the system only when the user explicitly ends the query. Results are pushed from the sensors out toward the user, and are delivered as soon as they become available. Information about available sensors in the world is stored in a catalogue, which is similar to a traditional database catalogue. The data that sensors provide is assumed to be divisible into a typed schema, which users can query much as they would any other relational data source. Sensors submit samples, which are keyed by sample time and logically separated into fields; the proxy converts those fields into native database tuples, which local database operators understand. In this way, sensors appear to be standard object-relational tables.

3.2.4. Conclusions

3.2.4.1 Issues

Among the research issues addressed by global computing systems, metadata representation, mobility support and querying of heterogeneous data are typical examples.

Metadata are data that summarize information about the data available in the system. Metadata representation is a very important issue in the design of a framework that supports diverse, heterogeneous, mobile devices that act as data sources and need to communicate and exchange data and services in a global, ubiquitous, pervasive and dynamic environment. Metadata are records which refer to the resources available across a network. Metadata provide a machine-understandable representation of the description of data, services, terminals, users, context and other information regarding the entities or the environment of a system. They act as a portable information delivery format to share knowledge about the system across time, space, and communities and therefore must be expressed in an international, multi-disciplinary, multi-application and uniform manner in order to achieve maximum interoperability between the various types of devices.

Metadata facilitate the essential functions of service, data, device or user lookup and discovery. Thus, they should be self-descriptive and fully extensible to allow the description of future entities that might enter the system. Both the querying capabilities of a system and the indexing scheme used for discovery depend on the way the metadata are represented and on the modelling policy (what metadata are stored for every system entity). XML and other state-of-the-art XML-based Metadata Formats (RDF, WSDL) are the standards used by most of the systems that deal with many mobile devices and service discovery while grid systems use LDAP structures.

The data models that most systems describe their resources with are based basically on two approaches. The first approach uses a description through attribute – value pairs and the second approach uses a unique identifier to describe any given service or data item. The identifier can be the result of an addressing scheme or a unique resource identifier (URI) in general. There are also systems that use a mixture of the two approaches with a URI and an associated RDF that contains the XML-based description of the service. The ordering of the attributes is used in some cases, to express the expected query workload and is used this information to minimize the irrelevant information each server processes.

Through the advertisement of these metadata descriptions the resources of the systems become available for querying by the users in the system. The kind of queries that a system is capable of answering varies and depends on the data model that systems supports, and the metadata that are kept by the service directories and the indexing scheme that routes the execution of the queries.

Many systems enable queries only with the use of the identifier of an object (service or data), other enable querying about an object with the use of attributes from its description. Other systems support both queries by a description or a unique identifier. There are also systems that provide support for queries that ask for the list of available services of a given server or a data source. Many systems in addition to finding a service, support remote service invocation through the automatic deployment of code or the shipment of the code of a given function to the desired site. Also some systems provide the execution of operators from multiple function providers.

There are many distributed systems that focus on optimization of the queries executed based on commonalities and materialization of views. On file sharing systems, the only query permitted is based on the name of the file. In grid systems (Globus) resources run a local information service that does not support searching but there is also a centralized collective-level index/searching service that acts as a cache and redirects to the resource if the information is stale.

Most of the systems do not have a centralized query executor but this functionality is assigned to modules of the system that query the indexing mechanism to decide on the execution plan of the query.

Locating a resource in an ubiquitous system is in principal just a query but because of its significance all systems attempt to address this as a separate issue. The existing systems deal with this problem using either, or in some cases both of two approaches: in the first approach, which is the most common, dedicated entities are used in order to provide such a service. In the second case there are no dedicated entities for this purpose, but the use of routing protocols is utilized.

In the case of dedicated entities, the architecture might consist of a unique entity, thus providing a centralized, non – fault tolerant, non – scalable architecture, or of a set of entities cooperating for the completion of the common goal. This last alternative was selected for the majority of the systems examined. In the case of numerous interconnected entities, the stored information might be either replicated or distributed among these entities. These entities might be hierarchically organized, or forming some other type of network, for instance, a linear chain. No matter, however, which approach is selected, there are two policies for discovering the desired service: the persistent, where the entity received the request, tries to satisfy it itself, and the non – persistent, where the entity indicates to the one searching for this information, where to locate it. In the second case, there is no dedicated entity for storing indexing information, but this information is retrieved on demand. This is achieved either by simply flooding the network with the respective request, or by using routing protocols, similar to the ones facilitated in computer networks.

The problem described above becomes more intense, when the system supports mobility. The location of resources and users may change through time. There are several types of mobility that a system may support, summarized in Table 3.2.

In order to support any kind of the above types, a system must provide some essential functionality. Firstly, the system must support an identification scheme and an authentication policy which verifies the identification of the entities. Then, it must provide an access control (authorization) policy: a means to restrict access to data or services depending on various parameters, i.e. credit limits, roaming restrictions, time restrictions, personal access rights, etc. Furthermore some policies for login/logout and session management as well as mechanisms for the registration and deregistration are required to provide the means for the dynamic functionality of entities that enter and leave the system at will.

Table 3.2: Types of Mobility

Terminal Mobility	Allows the terminal device (PMO) to change location or even move between networks that use different IP addresses while staying connected and maintaining all services
Component Mobility	Allows hardware components (peripherals) of a terminal to change location while maintaining all services.
Personal Mobility	Allows the user to access all services independently of networks and terminals.
Role Mobility	Allows the user to change roles and get access to distinct preferences and rights.
Session Mobility	Ensures that active sessions are not disrupted while users change terminals
Application/Service Portability	Allows software entities (code, objects and processes) to be relocated from one machine to the other or move between machines while processing.

Most systems presented assume a 1-1 relationship between users and devices and support terminal (or personal) mobility. The two main policies employed by most systems are migration and mapping. In the first case, the information concerning a resource can move from one place to another, together with the resource. In this case, agents can be used to discover the desired resource. One other strategy similar to this one is to migrate the whole execution environment with all if its functionality. Systems using the second approach in order to achieve support for mobile entities use an indexing schema, usually hierarchical. There, the description of the resource or entity is mapped to a unique system identifier, which will be a unique number, handler, address or other representation.

3.2.4.2 Comparison

In this section we compare the systems previously described. We compare these systems on terms of the area each system focuses on, the indexing architecture followed, the metadata representation used, the extensibility and scalability of the system, the support for mobility, and finally the querying schema. Table 3.3 summarizes our conclusions. There are two general domains in which the systems in the table can be divided according to their main area of focus: distributed systems which can be further divided to programming frameworks (One.world, Globe, Jini, VIA), distributed applications (Roma, OceanStore, OpenNap) and grid computing systems (Globus) and data-centric systems which can be further divided to data-centric applications (Piazza, MOCHA, ObjectGlobe) and stream processing systems (MUSE, STREAM Fjords).

Most distributed systems use an *indexing service*, in order to locate either the desired information or the desired service. Globe and VIA use a hierarchical indexing architecture. In Globe it is location based, while in VIA it is application specific. OceanStore uses arbitrary directory hierarchies, having location independent addressing. Also Globus uses a hierarchical index based on LDAP. Jini uses a look-up service for locating information which is based on discovery/join protocols. Finally, maybe the most popular pervasive system currently used, OpenNap uses chained architecture for its indexes. ROMA on the other hand uses an old – fashioned centralized metadata server. One.World does not seem to deal with this issue. Data Centric systems also use indexing techniques in order to locate their data sources. The Piazza system uses routing protocols, an approach which is well known from the computer networks domain. ObjectGlobe uses a replication architecture for its indexes, while MOCHA does not address this issue. From the stream queried processors reviewed, only MUSE uses indexing based on the Jini technologies while STREAM and Fjords do not have a specification for indexing, since their main focus is on dealing with data streams.

Metadata are data about data, i.e. information describing the data of the system. Many of the reviewed systems do not deal with this issue. The systems that have either weak or no reference to this issue are: One.world, OceanStore, OpenNap, MUSE, STREAM and Fjords. One.world and OceanStore do not make any particular reference on the metadata representation used. As for the OpenNap system, there is not a reference on the metadata stored. MUSE, STREAM and Fjords mainly focus on stream data manipulation, so they do not have to cope with this issue. In the distributed systems category, both Globe and ROMA use unique identifiers to describe their data, while VIA uses an ordered list of attributes based on the expected query workload. Jini describes services interfaces and characteristics through metadata and finally Globus uses a resource specification language for the metadata representation. Each of the data centric systems reviewed, uses a different way of representing metadata. In Piazza, the metadata representation depends on the used routing protocol. ObjectGlobe uses component specific metadata, while MOCHA uses views, data types and operators.

One of the most important issues of ubiquitous systems is *extensibility*. Most of the distributed systems examined claim high extensibility. This is achieved in various ways. One.World uses functionality migration from each entity participating in the system to another. Globe deals with each entity assuming it as an object, while Jini supports dynamic services that enter and leave the system at will. VIA does not address this issue. In the applications category, ROMA uses fully extensible metadata, OceanStore uses multiple servers and claims support for any type of device and the OpenNap applications only offer file-sharing services. Finally, Globus is very extensible since it has support for all kind of resources. In the data centric systems domain, Piazza claims support for any peer-to-peer system. MOCHA offers high extensibility by automatically deploying application specific functionality to remote sites, while Globe does not address this issue. Concerning the three stream query processors presented, it is obvious that their main focus is not on this issue.

The majority of the systems reviewed do not deal with supporting mobility for the system components. However, most of the distributed systems reviewed support mobility. These are: One.world, VIA, ROMA, and OceanStore and Globus which imply support. Also MUSE, the stream processing system, supports mobility.

One other important property of ubiquitous systems is their ability to *scale*. Three of the distributed systems reviewed One.world, Globe and OceanStore, use a hierarchical structure, which offers them high scalability. VIA uses distributed indexing, this contributes to its scalability, while ROMA has a limited function on this issue, since it only supports personal file retrieval. Globus and OpenNap are also highly scalable. All the data centric systems examined are highly scalable as well. Piazza supports locality for data discovery, ObjectGlobe has a hierarchical index structure similar to the distributed systems, and MOCHA is self – extensible. The STREAM and MUSE systems do not address this issue, while Fjords system experiments indicated that the system was highly scalable.

Table 3.3 summarizes the comparison.

Table 3.3: Comparison

System	Focus	Indexing	Metadata	Extensibility	Mobility Support	Querying	Scalability
One.World	Pervasive Computing	Central Server	Not addressed	High – supports any type of device by immigrating functionality	Support implied	Queries on tuples	High – hierarchical structure
Globe	Locating Objects	Hierarchical region – based architecture	Use of Unique identifiers	High – supports any type of entity by addressing them as objects.	Not addressed	Search for a unique identifier	High – hierarchical structure
Jini	Resource Discovery	Look-up Service (discovery-join protocols)	Service's interface and properties	High – supports dynamic services	Not addressed	Service Discovery	High
VIA/VIA*	Data sharing across discovery domains	Distributed Indexing-Formation of clusters' hierarchies (application specific)	Ordered list of attributes based on the expected query workload	Not Addressed	Supported	Queries for specific services, type of service, all services of a source	High because of distributed indexing
Roma	Locating current versions of personal files for mobile users	Not Applicable (Centralized Metadata Server)	Unique identifier, version number, data locations and optional domain-specific attributes	High – fully extensible metadata attributes for support of new methods of processing files	Designed for mobile users and different kind of PMOs	Retrieval and updates of personal files	Limited - Used only for personal files
OceanStore	Distributed Storage	Arbitrary Directory Hierarchies	Not addressed	High – any type of device may join the system	Support implied	Search for a unique identifier	High – hierarchal index structure
OpenNap	File sharing	Use of chained architecture	Not addressed	Low – designed only for file retrieval	Not addressed	Based on filename attribute	High – chained architecture followed
Globus	Share Distributed Resources	Grid Index Information service (LDAP-based / hierarchical)	Resource Specificatio n Language, MDS	Supports all types of resources	Implied	Remote Service Requests	High
Piazza	Dynamic Data Placement, Scalability	Use of routing protocols– no specific entity used,	Defined by the routing protocol used	Claimed support for any P2P system	Not addressed	Uses weights. Based on relativity & workload.	High, since designed to support locality support for

		use of materialized views				Support for any kind of data	data discovery
MOCHA	Query optimiza-tion on large scale data environ-ments	Not addressed	Views, data types and operators	High - application-specific functionality needed for queries deployed to remote sites automatically	Not Addressed	Queries on collection of data – Application –specific queries	High because of self-extensibility
Object Globe	Distributed Query Processing	Use of replication architecture	Component specific	Not addressed	Not addressed	Distributed query processing	High – hierarchal index structure
MUSE	Sensor smart spaces	Jini lookup Service	Not addressed	Not addressed	Supported	Queries on sensor data and derived context data	Not addressed
STREAM	Manage-ment of stream data	Not applicable	Not applicable	Not applicable	Not addressed	Use of Continuous queries	Not addressed
Fjords	Manage-ment of stream data	Not applicable	Not addressed	Not addressed	Not addressed	Use of Continuous queries	High – experimental proof

3.2.4.3 The Role of DBGlobe

Although, there are many research efforts on building systems for global computing, there has been no much work on semantic discovery and advanced querying. Most often, discovery is based on simple file lookup, while querying, even when supported, lacks a unified query language, even in the data-centric systems category.

DBGlobe innovation lies on:

- Service-orientation: data are wrapped into services
- Semantic service discovery: a service and a parameter ontology are used to select the appropriate service
- Community concept: data are clustered into communities based on context
- Mobility support
- Active XML language: a language that allows service calls inside XML documents

3.2.5. References for Section 3.2

[1] Robert Grimm, Janet Davis, Eric Lemar, Adam MacBeth, Steven Swanson, Steven Gribble, Tom Anderson, Brian Bershad, Gaetano Borriello, David Wetherall, Programming for Pervasive Computing Environments. Technical report UW-CSE-01-

06-01, University of Washington, Department of Computer Science and Engineering, June 2001.

[2] One.World project: http://one.cs.washington.edu/

[3] Maarten van Steen, Philip Homburg, and Andrew S. Tanenbaum: "Globe, a wide area distributed system"

[4] B.C. Popescu, C. Gamage, A.S. Tanenbaum:"Access Control, Reverse Access Control and Replication Control in a Worldwide Distributed System"

[5] http://wwws.sun.com/software/jini/

[6] Paul Castro, Benjamin Greenstein, Parviz Kermani, Richard Muntz, Maria Papadopouli, Chatschik Bisdikian. Locating Application Data Across Service Discovery Domains. Proceeding of the International Conference on Mobile Networking and Computing, July, 2001.

[7] UCLA Multimedia Systems Laboratory: http://mmsl.cs.ucla.edu/publications.shtml

[8] Edward Swierk, Emre Kiciman, Nathan C. Williams, Takashi Fukushima, Hideki Yoshiday, Vince Laviano and Mary Baker. The Roma Personal Metadata Service. Stanford University, Computer Science Department, October 2001

[9] Roma Personal Metadata Service Homepage: http://mosquitonet.stanford.edu/~hideki/roma/

[10] John Kubiatowicz, David Bindel, Yan Chen, Steven Czerwinski, Patrick Eaton, Dennis Geels, Ramakrishna Gummadi, Sean Rhea, Hakim Weatherspoon, Westley Weimer, Chris Wells and Ben Zhao. *OceanStore: An Architecture for Global-Scale Persistent Storage*, Nov.00, University of California, Berkeley. In Proc of the 9th International Conf. On Architectural Support for Programming Languages and Operating Systems (ASPLOS '00).

[11] Chris Wells. *The OceanStore Archive: Goals, Structures, and Self-Repair*. Computer Science Division, University of California, Berkeley. (Thesis)

[12] Sean Rhea, Chris Wells, Patrick Eaton, Dennis Geels, Ben Zhao, Hakim Weatherspoon and John Kubiatowicz. *Distributed Data Storage: Maintenance-Free Global Data Storage*, IEEE Internet Computing, September-October 2001.

[13] Zhao B.Y., Kubiatowicz J.D., and Joseph, A.D. *Tapestry: An Infrastructure for fault-tolerant wide-area location and routing.* Tech.Rep UCB/CSD-01-1141, University of California, Berkeley. Computer Science Division, April 2001.

[14] Steven E. Czerwinski, Ben Y. Zhao, Todd Hodes, Anthony D. Joseph, Randy Katz, An Architecture for a Secure Service Discovery Service. Fifth Annual International Conference on Mobile Computing and Networks (MobiCOM '99), Seattle, WA, August 1999 ACM Baltzer Wireless Networks: Selected papers from MobiCOM 1999.

[15] Beverly Yang, Hector Garcia-Molina: Comparing Hybrid Peer-to-Peer Systems. VLDB 2001: 561-570

[16] Stanford Peers Homepage: http://dbpubs.stanford.edu:8090/pub/2000-35

[17] Ian Foster, Carl Kesselman, Globus: A metacomputing infrastructure toolkit. International Journal of Supercomputer Applications, 11(2): 115-128, 1997.

[18] Ian Foster, Carl Kesselman, The Globus Project: A Status Report. In the Proceedings of the Heterogeneous Computing Workshop, pages 4-18, IEEE Press, 1998.

[19] Ian Foster Carl Kesselman, Jeffrey M. Nick, Steven Tuecke, The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration, Working Draft, 03/02/2002.

[20] Ian Foster, Carl Kesselman, Steven Tuecke, The Anatomy of the Grid: Enabling Scalable Virtual Organizations. To appear in Intl. J Supercomputer Applications, 2001

[21] Ian Foster, Carl Kesselman (eds.), The Grid: Blueprint for a New Computing Infrastructure, Morgan Kaufmann, 1999.

[22] Sudharshan Vazhkudai, Steven Tuecke, Ian Foster, Replica selection in The Globus Data Grid.

[23] Ann Chervenak, Ian Foster, Carl Kesselman, Charles Shalisbury, Steven Tuecke, The Data Grid: Towards an Architecture for the Distributed Management and Analysis of Large Scientific Datasets.

[24] William Allcock, Ian Foster, Steven Tuecke, Ann Chervenak, Carl Kesselman, Protocols and Services for Distributed Data-Intensive Science

[25] Steven Gribble, Alon Halevy, Zachary Ives, Maya Rodrig, Dan Suciu: What can Databases Do for Peer to Peer? WebDB Workshop on Databases and the Web, June 2001.

[26] data.cs.washington.edu/p2p/piazza/

[27] Manuel Rodriguez, Nick Roussopoulos. MOCHA: A Self-Extensible Database Middleware System for Distributed Data Sources. In the Proceedings of the ACM SIGMOD International Conference on Management of Data, Dallas, Texas, USA, May 2000

[28] www.cs.umd.edu/projects/mocha/

[29] Reinhard Braumandl, Markus Keidl, Alfons Kemper, Donald Kossmann, Alexander Kreutz, Stefan Seltzsam, Konrad Stocker: ObjectGlobe: Ubiquitous query processing on the Internet. VLDB Journal 10(1): 48-71 (2001)

[30] ObjectGlobe Homepage: http://www.db.fmi.uni-passau.de/projects/OG/

[31] MUSE: http://mmsl.cs.ucla.edu/muse/

[32] Shivnath Babu, Jennifer Widom: Continuous Queries over Data Streams. SIGMOD Record 30(3): 109-120 (2001)

[33] STREAM project Homepage: http://www-db.stanford.edu/stream/

[34] Sam Madden, Michael J. Franklin: Fjording the Stream: An Architecture for Queries Over Streaming Sensor Data, ICDE 2002.

[35] Michael J. Franklin Homepage: www.cs.berkeley.edu/~franklin/Papers/

[36] Manoj Parameswaran, Anjana Susarla, Andrew B. Whinston, P2P Networking: An Information-Sharing Alternative, IEEE Computer July 2001 (Vol. 34, No. 7)

[37] Matei Ripeanu, Peer-to-Peer Architecture Case Study: Gnutella Network, University of Chicago.

[38] Matei Ripeanu, Ian Foster, Adriana Iamnichi, Mapping the Gnutella Network, Properties of Large-Scale P2P Systems and Implications for System Design.

[39] Klaus Haller, Heiko Schuldt, Using Predicates for Specifying Targets of Migration and Messages in a Peer-to-Peer Mobile Agent Environment, Swiss Federal Institute of Technology (ETH) Institute of Information Systems, Nov. 2001.

[40] Richard G. Guy, Peter L. Reiher, David Ratner, Michial Gunter, Wilkie Ma, Gerald J. Popek: Rumor: Mobile Data Access Through Optimistic Peer-to-Peer Replication. ER Workshops 1998: 254-265

[41] Beverly Yang, Hector Garcia-Molina, Comparing Hybrid Peer-2-Peer Systems. In Proc. of Very Large Databases (VLDB), 2001

[42] Beverly Yang and Hector Garcia-Molina. Efficient Search in Peer-to-peer Networks. In ICDCS, 2002.

[43] Ian Foster, Carl Kesselman, Steven Tuecke, “The Anatomy of the Grid Enabling Scalable Virtual Organizations”, Intl J. Supercomputer Applications, 2001.

[44] Ian Foster, Carl Kesselman, Globus: A metacomputing infrastructure toolkit. International Journal of Supercomputer Applications, 11(2): 115-128, 1997.

[45] Ian Foster, Carl Kesselman, The Globus Project: A Status Report. In the Proceedings of the Heterogeneous Computing Workshop, pages 4-18, IEEE Press, 1998.

[46] Ian Foster Carl Kesselman, Jeffrey M. Nick, Steven Tuecke, The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration, Working Draft, 03/02/2002.

[47] Ian Foster, Carl Kesselman (eds.), The Grid: Blueprint for a New Computing Infrastructure.

[48] Mark Weiser, Ubiquitous Computing, IEEE Computer "Hot Topics", October 1993.

[49] M. Satyanarayanan, A Catalyst for Mobile and Ubiquitous Computing, IEEE Pervasive Computing, Vol. 1, No. 1 Jan.—Mar. 2002

[50] M. Satyanarayanan, Pervasive Computing: Visions and Challenges, IEEE Personal Communications, August 2001

[51] David B. Stewart, Miniature Software for Large Pervasive Computing Applications, White Paper, 2001

[52] http://www.eBiquity.org

[53] Ian Foster, Carl Kesselman, "Globus: A Metacomputing Infrastructure Toolkit", Proceedings of the Workshop on Environments and Tools for Parallel Scientific Computing, SIAM, Lyon, France, August 1996.

[54] Greg Lindahl, Andrew Grimshaw, Adam Ferrari, Katherine Holcomb, "Metacomputing--What's in it for me?", Legion a Worldwide Virtual Computer, Technical report

[55] Semantic Web Activity Statement.

[56] The Semantic Web, TIM BERNERS-LEE, JAMES HENDLER and ORA LASSILA, Scientific American, May 2001.

[57] Semantic Web History: Nodes and Arcs 1989-1999. The WWW Proposal and RDF, Dan Brickley.

[58] M. Satyanarayanan, "Pervasive Computing: Vision and Challenges", IEEE Personal Communications, August 2001

[59] M. Weiser, J.S. Brown "The Coming Age of Calm Technology", P J. Denning, R. M. Metcalfe, Eds, Beyond Calculation: The Next Fifty Years of Computing, Copernicus, 1998.

[60] Bernard Cole, "The Name Game Continues: Invisible Computing" http://www.embedded.com

[61] Javed I. Khan, Yihua He, Ubiquitous Internet Application Services on Sharable Infrastructure, Technical Report, 15 March 2002, Ken State University.

[62] Dieter Schmalstieg, Gerd Hesina, Distributed Applications for Collaborative Augmented Reality, Vienna University of Technology

[63] William Thies, Michal Karczmarek, Saman Amarasinghe, StreamIt: A Language for Streaming Applications, Laboratory of Computer Science, MIT, March 2002.

[64] Laks V.S. Lakshmanan, Sailaja Parthasarathy, On Efficient Matching of Streaming XML Documents and Queries, Univ. of British Columbia, Vancouver, 2002.

3.3 Comparing DBGlobe with Other Projects within the GC Initiative

The DBGlobe project aims at developing novel data management techniques to deal with the challenge of *global computing*. On the premise, global computing is a database problem: how to design, build and analyze systems that manage large amount of data. *However, the traditional database approach of storing data of interest in monolithic database management systems becomes obsolete in such environments.*

DBGlobe is part of the "Foundations of networks and large distributed environments" cluster. DBGlobe is the only project in the cluster cantered around data; as such each approach is complementary to other projects in the cluster. Next, we describe the other projects in the cluster along with ideas for potential co-operation.

The other projects in the cluster are:

CRESCCO: Critical Resource Sharing for Cooperation in Complex Systems

CRESCCO investigates at a foundational level the critical aspects and bottlenecks in highly dynamic and extremely complex global systems emerging from the integration of heterogeneous *communication infrastructures*, highly dynamic mobile user populations and selfish agent entities. The focus is on the efficient management of limited and scarce communication and computing resources to ensure co-operation and efficient access to advanced services in such environments, by investigating the design and implementation of high-speed, cost-effective and reliable communication and computing solutions.

A potential co-operation between CRESCCO and DBGlobe is through the integration of communication solutions advanced by CRESCCO into the DBGlobe architecture. Another possible interaction could be to consider data and services as critical resources and to apply approaches developed by CRESCCO.

FLAGS: Foundational Aspects of Global Computing

FLAGS plans to provide a unifying scientific framework and a coherent set of design rules, for global systems resulting from the integration of autonomous interacting entities, dynamic multi-agent environments and ad-hoc mobile networks. The focus is on the issues of (i) co-operation and antagonism, (ii) stability and fault-tolerance and (iii) communication and motion in such global systems.

There is potential for cooperation between the FLAGS and DBGlobe on the topic of data delivery.

SOCS: A Computational Logic Model for the Description, Analysis and Verification of Global and Open Societies of Heterogeneous Computees",

SOCS investigates computational and logical models of individual and collective behaviour of computational entities, called computees, interacting in the context of global and open computing environments. SOCS aims at supplying models of complex interaction that not only allow formal specification and verification of properties but also lend themselves to concrete realisations which can be proven correct with respect to those models, without relying upon simulation techniques. Such properties may be local - within a single computing environment, or global - within a dynamic collection of open and connected sub-environments.

SOCS and DBGlobe take quite different approaches to global computing. In an abstract level, communities of computees and ad-hoc databases may (in some sense) be modeling abstractions with similar goals. We see this a potential start for co-operation.

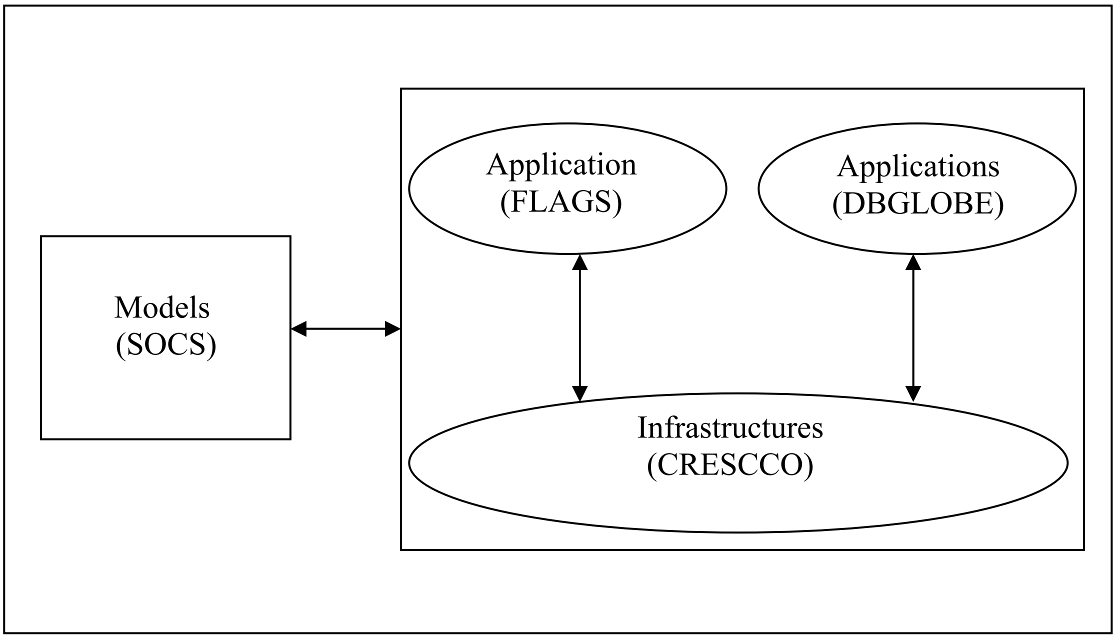

Fig 3.14: Dependencies among GC cluster projects

Figure 3.14 shows the interaction between the four projects as envisioned by P. G. Spirakis in his presentation of the cluster at the GC workshop in Malaga, July 12, 2002.

FLAGS and DBGlobe appear at the application level: DBGlobe considers data and services, while FLAGS considers communication and computation. CRESCCO deals with infrastructure issues on which both applications build. Finally, SOCS is concerned with formalizing the properties and the behavior of global computing applications and infrastructures.

4. Project Results and Achievements: Assessment

4.1 Comparison to Original Project Objectives

The DBGlobe project has successfully met the milestones set in Annex I.

Milestone 1: Overall system design. The design includes:

1. The overall architecture of the DBGlobe approach
2. A detailed specification of DBGlobe's functional components
3. Appropriate distribution and replication protocols.

Our design satisfies the goals set in deliverable 18.1:

- is extensible,
- is dynamically configurable,
- is fault tolerant, and
- provides high availability.

Milestone 2: The second milestone includes producing (a) the first research results in the areas of coordination/data delivery and in query processing, and (b) the first working version of simulator: our simulator will model mobile entities and their interaction.
In respect to part (a) of Milestone 2, we have reported our initial approach to data co-ordination in Deliverable D8 and on correctness issues in publications A3, A4 and A5 above. With respect to part (b), the first and second version of the prototype has been produced and is presented in Deliverables D6.

Milestone 3: The third milestone includes (a) Research results in the coordination/data delivery and query management. These were expected to be reported in research publications. (b) Extension of our simulator to model the creation of ad-hoc databases, data delivery, coordination and querying.

In respect to part (a), our research results in the coordination/data delivery and query management has been produced and is presented in Deliverable D9, D11, D12 and D13. With respect to part (b), the extension of our simulator has been produced and is presented in Deliverable D7.

Milestone 4: The fourth milestone includes prototype application: development of a prototype ad-hoc database system to support location aware queries.

In respect to Milestone 4, we have reported our development of a prototype ad-hoc database system to support location aware queries in Deliverables D14, D15 and the proof of concept application is presented in Deliverable D19.

In Section 4.1.1 we provide an *overall* assessment of the main results and we describe the overall interaction, cooperation and coordination of all partners towards achieving these key results. In Sections 4.1.2 - 4.1.7 we present an assessment of the results achieved *per workpackage*.

4.1.1 Overall Assessment and Partner Interaction

DBGlobe views the conglomeration of interconnected nodes carrying data as a virtual superdatabase with a dynamic schema. Our overall goal is to manage this superdatabase. In particular, our objective is to develop data management techniques for modeling, indexing and querying data hosted by open-ended networks of massively distributed nodes.

To resolve heterogeneity and semantic mismatch problems, we employ a service-oriented approach. In DBGlobe, the data of each peer are made publicly available through services, that is, access to data stored locally at each node is achieved by invoking an appropriate service. Direct querying of the structure and content of data is also supported by defining services that employ an XML-based query language. At a second level, metadata, i.e., data describing nodes and their services, are maintained to capture their behavior and state. A uniform XML-based representation for services and metadata is used to facilitate information exchange and sharing. Thus, our focus is on semantic matching of hierarchical service descriptions and data. The collection of data on devices that exist around a specific context (e.g., location or user) forms a data sharing community that we call an ad-hoc database or simply community. The configuration of such communities varies over time. Communities are designed to support context-specific behaviors such as location-aware queries.

In a nutshell, the main results of the project as a whole are:

- *A new routing scheme for path-queries based on Bloom filters:*
  We have designed and implement a distributed routing scheme based on small-sized XML indexes that is used to route path queries in large and dynamic distributed systems towards the nodes with matching XML documents. The indexes are novel data structures, called multi-level Bloom filters.

- *A novel language, AXML:*
  AXML incorporates service calls inside XML documents. It is now open source in the ObjectWeb industrial consortium. The European project Edos on open source distribution considers using ActiveXML for software distribution of Mandrake Software and the E.dot French RNTL project with Institut National de Recherche en Agronomy uses AXML to develop a warehouse on food risk. We are also considering the use of ActiveXML for network configuration. This is a joint work with the DistribCom team of INRIA Bretagne (ASAX French ARC). If successful, ActiveXML could be used by France Telecom and Alcatel (Swan French RNRT project).

- *A time and space based ontology for location based services:*

The Time Ontology we have defined includes the *movement class* (which is not available in other ontological environments) and is tailored to the needs of location-based services. Although the findings are merely theoretical, we collaborate with a Greek startup company Talent SA on using this new ontology. The specific product, Cruiser – a Web portal specialized in the exchange and publishing of spatially-referenced information, is based on the concept of exploring information by navigating based on its spatial aspect.

- *A new location-based service discovery scheme:*
  Within the field of global computing initiative, mobile service discovery plays an eminent role. The contribution lies in improving the precision and recall of mobile service discovery, by introducing the use of implicit contextual information related both to the user and the web service. This is achieved by a novel proposal for a context-aware service directory that provides searching facilities for web services based on contextual parameters. This approach is fully implemented in our prototype.

The above results have been published in prestigious international conferences (SIGMOD, VLDB, ICDCS, ICDE, EDBT, etc).

As an ***objective measure*** of take-up, we have added counters at the web page of the project. Currently, the web page has about 5 accesses per day, each deliverable has 1.5 accesses per day, which projects to about 1800 and 540 respectively accesses per year.

Google returns 646 results of which around 175 are non similar results containing the DBGlobe acronym. We estimated that this increases in a rate of 5 results (2 non similar results) per week.

Finally, DBGlobe publications have an average of 2 citations per paper so far. We expect that this figure will increase significantly the next years as more papers (that are currently under review) citing DBGlobe papers will be published.

Interaction among Partners

The interaction among partners has resulted in a relatively unified approach based on the principles of

- peer-to-peer computing,
- sharing of XML documents, and
- web-service computing.

The work of several project partners was focused on the goal of finding a compelling way to structure the underlying information domain.

UoI and CTI worked initially together on defining the metadata scenario in a global computing environment. During several overall project meetings and meetings organized individually by the project partners a variety of approaches and ideas relating to metadata in DBGlobe were discussed. As a result of these discussions, several approaches were actively pursued in this project, (i) data structures for metadata, (ii) context and (ii) communities.

Although finally working on different directions of this problem, the works of AUEB, UCY and CTI are complementary.

The common idea of localizing metadata by means of communities was adopted in the overall system architecture defined by all the project partners, in several publications by AUEB, UCY and CTI as well as the final prototype by UCY.

The idea of exploiting context for information discovery in a global computing setting was addressed in several publications of AUEB, CTI and jointly by UCY and UoI.

Web services developed at AUEB are regularly used from INRIA in an ActiveXML environment.

One possible limitation of the cooperation is that AXML is not fully integrated with our prototype of the context-aware service discovery protocol. This is mainly because both AXML and service discovery were being developed in parallel and thus the full-fledged implementation of AXML was not available at the time of the implementation of the prototype of the service discovery protocol.

The benefits of the integrated approach are evident in the AXML design and implementation. They have led to extending it towards supporting mobile handheld devices with limited resources (memory, bandwidth and battery power).

There are relationship between ***grid research*** and *global computing*. Initially, grid research focused essentially on a range of middleware technologies intended to support resource sharing among groups of computers. The common goals of the two approaches as well as their different objectives are also highlighted at the official web page of the Global Computing (GC) initiative [1].

The ***applications*** envisioned for DBGlobe include distributed systems with the following characteristics:

- Large scale (both in the number of sites involved and the geographical distribution of the computing devices)
- Mobility of users, devices and/or data
- Heterogeneity of the resources

Such applications include for example large scale *peer-to-peer information sharing* systems, where the information shared goes beyond simple files and the search queries are more advanced than simple keyword-based ones. The information shared may include XML documents and web services and the queries may be based on location and other context parameters. Another compelling application for DBGlobe is *car-to-car* interaction. In this scenario, cars in highways communicate to each other information regarding the state of the road (e.g., bumps, accidents) or other useful information (prices of gasoline at nearby gas stations).

In retrospective, we believe that the project would have benefited if:

- We could support longer visits among the different sites of the project partners; this would enhance the achieved level of integration of our research results.

- We have designed the project as a three-year effort.
  Initially, the project was a two-year effort; the requested extension for a third year was mainly for keeping in touch with the other projects within the global computing initiative. With hindsight, we believe that having instead scheduled the project as a three-year one would have allowed us to allocate more resources and effort at the third year and that would have enhanced the project.

[1] Global computing initiative web site: http://www.cordis.lu/ist/fet/gc.htm

4.1.2 Assessment of Results in System Architecture (WP1)

Introduction

The main target in the first work package (WP1) of DBGlobe was to derive appropriate architectures for ad-hoc databases of mobile entities. Such architectures should be metadata driven for facilitating transparent interoperability among the involved entities, and enhancing location management, which is a critical aspect of a global computing system that relies mostly on mobile distributed entities.

The process of assessment and evaluation of the system architecture in DBGlobe is based on the comparison and evaluation of the results of this workpackage against the aims and the goals specified by the corresponding milestones and expected results. In the rest of this section, we present these assessment and evaluation results. First, we provide an overview.

Overview of Results

In a nutshell, the technical results that were attained are:

- A service-oriented architecture in which data are encapsulated within services to support interoperability and extensibility

- An analysis of metadata and models for their representation and storage

- General research results on content caching

- A location-based distribution of DBGlobe servers that supports extensibility

Meeting the Milestones, Design Objectives and Success Criteria

In order to be able to assess the progress in this workpackage, we considered to what extent the objectives comprising the milestones of WP1 have been achieved.

The design of the system includes:

- the overall architecture of the DBGlobe system ,
- a detailed specification of DBGlobe's functional components
- the design of the metadata management system
- appropriate distribution, caching and replication protocols

In particular, the following objectives with respect to the system architecture of the DBGlobe system have been met:

- We defined what is the appropriate metadata information to describe mobile entities; such metadata include information in various levels of detail and granularity,

- We derived an appropriate and universal metadata definition and manipulation language for the effective and efficient definition, management and maintenance of the scores of the available metadata,

- We designed distribution and replication protocols for the DataStores that are mainly storage devices and hold metadata among other things, and the main computational entities of the system (called DataHandlers in the initial architecture and CAS in the revised one).

- We designed our architecture to be dynamically configurable and extensible, in order to accommodate for possible expansions in the scale of the participating entities.

- Our distributed design achieves fault-tolerance and availability.

We used the simulation environment WP2 to verify that:

- every component of the system was examined with respect to the provision of the appropriate services or functionalities,

- the appropriateness of the interfaces among the various components.

Evaluation Methodology

As outlined in Deliverable 18.1, our self-assessment process includes a set of steps that provides a structured scenario-based architectural analysis. In particular, we concentrated on the following steps:

- *Describe the candidate architecture:* For any analysis, we must first have an architectural representation of the product with a well-specified semantic interpretation that describes what it means to be a component or a connector. The description should be in a syntactic architectural notation that all parties involved in the analysis can understand. We made three meeting of all partners and several work group meeting where all ambiguities and disagreements concerning the notation have been discussed and clarified.

- *Develop, evaluate, and weight scenarios and scenarios and scenario interactions:* Several task and user scenarios have been considered and evaluated in relation to the system architecture. For this purpose, we agreed on adopting a quite general application framework of a community of people (with devices/data) who move and who share dynamically changing data and services. Snapshots of this framework that were used to produce the task and user scenarios were: traveling communities, conferencing, and the community web environment.

Meeting Evaluation Criteria

The status of this workpackage was evaluated with focus on the following major issues.

Successful capture of system requirements. The DBGlobe architecture captures a number of properties. As these were discussed and defined by the partners of the project, DBGlobe architecture should:

- be distributed,
- provide for publication, discovery, and querying of data,
- support integration, mobility, and change of control in a seamless manner,
- perform efficiently metadata management
- support context-awareness
- treat data, metadata, and services uniformly
- provides for streaming and multicasting

Accommodating for mobility and location awareness. Mobility and location awareness is a key feature of the design in all DBGlobe components.

*Identification of specific properties for mobile entities***.** We have defined the properties of mobile entities (called PMOs in our architecture).

*Definition of system architecture***.** This was presented in deliverable D3.

*Successful specification of functional components***.** These definitions were presented in D3.

*Correspondence between modules and system requirements***.** The system was design so as to comply with all requirements determined in Task 1.1

Metadata for content, mobile devices, user preferences, etc. This was accomplished in Task 1.2 and the results were reported in Deliverables D2 and D3.

Determination of replicating and caching: Methods for replication and caching

Iterative design. The design of the simulator and the prototype application gave feedback to system architecture design.

Contingency Plan

A valid contingency plan, which ensures that the workpackage will be able to recover from unexpected events and will deliver the proposed results, is always required in a project like DBGlobe.

Recommendation made by the experts during the first review concerning: (i) the support of small devices, (ii) integration with AXML and (iii) scale were taken into consideration.

In particular, our new design (and our prototype) supports a variety of small devices and communication protocols. AXML services are also integrated and can be invoked and executed in the prototype. Finally, special emphasis has been put on issues such as caching, replication and semantic clustering as means to address scale. Note that our decision to design and implement both a simulator and a prototype aim at handling scale: large-scale experiments are made using the simulator and the output and conclusions drawn by such experiments are used to drive the design of the prototype.

4.1.3 Assessment of the Simulation Environment (WP2)

Introduction

The main target in the second work package (WP2) of DBGlobe was to design and implement a virtual test bed of the proposed DBGlobe architecture (WP1), which simulates a real-world dynamic environment of mobile entities able to co-operate with each other. Input from the simulator will be used for the development of the proof-of-concept prototype of DBGlobe (WP5).

A detailed description of the initial design of the simulator was provided in deliverable D6: “Initial simulator prototype and web page” and of its final version in deliverable D7: “Final Simulator Prototype and Simulator Web Page”.

Evaluation of the Simulator

The simulation system was been thoroughly tested according to widely approved testing techniques. Testing was not simply the process of performing tests to the completed system (i.e. the location-aware tourist information exchange application). The most common testing practice was consisted of 4 test-tasks that should been applied recursively for each subsystem if necessary to uncover errors related to the design, development and integration process.

(1) Unit Code-Test: was focused testing on each separate unit (or component) as implemented in source code, to verify that it functions properly. Units (PMO server module, PMO client module, registration, authentication, login, session management, handover, location service) were tested making heavy use of white-box techniques.

(2) Integration (or Design) Test: was applied after the process of assembling the units and integrating them to produce the complete system. The focus of testing was shifted to the design and the construction of the software architecture and the data-flows between the subsystems. Heavy use of black-box test-case design techniques performs post-integration tests on the units, to locate integration errors or communication problems.

(3) Requirements Validation Test: At this stage the requirements established during the early stages are validated against the characteristics of the system that has been constructed and assembled to finally ensure that the software meets all functional, behavioral and performance requirements, using exclusively black-box techniques.

(4) System Engineering Test: Software, hardware, framework and other system elements are tested as a whole combined with additional external entities that affect the use of the simulator (e.g. users, devices, external services). At this final stage the testing process will have to focus on correct extraction of metadata about the devices and ensure that the system tracks and interprets correctly the location and movement of PMO devices.

The 4-step testing process was reapplied after the completion of the proof-of-concept prototype and the simulation environment and was reassessed and modified accordingly.

Evaluation of research results

A first evaluation cycle was performed by an internal committee, the second by a consortium committee that was appointed in the third meeting.

Specific experiments we conducted to address issues at the networking level such as:

- To identify the maximum number of clients that could be served in terms of acceptable response time in a DBGlobe Cell
- To simulate mobility scenarios and identify possible variations of the maximum number of clients that could be served in terms of acceptable response time in a DBGlobe Cell

In the second year, special emphasis was also put on context management and on creating communities (or adhoc databases). The simulator was extended and various experiments were performed towards exploiting contextual information to improve service discovery for mobile devices. In particular, to enable searching for services based on:

- The context of the requesting device (properties, characteristics, capabilities)
- The context of the user (user profile)
- The context of the web service (location, time, version, type of returned results)

A demo of the contextual aspects of the DBGlobe simulator is available at:

http://faethon.db-net.aueb.gr/ServiceDiscoveryDemo/DemoForm.aspx

Interesting research results have been produced in the following areas:

- modeling the distribution, mobility and the data contents of mobile entities and deriving upper limits base on the networking-layer capabilities
- mobility issues in cell based systems.
- expressing the interaction among the entities.
- forming dynamic communities of mobile objects that act as ad-hoc databases.
- running computations and integrating data on large collections of geographically dispersed computing resources.
- capturing contextual information regarding users, devices and services
- specifying a formal language that allows service description and discovery.

Information about the simulator can be found at the web page:

http://www.db-net.aueb.gr/dbglobe/Simulator1.htm

A detailed description of the prototype and research results and issues raised by it are presented in deliverables D6 and D7, as well as in related publications in the IEEEWIC Web Intelligence (WI) 2003 conference, the TES 2003 VLDB workshop, HDMS04, MDM05 and the LOBSTER 2002 workshop.

4.1.4 Assessment of Results in Data Delivery and Coordination (WP3)

Introduction

The goal of WP3 was twofold:

- To derive adaptive data delivery mechanisms that will combine push (transmission of data without an explicit request) and pull, periodic and aperiodic, as well as multicast and unicast delivery.

- To use workflow models and co-ordination techniques from multi-agent research to capture the co-ordination among the mobile entities and advanced transaction models to reason about the correctness of the interaction among the entities.

Meeting the first goal

One important aspect in DBGlobe was the way data communicated among the components that participate in the system: (a) the mobile entities, (b) the DBGlobe servers and (c) the users. Current database servers support a pull-based mechanism of data delivery: data are sent to clients after having been explicitly requested. This request-response time of data delivery was not appropriate for all interactions in the DBGlobe system. Some of the mobile entities deliver information in a push-based fashion, that is, without an explicit request. Both push and pull data delivery may be periodic or aperiodic. For instance, for push delivery, data may be sent, when a condition is met (e.g., the refrigerator notifies the user's PDA, that there is no milk) or periodically (a sensor reports measurements of the ozone level every hour). Finally, data may be received either by a single component or by a group of components. DBGlobe's data delivery mechanism must incorporate:

- Push and pull data
- Periodic and aperiodic
- Multicast and unicast.

In global computing, data are delivered from sources to clients in a variety of modes. In Deliverable D8 we have studied the properties of the various data delivery mechanisms and how they can be deployed in the overall DBGlobe framework.

We classified the various data delivery mechanisms based on the following four basic characteristics: (1) how is data delivery initiated (push vs. pull), (2) the frequency of the delivery (periodic vs. aperiodic), (3) the receivers of the data delivery (unicast vs. broadcast) and (4) the unit of interaction (data vs. query shipping). Various combinations along these four dimensions are possible. An interesting such combination is data diffusion in which an interest (query) for a data item is periodically injected (pulled) in the network and then broadcasted to the appropriate nodes for evaluation.

A central issue in data delivery is ensuring that the clients receive consistent data. We developed a formal framework for expressing the correctness properties of data delivery and present protocols that ensure both the temporal and the semantic coherency of data for the various data delivery mechanisms. By temporal coherency, we refer to the currency related properties of the data sets read by a client, e.g., when were the data items read by a client current at the server? Semantic coherency refers to transaction-based consistency properties, e.g., do the data items read by a client correspond to a "serializable" execution of the server transactions? Without both semantic and temporal coherency, clients may be seeing either outdated data sets or data sets that correspond to inconsistent database states.

We showed how such mechanisms fit into the DBGlobe architecture as defined in DBGlobe's "Deliverable D3: Overall System Architecture" and present directions regarding an adaptive deployment of each mechanism.

In the second year of the project, we focused on experimental evaluation of the above mechanisms and on deriving an adaptive mechanism. We have proposed a suite of broadcast organization schemes for multiversion data broadcast, i.e., data broadcast in which more than one value is broadcast per data item. Besides increasing the concurrency of client transactions, multiversion broadcast provides clients with the possibility of accessing multiple server states. For example, such functionality is essential to support applications that require access to data sequences and have limited local memory to store the previous versions, such as in the case of sensor networks. We identify two basic multiversion organizations, namely vertical and horizontal broadcasts and propose an efficient compression scheme applicable to both. We also consider a multiversion client data cache and introduce appropriate cache replacement techniques. Finally, we propose an adaptive scheme that dynamically selects the appropriate broadcast organization based on the client access pattern. We provide performance evaluation results for both flat and broadcast disk organizations and for a variety of client query patterns.

Related results are reported in Deliverable D10: "Data Delivering and Querying: Results". They are also presented at ICDT03 conference, the *Information Systems* Journal and the MDM04 conference.

In addition, we contributed a new paradigm for publish/subscribe systems centered on the novel notion of subscription summarization. This work was accepted and presented at the ICDCS 2004 conference.

Finally, we have designed and implemented a module supporting continuous and asynchronous Web services in Active AXML. The embedding of this module in the systems allows for flexible use of pull and push modes for data acquisition and integration in the AXML layer of DBGlobe.

Meeting the second goal

A workflow is an activity involving the coordinated execution of multiple tasks by different processing entities. We have investigated the use of workflows in both the specification of computation in DBGlobe as well as for providing execution guarantees. In particular, workflow concepts are used for detailing the tasks to be carried out and the execution requirements. Furthermore, transaction workflows are deployed for providing safeguards of traditional databases systems related to computation correctness, data consistency and durability. The computation model of DBGlobe is service-oriented. Thus, our approach is also related to service coordination and specification.

In particular, computation in the DBGlobe system evolves continuously and non-deterministically. We have studied how workflow concepts can be used to model and enforce the coordination of activities in our system. At the current state of DBGlobe, there is no control flow or any direct dependencies among the services. There is a variety of workflow or process definition languages. Taking into consideration such languages, we have considered how our computational model can be extended by adding to it control flow, exception handling and other workflow-related constructs.

Furthermore, transaction workflows have been deployed for providing safeguards of traditional databases systems related to computation correctness, data consistency and durability. We study how such concepts can be integrated within DBGlobe and propose some initial protocols.

Related results are reported in Deliverable D9: “Modelling Coordination through Workflows”.

Meeting the Objectives

The evaluation of this work was accomplished following the criteria set in Deliverable 18.1:

(a) Consensus of all partners regarding the mechanism: related issues were discussed and agreed upon at the various project meetings

(b)-(c) Presentation in competitive forums and specialized workshops, and evaluation of the feedback from the research community and publication in competitive conference proceedings and international journals.

Work in this workpackage was presented/will be presented at the ICDCS 2004 conference, the Information Systems Journal, MDM 2004, DOA04 and ICDT 2003.

(d) Performance results using the simulation environment.

Contingency Plan

Recommendation made during the second review concerning the use of SOAP, workflows and AXML were taken into account and new research was spawn towards integrating AXML within workflow languages.

4.1.5 Assessment of Results in Information Discovery and Querying (WP4)

Introduction

The basic objective of WP4 is to derive methods and techniques for querying and discovering knowledge in environments consisting of autonomous, dynamic, and mobile entities. The first fundamental challenge is choosing an appropriate data model to describe the information available to, and offered by, these dynamic entities, and designing a query language and a query execution paradigm for this context. Other challenges include query evaluation and information discovery.

Specific research results related to this workpackage are described below. These results have been published in a number of highly competitive database conferences including SIGMOD 2003, SIGMOD 2004, and EDBT 2004.

Research Results

As often when dealing with heterogeneous and autonomous information sources, our goal is to hide the heterogeneity and the data distribution details from the naive users, and enable them to manipulate and query the data as if it resides in a single source. The solution developed here is based on the following two main observations. XML, and emerging standards for web services, like SOAP and WSDL, normalize the way that distributed remote resources can be accessed, and more generally the way programs can be invoked over the web. De-centralized architectures, in which resources are shared by direct exchange between systems, propose a scalable solution for dynamic distributed environment with independent data sources. These two paradigms, combined together, form the basis for our scalable data integration platform. We propose Active XML (AXML in short), a language that leverages services for data integration and is put to work in a global computing architecture. The AXML framework is centered on AXML documents. These are XML documents that may contain calls to services. When calls included in an AXML document are fired, the document is enriched by the corresponding results. An AXML PMO or DBGlobe system component contains AXML documents and offers (AXML) services, which can be used to enrich the AXML documents of the same or of other components. In some sense, an AXML document can be seen as a (partially) materialized view integrating plain XML data and dynamic data obtained from service calls. As a simple example, consider an AXML document for the home-page of a local newspaper. It may contain some plain XML data, such as general information about the newspaper, and some dynamic fragments, e.g. one for the current temperature in the city, obtained from a weather forecast Web service, and a listing of current art exhibits, obtained from the local TimeOut guide service. While documents with embedded calls have been used before, to our knowledge, AXML is the first to actually turn calls to web services embedded in XML documents into a powerful tool for data integration.

Controlling Service Call Activation. The service calls inside AXML documents are represented by special XML elements carrying the tag <sc> (for service call) which are interpreted as calls to services. An <sc> element encodes the URL of the site providing the service, the service name, the particular functionality of the service that is invoked, and the appropriate parameters for the call. Particular attributes of the element allow to specify when a service call should be activated (e.g. when needed, every hour, etc.), and for how long its result should be considered valid. This simple mechanism allows capturing and combining different styles of data integration such as warehousing and mediation.

Continuous services. Simple services are similar to remote procedure calls. The service is called with some arguments, and eventually returns an answer. For the interesting class of continuous services, the interaction is more complex. Once a service call has been registered, a stream of answers is returned for this single service call. Streams of answers are encountered in many real-life applications: the stream of source updates for the maintenance of a data warehouse, the readings of a temperature sensor or surveillance system, answers returned by continuous queries or publish-subscribe systems. In the case of such continuous services, the activation of the service call encapsulates a service subscription, and all the results received subsequently from the service are integrated in the caller document.

Intensional Parameters and Results. The parameters and result of a service call may themselves be AXML documents. A system component receiving a call with parameter containing service calls may have to activate the calls it includes before actually performing the service. Similarly, a service result may contain further service calls, which have to be activated by the site receiving the result. Thus, service call activations entail exchanging intensional data, and lead to a form of distributed computation. The use of intensional parameters/results involves security issues. For instance, a malicious user may force a site to perform a dangerous action by calling one of the site's services with an intensional parameter that contains a call to a dangerous service. Besides security, the peer capabilities need also to be taken into consideration. For instance, if the receiving site is a non-AXML client (hence unable to activate the service calls embedded in the data) only fully extensional data must be sent. To address these issues we introduced a typing mechanism to control the execution of services. Just like for regular XML data, AXML schemas (ala XML Schema) are used to control the exchange of AXML data and, in particular, to determine which services should be called before sending an AXML document, and which should not. We formalized the problem and developed algorithms to solve it.

Defining Services. The AXML framework allows to use arbitrary existing services as well as to define new services on top of the enriched AXML documents. The definition of AXML services relies on parameterized XML queries expressed in XQuery, the standard language for querying XML data, extended with updates. AXML services may query and update AXML or regular XML documents. The AXML service specification allows, in particular, the definition of continuous services, and of services with intensional input/output. One should note that an

AXML service, when called by a non-AXML client, will detect the limitations of its caller and in such case will return only extensional answers.

Distribution and Replication. By the sole presence of PMO- and DBGlobe- services, AXML documents already include inherently some form of distributed computation. A higher level of distribution that also allows (fragments of) AXML documents and services to be distributed and/or replicated over several sites is highly desirable in a dynamic mobile architecture: A single XML document may need to be distributed on more than one PMO if the PMO does not have enough storage; Replication of services, as well as the data they utilize, is needed, since a PMO will tend to use services "nearby". To address this, we extended the AXML data model with distribution and replication, and provided a location-aware extension to XQuery to handle distributed/replicated data and services. We developed a comprehensive cost model for query evaluation that applies to both user queries and service class. Finally, we designed an algorithm that (based on the above cost model) chooses data and services that DBGlobe should replicate to improve the efficiency of maintaining and querying its dynamic documents.

Filtering and Knowledge Acquisition. To support flexible exchange and filtering of information, we introduced a novel paradigm that allows the exchange of intensional Active XML documents between applications, and studied the new possibilities and the great flexibility they bring to application design.

Query Evaluation and Optimization. We studied query evaluation on Active XML documents. A major challenge in the efficient evaluation of queries over such documents is to detect which calls may bring data that is relevant for query execution and to avoid the materialization of irrelevant information, The problem is intricate, as service calls may be embedded anywhere in the document and service invocations possibly return data containing calls to new services. Hence, the detection of relevant calls becomes a continuous process. Also, a good analysis must take the service signatures into consideration. We formalized the problem and provided algorithms to solve it. We also presented an implementation that is compliant with XML and web services standards and is used as part of the DBGlobe system. Finally, we experimentally measure the performance gains obtained by a careful filtering of the service calls to be triggered.

Service Discovery. We have developed a new data structure, termed multi-level Bloom filters, for the representation of the structure of XML documents. The multi-level Bloom filters are used to route path queries such extending the traditional keyword based search. In addition, we have proposed a content-based grouping of nodes so that nodes with similar content are clustered together.

Testing and Evaluation Results

To validate the proposed ideas we intend to build a prototype implementation of the system and use it as an applications development platform. As mentioned before, this includes a lightweight version capable of running on standard PDA's, which will be useful for testing aspects related to peer mobility. The implementation will be done by students and the feedback will be used to refine and tune the proposed query paradigm.

On the industrial side, it is quite clear that information management for heterogeneous mobile entities is in great demand. The technology that will possibly emerge from this research has thus the potential for a fast industrial implementation, given the increasing need for such support on the one hand and the fact we base our research on industrial standards such as XML, SOAP, and WSDL. AXML is used in a project with the French National Institute of Agronomy for the construction of a warehouse on food risk. Discussions for possible transfer of AXML are also conducted with ACATEL for network configuration and with Mandrake Software for software distribution.

4.1.6 Assessment of the Proof–of-Concept Prototype (WP5)

The aim of WP5 is to develop a proof of content prototype that verifies the various

concepts developed within the DBGlobe project. In summary,

- To demonstrate through a concrete example the idea of ad-hoc databases and the various forms of querying
- To provide a concrete example of the workflow interaction model and the different data delivery mechanisms
- To capture, design and implement a location-aware system

The functional specification of the architecture and the definition of ad-hoc databases are presented in detail in Deliverable "D14: Location Aware Ad-hoc Databases and Query Processing".

To this end, we created "SpotMe", an application that utilizes the idea of communities as this was presented in D14. The general operation and functionality of "SpotMe" is to enable a user to find out where his friends/colleagues/partners are, what services are offered to him (either by his friends/partners or others that registered with the DBGlobe system) and allow him to advertise/post his own services. In detail, "SpotMe" provide its users with the following functionality:

- Location Management (for example, find the nearest friend/service)

- Notion of community (find services of a given type)

SpotMe is a context aware application built around web services. Context aware application servers are connected with the DBGlobe system by calling the appropriate web-services (e.g. CAS web-service, or Community web-service). In this way, any SpotMe client that enters a DBGlobe covered area can use the SpotMe application.

The overall architecture of the prototype and the application as well as a user manual are provided in Deliverable "D15: Prototype System for Location-Aware Query Processing".

Evaluation

The evaluation criteria of the outcomes of the this workpackage as set in Deliverable D18.1 "Parameters and Criteria for Self-Assessment" are related publications in conferences or journals covering the following topics: (i) Context aware applications build on top of the infrastructure produced by WP2, (ii) Location based innovative services and architectures, (iii) Description of the DBGlobe application and its innovative features and (iv) Demonstration type of papers.

There are various publications as well as demonstrations during the second year of the project covering the above topics as described in detail in the dissemination reports D16.2 and D16.3.

Work in this workpackage included the extension and refinement of our prototype. Detailed description of this research is reported in deliverable D19. In summary, besides various enhancements in the service discovery mechanism, we have also: (i) evaluated our prototype in a distributed setting and (ii) incorporated in it small sensor devices.

In DBGlobe, we have proposed creating a dynamic overlay network above the core system of web services to group together semantically related services, thus creating a network of communities. Each of these communities is a set of references to semantically related services that are distributed over the global mobile environment (for example, a community of weather services, or a community of services related to music). Communities are distributed; they are organized in a global taxonomy whose nodes are related semantically. This taxonomy can be seen as an expandable, flexible and distributed semantic index over the system services, which aims at improving the cost of service discovery.

In addition, we support the notion of context. Context is used to constraint the number of the semantic related services to those that are appropriate for a given context. We model context as a set of attribute value pairs. Having the communities managing concept-related services provides for a more efficient service discovery. The idea of communities and the associated query processing mechanisms were presented in Deliverable D14.

To demonstrate the viability of our approach, we have implemented the infrastructure for supporting communities as well as a prototype application that utilizes this infrastructure. Our prototype implementation called "SpotMe" is an application that utilizes the idea of communities. The general operation and functionality of "SpotMe" is to enable a user to find out where his friends/colleagues/partners are, what services are offered to him (either by his friends/partners or others that registered with the DBGlobe system) and allow him to advertise/post his own services. SpotMe is presented in Deliverable D15.

In this last year of the project, we enhanced our distributed service discovery mechanism that utilizes communities for context-based service discovery. Our performance results indicated that community-based service discovery works well under various workloads. In particular, in wireless environments and for a service request rate of 200 requests/second, the framework provides performance improvements even over an approach in which each unit can satisfy every service request locally.

We also incorporate in our prototype small devices. In particular, we have developed a framework for incorporating in DBGlobe web services offered by small devices controlled by SMS messages. Such devices include observation cameras and various types of sensors. SMS messages are traditionally used as means for controlling GSM standalone devices. Information collection from such devices is performed either via SMS messages (e.g. temperature, atmospheric pressure, humidity) or via MMS messages (e.g. observation images or video).

A publication describing the current version of the prototype will be published in the proceedings of MDM05. A report on the work on supporting GSM-enabled mobile sensors has been submitted for publication.

The prototype can be accessed at:

http://cs458.cs.ucy.ac.cy:8080/dbglobe/jsp/login.jsp

and the sensor proxy can be accessed at:

http://sensor-proxy.cs.uoi.gr/index_ds.htm and
http://sensor-proxy.cs.uoi.gr/index_ds.wml

4.1.7 Assessment of the Project Management and Dissemination of Results (WP6)

This work package is responsible for the co-ordination of the project in both administrative and technical terms aiming towards achieving effective operation of the project, as well as timely delivery of the results.

We have achieved:

- all project objectives within the work program time and resource constraints,
- deliverables with "publishable" research quality, as suggested by the publications resulted from the project
- the compliance with the reporting requirements outlined in the Contractual Agreement entered into the European Community,
- a number of meetings and the proper functioning of the working groups

- the wide distribution of the research results of the project by publishing at competitive journals and conferences and updating the web page of the project,
- the attendance of maximum visibility for the project in the research (and industry) community by presenting our work in many scientific and industrial forums,
- to disseminate the simulator and the proof of concept prototype (we have set up pages for distributing the simulator and the proof of concept application through the web, also we have demonstrated both the simulator and an early version of the prototype application in competitive forums),
- the continuous observation and assessment of areas of impact which provided our project with invaluable feedback.
- describe the activities related to the integration of DBGlobe project with the other project of the Global Computing project (both with projects inside the "Foundation of Network sand Large Distributed Systems" cluster and with related projects at other clusters).

The achievements of disseminating the results are described in detail in deliverables: D16.1 - D16.4. The results of the clustering activities went beyond the initial target. Many fruitful collaborations and complementary cooperative efforts have been launched in this context. These are described in detail in deliverable D20.

4.2 Relations and Synergies with Other Relevant Projects

Research wise we have compared our work with other research in the literature in the research papers that we have produced. Our results are novel or tackle problems that have not been discussed by other researchers. To this end we have performed an extensive survey of system architectures that may be capable of supporting global computing environments. The results of this work are presented in Section 3. Our conclusion is that although there are many research efforts on building systems for global computing, there has been no much work on semantic discovery and advanced querying. Most often, discovery is based on simple file lookup, while querying, even when supported, lacks a unified query language, even in the data-centric systems category. Section 3 provides also a comparison with other projects of the GC Initiative.

DBGlobe innovation lies on:

- Service-orientation: data are wrapped into services
- Semantic service discovery: a service and a parameter ontology are used to select the appropriate service
- Community concept: data are clustered into communities based on context
- Mobility support

- Active XML language: a language that allows service calls inside XML documents

4.3 Contribution to Programme/Key Action Objectives

The vision of global computing encompasses nearly all areas in computer science. DBGlobe addresses the related data management issues, thereby representing the *database community's contribution* to this emerging area.

In DBGlobe we developed *data management* techniques for constructing dynamic systems for the co-operation of autonomous dynamic entities. The traditional database approach of storing data of interest in monolithic database management systems becomes obsolete in such environments. A completely different paradigm is needed to address the requirements of global computing.

Our focus is on the *overall* design, description and performance of such dynamic database systems. In alignment with the Global Computing Initiative, our dominant concerns were on:

- feasibility rather than on efficiency,
- functionality rather than on performance,
- principles and foundations rather than on implementation techniques.

DBGlobe has a clear focus on the design and analysis of systems of mobile and autonomous entities and addresses all four characteristics of such systems namely:

(i) autonomy and lack of central control (DBGlobe is a federation of autonomous database entities)

(ii) mobility (both data and processing entities are mobile, furthermore, location management is a central issue in DBGlobe)

(iii) dynamic configuration (federation of databases are dynamically created as entities join and leave the system)

(iv) incomplete information.

DBGlobe addressed foundational aspects of data management and revisited most areas of database research including architectures, communication/co-ordination and querying. Thus, it has the potential to lead to innovations in theory and design and create new topics in database research as well as in the more general area of distributed computing.

4.4 DBGlobe's Innovations and Technical Contributions

In the near future, there will be increasingly powerful computers in smart cards, telephones, and other information appliances. There will be substantial computing engines in the cellular phones that we carry. Smart buildings will put computers in light switches, vending machines, and home appliances. Software objects, in the form of mobile software agents, will roam the Internet. This will create a massive infrastructure composed of highly diverse interconnected mobile entities. Novel computing paradigms are needed to model and manage this dynamic environment.

The vision of global computing encompasses nearly all fields in computer science and will require fundamental research contributions from all these communities. DBGlobe *made a first attempt to address the related data management issues, thereby representing the database community's contribution to this emerging area.*

In the DBGlobe project, we take a data-*centric* approach.

- At a first level, we view each mobile entity as an autonomous database system.
- At a second level, metadata – data about the entities – are used to capture and manage their behavior and state.

In this context, data exchange and computation occurs in the background in response to cues or queries from users. Computing is performed on a multitude of such databases (networked mobile processing entities and their data), rather than on a centralized device or system.

The collections of databases on mobile entities that exist around a specific context (e.g., location or user) form a data sharing community that we call an *ad-hoc database*. DBGlobe *is designed to facilitate the seamless creation and operation of such ad-hoc databases.* Ad-hoc databases are designed to support behaviors such as the following:

- Location-aware: activities are suggested based on locations' cost and proximity, in addition to user preferences
- Home: home appliances "collaborate" to ensure that all ingredients are in hand for the family's favorite desert; lights, audio and video systems work together to set different moods.

The configuration of such ad-hoc databases varies over time. The entities constituting the ad-hoc database are mobile and autonomous. Such ad-hoc databases are databases of unprecedented complexity in terms of scale, heterogeneity, autonomy, and co-ordination. They form a radically new form of database management systems. Fundamentally novel techniques are needed to capture them.

The need to radically broaden database research focus is well recognized among the database community [Asilomar Report]. DBGlobe extends the scope of database research to new endeavors.

Some of the novel ideas in our approach are:

Metadata-driven data management: a framework for determining and implementing data management policies based upon the analysis of metadata such as information about the user data requirements specified in profiles or the location of mobile entities.

Lack of centralized control: Execution and co-ordination is spread among numerous processing entities.

Dynamic Configuration: mobile entities enter and leave an ad-hoc database dynamically.

We describe next DBGlobe project's innovations in specific research areas, namely: databases systems architectures, co-ordination/data delivery and querying. Then, we highlight the novel research issues addressed by our simulator and proof-of-concept prototype.

4.4.1 System Architectures for Ad-Hoc Databases

Global computing renders current system designs inappropriate. Our goal was to develop scalable architectural paradigms for developing self-configuring systems under various constraints and preferences. Our architecture has gone beyond traditional approaches to support the following requirements.

- Each component of the system (mobile entity or InfoStation) as well the system as a whole is self-tuning: must be able to adapt as conditions change.

- The architecture is extensible: It is possible to add a new mobile entity to the DBGlobe system and have the system automatically discover and interact with the other databases systems. This information discovery process requires that system components provide substantially more metadata that describe the meaning of objects they manage.

- To increase availability and failure recovery, there is a need to dynamically replicate information.

To achieve such ad-hoc communities of database systems, there is need for maintaining metadata about the mobile entities. The architecture of the DBGlobe is metadata and web-service driven.

Metadata information about the mobile entities includes the type of data in each mobile entity, the location of the mobile entities as well any potential constraints or conditions. Multiple Ontologies have been defined to represent the semantics of data. Metadata information is saved at multiple levels of details. We have determined a metadata model and a metadata definition and manipulation language.

Finally, there is a need for a uniform representation of the mobile entities, their functionality and data. To this end, AXML have been adopted as the common platform for information exchange.

4.4.2 Data Delivery and Co-ordination

One important aspect in DBGlobe is how data are communicated among the components that participate in the system: (a) the mobile entities, (b) the InfoStations and (c) the users. Current database servers support a pull-based mechanism of data delivery: data are sent to clients after having been explicitly requested. This request-response time of data delivery is not appropriate for all interactions in the DBGlobe system. Some of the mobile entities deliver information in a push-based fashion, that is, without an explicit request. Both push and pull data delivery may be periodic or aperiodic. For instance, for push delivery, data may be sent, when a condition is met (e.g., the refrigerator notifies the user's PDA, that there is no milk) or periodically (a sensor reports measurements of the ozone level every hour). Finally, data may be received either by a single component or by a group of components. DBGlobe's data delivery mechanism must incorporate:

- Push and pull data
- Periodic and aperiodic
- Multicast and unicast.

Another issue is how to model the co-ordination of the mobile entities. We intend to use workflow management concepts. Workflow management is a technique to model, integrate and automate the execution of steps that comprise a complex process, such as a business activity. Recently, there has been a lot of research activity in the database community on the relation between workflows and advanced transaction models [GHS95]. Another approach to modeling is to adopt co-ordination concepts developed by the multi-agent community. In DBGlobe, we will investigate the use and integration of workflow and multi-agent co-ordination models to capture the co-ordination among the mobile entities and advanced transaction models to reason about the correctness of the interaction among the entities.

4.4.3 Information Discovery and Querying

In traditional distribute database management systems, users interact with the system, by posing queries in some declarative query language (e.g., SQL). The underlying system executes the query, by first computing an optimal, in terms of cost, plan and then shipping parts of the query at the appropriate database sites. Dynamic environments of autonomous and mobile entities make this query management paradigm obsolete and create vast opportunities for innovation.

The first fundamental challenge is defining a query language. What is an appropriate query language for querying autonomous and mobile entities? Which attributes of such entities should be queried? Such languages should have both a temporal and spatial dimension; querying the location of mobile entities is a central issue. Furthermore, even if location is not an explicit attribute in queries, the result may depend on the location of the data or the user (for instance as a query about the best near-by restaurant).

Traditional query languages assume that the user has complete knowledge of the underlying information captured by the database schema. Queries in DBGlobe must incorporate some of the features of search languages, since users do not have exact knowledge of the system.

Another central question is what the appropriate semantics of such languages are. There is imprecision both at the data sources and at the output of the query. One way to deal with imprecise queries is by progressive refinement: present the user with a coarse answer quickly and then refine it over time. This requires a synthesis of statistical techniques along with data delivery and user interfaces.

Then, the execution of such queries may require knowledge acquisition, since all data necessary to answer it may not be available.

Furthermore, while in traditional systems, the execution of a query is initiated by the user, in DBGlobe, queries may be triggered without the intervention of the user, for instance when a condition is met. Another form of query execution that is important in this context is continuous queries, e.g., queries that are re-evaluated periodically.

Also, since some of the devices (e.g., sensors) may continuously produce data, it may be necessary to have some form of filtering. In this case, data from many resources go through the filter and the relevant data are used. This asks for queries on infinite data streams rather than on finite stored data sets, as in traditional database research.

To summarize, DBGlobe re-defines query management to amalgamate:

- Knowledge acquisition,
- Filtering,
- Context-awareness,
- Implicit and continuous evaluation.

A traditional cost-based optimizer computes an optimal plan for executing a query by making assumptions unrealistic in global computing. There is a need for adaptive query optimizers and execution engines that cope with highly unpredictable and changeable environments. Figure 4.1 provides a survey on adaptive query processing [HFC+00]. Query processing in DBGlobe calls for fundamentally different approaches to the problem due to the increased complexity in a number of dimensions such as in hardware and workload complexity, data complexity, and complexity in the interaction among the units.

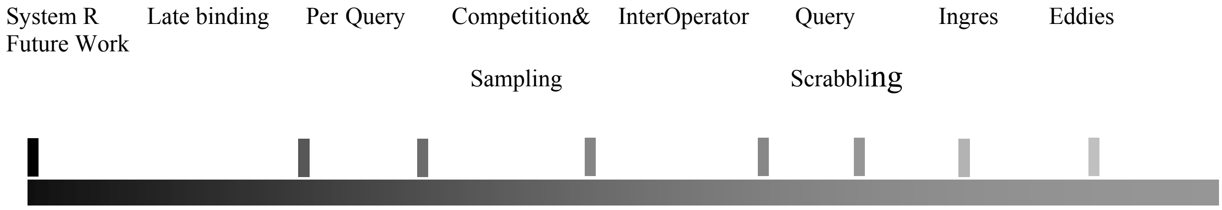

Fig.4.1. Evolution of Adaptive Query Processing Techniques [HFC+00]

4.4.4 Simulation Environment for Dynamic Entities

To test our research ideas, we intend to build a simulator for such dynamic environments. Our simulator will model mobile entities and their interactions. As the project proceeds, the simulator will be incrementally extended to model the creation of ad-hoc databases, delivery of data, co-ordination and querying.

Building such a simulator is a research challenge itself. Novel issues include:

- How to model the distribution, mobility and data of the mobile entities
- How to express the interaction among the entities
- How to capture the ad-hoc creation of databases

4.4.5 Proof-of-Concept Prototype: Location-Aware Querying

Our system architecture and its data delivery and query management components will be used to implement a proof-of-concept prototype. Our choice for prototype implementation is developing and ad-hoc database location-aware queries. The motivation for designing and implementing a prototype is to provide a concrete example for our formalisms.

The innovative features of our prototype include:

- Working example of the concept of "ad-hoc" database
- Use of metadata to include location and user preferences in dynamic query processing
- Co-ordination and co-operation of autonomous entities

Selected Area References

[ASILOMAR REPORT] P. A. Bernstein, et. al., The Asilomar Report on Database Research. *SIGMOD Record 27(4):* 74-80 (1998) (report of the meeting of 16 database system researchers whose goal was to access the database system research agenda for the next decade)

[GHS95] D. Georgeakopoulos, M. F. Hornick, A. P. Sheth: An Overview of Workflow Management: From Process Modeling to Workflow Automation Infrastructure. *Distributed and Parallel Databases* 3(2): 119-153 (1995)

[HFC+00] J. M. Hellerstein, M. J. Franklin, S. Chandrasekaran, et. al., Adaptive Query Processing: Technology in Evolution. *IEEE Data Engineering Bulletin 23*(2): 7-18 (2000)

4.5 Community Added Value and Contribution to EU Policies

"In future most objects that we work with will be equipped with processors and embedded software to perform and control a multitude of tasks in our everyday environment. Many of these objects will be able to communicate with each other and to interact with the 'environment' in carrying out the tasks they are designed for. With the internet (and its successors) sitting on top of communication networks of increasing flexibility and bandwidth, the result will be a massive networked infrastructure composed of highly diverse interconnected objects that should support the design and use of systems with a predictable and desirable behaviour."

GC Initiative Vision

The DBGlobe project focuses exactly on fundamental research for inventing a framework for Global Computing which will provide in the future the means for designing and using highly distributed and massively networked software systems which will be robust, secure, user-friendly and powerful. There are several projects and experimental systems that provide the (software) infrastructure for running computations on large collections of geographically dispersed computing resources. DBGlobe provides a unified framework for the seamless development of such systems.

Some examples of systems where the DBGlobe results can be applied include: user-programmable powerful handheld devices with corresponding telecommunication and information services, road and air traffic management systems (used e.g., for navigation, or taxi and carrier dispatching), systems that consist of mobile entities distributed over the internet. Also results from DBGlobe may impact research in distributed systems in the context of 'computational grids' and interactive multimedia services.

A Green Paper, published by the European Commission in December 1997 on the *convergence of the* telecommunications, media and information technology sectors, underlines that the new services which are developed and upon which the Information Society will be based must satisfy the following main requirements: they must increase the competitiveness of European businesses; substantially improve the quality of life of the European citizen; contribute to the integration of the world economy, but also, preserve and build on Europe's rich cultural diversity.

DBGlobe supported the offering of existing services/products, via the new global computing paradigm, and enhance the possibilities for the creation of new services and products. Applying the foundational work of DBGlobe, users of mobile phones will be offered relevant personalized services that suit their needs. DBGlobe will ensure constant and instant availability, which is the key concept for *European business competitiveness.*

Furthermore, the European Commission points out that new services developed need to *improve the quality of life of the European citizen.* As argued in the paper "*Reinforcing Cohesion and Competitiveness through Research, Technological Development and Innovation*", the social aspects of innovation need to be taken under consideration. The DBGlobe project offers new services to citizens on the move by offering future capability for enhancing existing services by the use of the personal mobile terminal as the channel for information access, relieving infrastructure and ameliorating the feeling of alienation of the European citizen. DBGlobe also targets at the enhancement of *user-friendliness of the information* society (the main IST focus). DBGlobe's vision is towards transparent and user-friendly access to data and applications. Adaptation to user's preferences is one of the key issues addressed by the project.

It is considered important to develop the *critical mass nation wide*. Another key issue is the need to establish critical mass of human capital, efficiently covered in European level, that is able to take up emerging responsibilities of new job descriptions in Information Society (e.g. mobile applications developers).

A central observation on Information Society is that new technologies offer new opportunities for social, economic and business transformation. As digital economy emerges in a global level, the confrontation of challenges will be beneficial in both national and international environments. In EU countries transformations are evident in the way that new forms of value chains are arising. DBGlobe results make feasible to transform traditional business models around new types of mediators with enabling mobile technologies. The international character of business arises the need for close co-operation among global partners that combine the necessary technology and business expertise in all aspects of this project, a fact that totally agrees with the project's consortium. Furthermore, the inter-European involvement in the DBGlobe project provides significant technological and business advantages to the European Research and Industry. Europe, as a community, depends on information interchange between the inhabitants of its single countries as well as interacting with the whole planet. It is therefore indispensable to carry out such *a project on a European level*.

Another central point is access to everything from everywhere. Through DBGlobe, distance is no longer important. Applications of DBGlobe research have the potential to assist the development of physically remote and underdeveloped regions.

DBGlobe explicitly follows the lines of the EU eEurope initiative that dictates the creation of a digitally literate and entrepreneurial Europe. The DBGlobe research part that focuses on communication related services will provide terminal and service independence, taking advantage of the convergence of existing cable and wireless networks and the potential synergies that can be derived from such a convergence. The key benefits that such systems promise include improvements in quality, incorporating broadband and networked services, flexibility of future service creation and introduction, and offering of ubiquitous service portability.

The DBGlobe project directly addressed the trite stated community need for

1. Information access to the user, regardless of location, network or terminal
2. Seamless delivery of all type of data and
3. High-quality, low-cost value for personal telecommunication services

The project will have also impact on the EU issue concerning the provision of interactive multimedia services for terrestrial and satellite broadcasting, and broadband wireless access. To this extend, the project also closely follows the standards that emerge as key elements of the multimedia application success. The MHP specification, which evolves as unique in involving all sectors of the industry and new products appear testing the market response to its concept, is highly considered by the DBGlobe applications that derive as MHP compatible. Within this context, DBGlobe will offer novel services to the European citizen.

5. Deliverables & References

5.1 Deliverables Table

Project Number: IST-2001-32645

Project Acronym: DBGlobe

Title: DBGlobe: A Data-Centric Approach to Global Computing

Del. No	Title	Type	Classifi-cation	Due Date	Issue Date
D1	Consortium Agreement	R	Int	2/2002	22/2/2002
D4.1	Project Presentation	R	IST	3/2002	29/3/2002
D18.1	Parameters and Criteria for Self - Assessment	R	Pub	6/2002	27/6/2002
D5	Dissemination and Use Plan	R	IST	6/2002	27/6/2002
D2	Metadata Management	R	Pub	7/2002	29/7/2002
D3	Overall System Architecture	R	Pub	9/2002	29/9/2002
D8	Data delivery mechanisms	R	Pub	11/2002	15/11/2002
D6	Initial Simulator Prototype and Simulator Web Page	O	Pub	1/2003	1/15/2003
D16.1	Dissemination Results	R	Pub	1/2003	1/15/2003
D18.2	Self Assessment Report	R	Pub	1/2003	1/15/2003
D4.2	Periodic Progress Report	R	IST	1/2003	1/15/2003
D9	Modeling Coordination through Workflows	R	Pub	4/2003	19/6/2003
D7	Final Simulator Prototype and Simulator Web page	O	Pub	6/2003	18/6/2003
D11	The Query Language of *DBGlobe*	O	Pub	6/2003	24/7/2003
D12	Query Optimization and Execution in *DBGlobe*	O	Pub	6/2003	28/7/2003
D16.2	Dissemenation Results	R	Pub	6/2003	10/8/2003

Del. No	Title	Type	Classifi-cation	Due Date	Issue Date
D14	**Location Aware Ad-hoc Databases and Query Processing**	R	Pub	8/2003	2/9/2003
D10	**Data Delivery and Querying – Results**	Conference	Pub.	8/2003	15/9/2003
D13	**Information Discovery and Querying – Results**	R	Pub	10/2003	5/12/2003
D15	**Prototype System for Location-Aware Query Processing**	O	Pub	10/2003	17/12/2003
D16.3	**Dissemination Results**	R	Pub	12/2003	22/2/2004
D18.3	**Self Assessment Report**	R	Pub	12/2003	22/2/2004
D4.3	**Periodic Progress Report**	R	IST	12/2003	22/2/2004
D19	**Proof - of - Concept Prototype and Demonstration**	Demo, R	Pub	12/2004	15/1/2005
D18.4	**Self Assessment Report**	R	Pub	4/2005	19/4/2005
D16.4	**Dissemination Results and Technological Implementation Plan**	R	Pub	4/2005	18/3/2005
D20	**Results of Clustering Activities**	R	Pub	4/2005	14/4/2005
D4.4	**Progress Report**	R	IST	12/2004	19/4/2005
D.4.4e	**Progress Report – extension period**	R	IST	4/2005	19/4/2005
D17	**Final Report**	R, Web page	Pub	4/2005	7/6/2005

5.2 Deliverable Summary Sheet

Project Number: IST-2001-32645

Project Acronym: DBGlobe

Title: DBGlobe: A Data-Centric Approach to Global Computing

Deliverable D2: "Metadata Management"
Due date: 7/2002
Delivery Date: 29/7/ 2002

Short Description:

Emerging computational paradigms such as global and ubiquitous computing require some rethinking and innovative research ideas in many computer science areas. In the DBGlobe project, we aim at studying a mobile computing scenario from a 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 the example application of tourist information services to abstract the requirements to such a computing environment. Using this more concrete handle, we outline the various existing types of data. Included here are 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 the concept of ontologies.

In DBGlobe, we present a data-based view of a global computing environment that relies on mobile clients. In this report, we focus on the overall data scenario and the issue of handling the metadata in such systems. 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 devices posing queries. Metadata stands for extracted structure and meaning of data. We envision that in a global computing environment, data sources possess, both, structure and semantics to a varying degree, e.g., ranging from relational databases to pure textual information. Further, we propose the use of UML (Universal Modeling Language) as a design and notational means. Finally, we introduce essential metadata to contain an abstract view of the content data and profile data.

Partner leading: CTI
Partners contributed: UoI, INRIA, UCR
Made available to: All

Deliverable D3: "Overall System Architecture"
Due date: 9/2002
Delivery Date: 29/9/2002

Short Description:

There is an increasing number of autonomous, diverse, mobile electronic devices in use, such as smart cards, personal digital assistants, sensors, and digital cameras. As a result, a great amount of data and services - that need to be accessed by individuals or communities - resides on these devices. This creates a massive infrastructure composed of highly diverse interconnected mobile entities as well as a virtual superdatabase of services and data. The DBGlobe project takes a data-centric approach to modeling, integrating, storing and querying data and services in such an environment. In this deliverable, we (a) describe the overall architecture of our approach, (b) focus on service discovery and indexing, (c) describe the functionality of the system and present an example application, (d) present more details on service composition and related metadata and finally, we compare our approach with related research.

Partner leading: CTI
Partners contributed: UoI, INRIA, UCR
Made available to: All

Deliverable D8: "Data Delivery Mechanisms"
Due date: 11/2002
Delivery Date: 15/11/2002

Short Description:

In global computing, data are delivered from sources to clients in a variety of modes. In this report, we survey the various such modes of data delivery and classify them along a number of dimensions. We also focus on the correctness properties of data delivery and discuss protocols that ensure both the timeliness and the consistency of data under the various data delivery mechanisms. Finally, we show how such mechanisms fit in the DBGlobe framework and present initial directions regarding an adaptive deployment of each mechanism.

Partner leading: UoI
Partners contributed: UoI, INRIA, CTI, AUEB, UCY, UCR
Made available to: All

Deliverable D1: "Consortium Agreement"
Due date: 2/ 2002
Delivery Date: 22/2/2002

Short Description:

In this report, we present a determination of a project co-ordination board, coordination, control and management of the project (preparation of meetings, preparation of periodic and final reports, administration of the project resources, monitoring of the overall progress).

Partner leading: UoI
Partners contributed:
Made available to: UoI, INRIA, CTI, AUEB, UCY, UCR, (and Commission Project Officer + reviewers if requested)

Deliverable D4.1: "Project Presentation"
Due date: 3/2002
Delivery Date: 29/3/2002

Short Description:

On one view, global computing can be seen as a database problem: how to design, build and analyze systems that manage large amount of data. However, the traditional database approach of storing data of interest in monolithic database management systems becomes obsolete in such environments. In current database research, data are relatively homogeneous, exhibit a small degree of distribution (just a few network sites) are passive in that they remain unchanged unless explicitly updated. All these assumptions do not hold in the global computing world. This creates the need for new theoretical foundations in all aspects of data management: modeling, storage and querying.

DBGlobe aims at revolutionizing the way we think of databases, as the mundane task of processing static data in monolithic database management systems. It broadens database management research focus to attack the issues of mobility, autonomy, incomplete information, scale, and adaptability that arise in dynamic environments. In this report, we present a short description of the DBGlobe project.

Partner leading: UoI
Partners contributed: All
Made available to: IST Programme participants

Deliverable D18.1: "Self Assessment Report"
Due date: 6/2002
Delivery Date: 27/6/2002

Short Description:

In this report, we set the criteria for the evaluation of the project. DBGlobe is a research project on future and emerging technologies. As such the focus of DBGlobe is on scientific innovation. Thus, our long-term goal is impact among the scientific community. Innovation in research is mainly assessed in the short term by *publications* in competitive forums. Thus, one main assessment criterion for the success of our effort is the number and the quality of publications produced through the research performed within the project. Such publications should appear in appropriate forums, such as competitive journals, conferences and workshops in the area of databases and distributed systems. Another evaluation criterion for our research will be through comparison of how our research outcome compares with those of other projects and research efforts in the same field. Besides publications, we also set in this report, some other quantitative assessment criteria. These criteria are specific for each workpackage and depend on the nature of research performed at each of them. Finally, to evaluate progress we have identified a set of milestones. We also report how we assess success of the related milestones in each workpackage.

Partner leading: UoI
Partners contributed: All
Made available to: All

Deliverable D5: "Dissemination and Use Plan"
Due date: 6/2002
Delivery Date: 29/3/2002

Short Description:

The DBGlobe project is part of the FET pro-active initiative on Global Computing. In this report, we set the goals related to the dissemination of our results.

Partner leading: UoI
Partners contributed: All
Made available to: IST Programme participants

Deliverable D6: "Initial Simulator Prototype and Simulator Web page"
Due date: 1/2003
Delivery Date: 15/1/2003

Short Description:

In this deliverable, we report on the DBGlobe simulator, called MobiShare, and its architecture. The MobiShare architecture outlined in this report provides the infrastructure for ubiquitous mobile access and mechanisms for publishing, discovering and accessing heterogeneous mobile resources in a large area, taking into account the context of both sources and requestors. Any wireless communication protocol could be used between the device and the system. The use of XML-related languages and protocols for describing and exchanging metadata gives the system a uniform and easily adaptable interface, allowing a variety of devices to use it. The overall approach is data-centric and service-oriented, implying that all the devices are treated as producers or requestors of data wrapped as information services.

Partner leading: AUEB
Partners contributed:All
Made available to: All

Deliverable D16.1: "Dissemination Results"
Due date: 1/2003
Delivery Date: 15/1/2003

Short Description:

In this report, we present the dissemination activities undertaken by the DBGlobe partners during the first year of the project. The activities include the resulted publications, the maintenance of the project webpage as well as a number of other activities towards the promotion of the project.

Partners owning: UoI
Partners contributed: All
Made available to: All

Deliverable D18.2: "Self Assessment Report"
Due date: 1/2003
Delivery Date: 15/1/2003

Short Description:

In this report we present the results of the self assessment of our work in the first year of the DBGlobe project according to the parameters and criteria set out in deliverable D18.1: "Parameters and Criteria for Self Assessment

Partners owning: UoI
Partners contributed: All
Made available to: IST Programme participants.

Deliverable D4.2: "Periodic Progress Report"
Due date: 1/2003
Delivery Date: 15/1/2003

Short Description:

In this report we give a detailed account of all the activities planned and carried out during the reporting period 1/1/2002 – 31/12/2002.

Partners owning: UoI
Partners contributed: All
Made available to: All

Deliverable D9: "Modelling Coordination through Workflows"
Due date: 4/2003
Delivery Date: 19/6/2003

Short Description:

A workflow is an activity involving the coordinated execution of multiple tasks by different processing entities. In this report, we investigate the use of workflows in both the specification of computation in DBGlobe as well as for providing execution guarantees. In particular, workflow concepts are used for detailing the tasks to be carried out and the execution requirements. Furthermore, transaction workflows are deployed for providing safeguards of traditional databases systems related to computation correctness, data consistency and durability. The computation model of DBGlobe is service-oriented. Thus, our approach is also related to service coordination and specification.

Partner leading: UoI
Partners contributed: INRIA, CTI, AUEB, UCY, UCR
Made available to: All

Deliverable D7: "Final Simulator Prototype and Simulator Web page"
Due date: 6/2003
Delivery Date: 19/6/2003

Short Description:

In dynamic global computing environments scalability and performance are key issues of system design. Therefore, in DBGlobe we explore the area of simulation trying to evaluate and adapt current techniques and practices to the demands of massive networked infrastructures composed of highly diverse interconnected mobile objects. Simulation will be used to analyze the expected system behaviour, make estimations regarding the flexibility, efficiency, dependability and robustness of the proposed DBGlobe architecture, understand its limits, potentials and decide on implementation strategies for the proof-of-concept prototype.

Partner leading: AUEB
Partners contributed: UoI, CTI, UCY, UCR
Made available to: All

Deliverable D11: "The Query Language of DBGlobe"
Due date: 6/2003
Delivery Date: 24/7/2003

Short Description:

The query language of DBGlobe is based on Active XML (AXML, for short), a declarative framework that harnesses web services for data integration, and is put to work in the DBGlobe peer-to-peer architecture. It is centred on AXML documents which are XML documents that may contain calls to web services. The language allows the specification of such documents, querying them, and defining new web services based on them. In particular, the language includes linguistic features to control the timing of service call activations, the lifespan of data, and the exchange of extensional and intensional data. This allows PMO's that use them to act in various scenarios including mediation, data warehousing, and distributed computation.

Partner leading: INRIA
Partners contributed: UoI, CTI, AUEB, UCY
Made available to: All

Deliverable D12: "Query Optimization and Execution in DBGlobe"

Due date: 6/2003

Delivery Date: 28/7/2003

Short Description:

Active XML (AXML for short) has been chosen as the data model and query language of DBGlobe. The goal of this deliverable is to study the new issues raised by the distribution and replication of Active XML data on DBGlobe PMOs, and develop appropriate query optimization and execution techniques for this context. Starting from a data model and a query language, we describe in this report a complete query optimization framework for distributed and replicated Active XML documents in DBGlobe. We provide a comprehensive cost model for query evaluation and show how it applies to user queries and service calls. Finally, we describe an algorithm that, for a given peer, chooses data and services that the peer should replicate to improve the efficiency of maintaining and querying its Active data.

Partner leading: INRIA

Partners contributed: UoI, CTI, AUEB, UCY

Made available to: All

Deliverable D16.2: "Dissemination Results"

Due date: 6/2003

Delivery Date: 10/8/2003

Short Description:

In this report, we present the dissemination activities undertaken by the DBGlobe partners during the first part of the second year of the project (Jan 2003 – June 2003). The activities include the resulted publications, the maintenance of the project webpage as well as a number of other activities towards the promotion of the project.

Partner leading: UoI

Partners contributed: All

Made available to: All

Deliverable D14: “Location Aware Ad-hoc Databases and Query Processing”
Due date: 8/2003
Delivery Date: 2/9/2003

Short Description:

Mobile devices (or PMOs) and wireless networking brings about a new computing paradigm where location and wireless services are of prime importance. Mobile devices as they relocate themselves create ad hoc databases around their current location. A client usually requests a particular service that is of close proximity, usually within his local Cell (or CAS). We can thus view location as a concept around which we posed queries. Concept is a semantic notion and describes a specific property, for example, “travelling”, “weather”, “taxi reservation” etc. In this deliverable we expand on this idea by creating ad hoc databases around concepts and not just location. We call these ad hoc databases, “communities” and instead of issuing queries just around the notion of location we do issue them over these communities and around the concept they represent. Context based queries, containment and continuous queries are also of special interest within this new idea of communities and concepts and we attempt to study them within this deliverable.

Partner leading: UCY
Partners contributed: All
Made available to: All

Deliverable D10: "Data Delivery and Querying – Results"
Due date: 8/2003
Delivery Date: 15/9/2003

Short Description:

The goal of this report is twofold: (a) to present some of our research results along data delivery mechanisms and (b) to present the query routing mechanisms of DBGlobe. Consequently, the report is structured into two parts. Part I focuses on data delivery, while Part II focuses on querying.

In global computing, data are delivered from sources to clients in a variety of modes. Broadcasting provides an efficient means for disseminating information in both wired and wireless setting. In Part I of this report, we propose a suite of broadcast organization schemes for multiversion data broadcast, i.e., data broadcast in which more than one value is broadcast per data item. Besides increasing the concurrency of client transactions, multiversion broadcast provides clients with the possibility of accessing multiple server states. For example, such functionality is essential to support applications that require access to data sequences and have limited local memory to store the previous versions, such as in the case of sensor networks. We identify two basic multiversion organizations, namely vertical and horizontal broadcasts and propose an efficient compression scheme applicable to both. We also consider a multiversion client data cache and introduce appropriate cache replacement techniques. Finally, we propose an adaptive scheme that dynamically selects the appropriate broadcast organization based on the client access pattern. We provide performance evaluation results for both flat and broadcast disk organizations and for a variety of client query patterns.

An important challenge in a data management system for global computing, such as DBGlobe, is discovering the appropriate data and services. In Part II of this report, we propose a routing mechanism for queries. In DBGlobe, services and data, collectively called resources, are described using hierarchically structured metadata. There is no centralized index for the resources; instead, appropriately distributed filters are used to route queries to the appropriate nodes. We propose two new types of filters that extend Bloom filters for hierarchical documents. Two alternative ways are considered for building overlay networks of nodes: one based on network proximity and one based on content similarity. Content similarity is derived from the similarity among filters. Our experimental results show that networks based on content similarity outperform those formed based on network proximity for finding all matching documents

Partner leading: UoI
Partners contributed: All
Made available to: All

Deliverable D13: "Information Discovery and Querying – Results"
Due date: 10/2003
Delivery Date: 5/12/2003

Short Description:

In this report we detail the results of our research regarding Information Discovery and Querying in DBGlobe. As described in the previous reports, DBGlobe is using the Active XML (AXML) language and system as basis for information management. We detail here the advances we made regarding the use of AXML as a tool for the discovery and querying of information held by PMOs.

We finished the implementation of the concepts described in the previous deliverables. Next we built two system prototypes to validate the use of AXML as the basis for information discovery and querying tools. The two prototypes were demonstrated in VLDB'03. The first demo illustrates the use of AXML for the construction of a peer-to-peer news publication and syndication system. The second demo illustrates the use of AXML as platform for the management of distributed workspaces. While developing these two applications we have discovered that an important issue for query optimization, in this context, is the lazy calls to services. So, some special work was devoted to solve the problem. Finally, as mentioned in the proposal, the discovery of services to be used in the system is a key issue.

Partner leading: INRIA
Partners contributed: UoI, CTI, AUEB, UCY
Made available to: All

Deliverable D15: “Prototype System for Location-Aware Query Processing”
Due date: 10/2003
Delivery Date: 17/12/2003

Short Description:

Mobile devices (or PMOs) and wireless networking brings about a new computing paradigm where location and wireless services are of central importance. Mobile devices as they relocate themselves create ad hoc databases around their current location. A client usually requests a particular service that is of close proximity, usually within his local Cell (or CAS). We can thus view location as a concept around which we pose queries. Concept is a semantic notion and describes a specific property, for example, “traveling”, “weather”, “taxi reservation” etc. In this deliverable, we provide a description of the proof of concept. In particular, we will present a prototype implementation of an application that uses the “communities” that were described in deliverable D14.

Partner leading: UCY
Partners contributed: All
Made available to: All

Deliverable D16.3: “Dissemination Results”

Due date: 12/2003

Delivery Date: 22/2/2004

Short Description:

In this report, we present the dissemination activities undertaken by the DBGlobe partners during the second part of the second year of the project. The activities include the resulted publications, the maintenance of the project webpage as well as a number of other activities towards the promotion of the project.

Partner leading: UoI

Partners contributed: All

Made available to: All

Deliverable D18.3: "Self Assessment Report"

Due date: 12/2003

Delivery Date: 22/2/2004

Short Description:

In this report, we present the results of the self assessment of our work in the second year of the DBGlobe project according to the parameters and criteria set out in deliverable D18.1: "Parameters and Criteria for Self Assessment".

Partners owning: UoI

Partners contributed: All

Made available to: All

Deliverable D4.3: "Periodic Progress Report"

Due date: 12/2003

Delivery Date: 22/2/2004

Short Description:

In this report we give a detailed account of all the activities planned and carried out during the reporting period 1/1/2003 – 31/12/2003.

Partners owning: UoI

Partners contributed: All

Made available to: IST Programme participants.

Deliverable D19: "Proof –of- Concept Prototype and Demonstration"
Due date: 12/2004
Delivery Date: 10/1/2005

Short Description:

Mobile devices (or PMOs) and wireless networking brings about a new computing paradigm where location and wireless services are of central importance. In DBGlobe, we consider semantic service discovery in a global computing environment. We propose creating a dynamic overlay network by grouping together semantically related services. Each such group of services is termed a community. Communities are organized in a global taxonomy whose nodes are related contextually. The taxonomy can be seen as an expandable distributed semantic index over the system, which aims at improving service discovery. In a previous report (Deliverables D15), we have presented both our implementation of an infrastructure for supporting communities as well as of our prototype application, namely SpotMe that utilizes this infrastructure. In this report, we present our enhanced distributed service discovery mechanism that utilizes communities for context-based service discovery. We also present performance results of the deployment of our prototype application in a distributed setting. Finally, we present our experiences of incorporating in our prototype small devices, such as camera and sensor devices.

Partner leading: UCY
Partners contributed: INRIA, CTI, AUEB, UoI, UCR
Made available to: All

Deliverable D4.4: "Periodic Progress Report"
Due date: 12/2004
Delivery Date: 19/4/2005 (Revised version)

Short Description:

In this report we give a detailed account of all the activities planned and carried out during the reporting period 1/1/2004 – 31/12/2004.

Partners owning: UoI
Partners contributed: All
Made available to: All

Deliverable D4.4e: "Periodic Progress Report"
Due date: 4/2005
Delivery Date: 19/4/2005 (Revised version)

Short Description:

In this report we give a detailed account of all the activities planned and carried out during the extension period 1/1/2005 – 28/4/2005.

Partners owning: UoI
Partners contributed: All
Made available to: All

Deliverable D18.4: "Self Assessment Report"
Due date: 4/2005
Delivery Date: 19/4/2005 (Revised version)

Short Description:

In this report we present the results of the self assessment of our work in the last 16 months of the DBGlobe project according to the parameters and criteria set out in deliverable D18.1: "Parameters and Criteria for Self Assessment".

Partners owning: UoI
Partners contributed: All
Made available to: All

Deliverable D16.4: “Dissemination Results and Technological Implementation Plan”
Due date: 4/2005
Delivery Date: 18/3/2005

Short Description:

In this report, we present the dissemination activities undertaken by the DBGlobe partners during the second part of the last 16 months of the project. The activities include the resulted publications, the maintenance of the project webpage as well as a number of other activities towards the promotion of the project.

Partners owning: UoI
Partners contributed: All
Made available to: All

Deliverable D20: “Results of Clustering Activities”
Due date: 4/2005
Delivery Date: 14/4/2005

Short Description:

In this report, we describe the activities related to the integration of DBGlobe project with the other project of the Global Computing project (both with projects inside the “Foundation of Networks and Large Distributed Systems” cluster and with related projects at other clusters).

Partners owning: UoI
Partners contributed: All
Made available to: All

Deliverable D17: "Final Report"
Due date: 4/2005
Delivery Date:

Short Description:

In this report , we give a detailed account of all the activities planned and carried out during the reporting period 1/1/2002 – 28/4/2005.

Partners owning: UoI
Partners contributed: All
Made available to: All

5.3 DBGlobe Publications

A1. D. Pfoser and C. S. Jensen. Trajectory Indexing Using Movement Constraints. GeoInformatica Journal, to appear, June, 2005.

A2. N. Tryfona and D. Pfoser. Data Semantics in Location-Based Services. Journal on Data Semantics, to appear, April 2005

A3. S. Brakatsoulas, D.Pfoser, and Nectaria Tryfona. Practical Data Management Techniques for Vehicle Tracking Data. 21st International Conference on Data Engineering (ICDE), to appear, 2005

A4. S. Brakatsoulas, D. Pfoser, and N. Tryfona. Modelling Storing and Mining Moving Object Databases. 8th International Database Engineering & Applications Symposium (IDEAS), 2004

A5. D. Pfoser, N. Tryfona and V. Verykios, "Services-Based Data Management in a Global Computing Environment", Third International Workshop on Web and Wireless Geographical Information Systems (W2GIS 2003)

A6. L. Speicys, C. S. Jensen, A Kligys, "Computational Data Modeling for Network-Constrained Moving Objects", in proceedings of the 11th ACM International Symposium on Advances in Geographical Information Systems (ACM-GIS), 2003.

A7. D. Pfoser, and C. S. Jensen, "Indexing of Network-Constrained Moving Objects", in proceedings of the 11th ACM International Symposium on Advances in Geographical Information Systems (ACM-GIS), 2003.

A8. M. A. Nascimento, D. Pfoser, and Y. Theodoridis, "Synthetic and Real Spatiotemporal Datasets", Invited contribution DEBulletin, June 2003.

A9. D. Pfoser, E. Pitoura, and N. Tryfona. Metadata Modeling in a Global Computing Environment. Proceedings of the 10th ACM International Symposium on Advances in Geographic Information Systems, McLean, VA November 8-9, 2002.

A10. C. Doulkeridis, V. Zafeiris and M. Vazirgiannis. "The Role of Caching and Context-Awareness in P2P Service Discovery", to appear in the 6th International Conference on Mobile Data Management (MDM'05), Ayia Napa, Cyprus, May 9-13, 2005.

A11. C. Doulkeridis and M. Vazirgiannis. "Querying and Updating a Context-Aware Service Directory in Mobile Environments", proceedings of the 2004 IEEE/WIC/ACM International Conference on Web Intelligence (WI'04), Beijing, China, pages 562-565, September 20-24, 2004.

A12. C. Doulkeridis and M. Vazirgiannis. “Updating a Context-Aware Service Directory for M-Services”, in the 3rd Hellenic Data Management Symposium, HDMS'04 , Athens, Greece, June 28-29, 2004.

A13. E. Valavanis, C. Ververidis, M. Vazirgiannis, G.C. Polyzos, K. Nørvåg, “MobiShare: Sharing Context-Dependent Data and Services from Mobile Sources", Procs of the 2003 IEEE/WIC International Conference on Web Intelligence WI 2003 October 13-17, 2003, Halifax, Canada

A14. C. Doulkeridis, E. Valavanis and M. Vazirgiannis. “Towards a Context-Aware Service Directory”. Proceedings of the 4th VLDB Workshop on Technologies for E-Services (TES'03), Sept.8 , 2003, Berlin, Germany. Proceedings published in the Springer Lecture Notes in Computer Science (LNCS) series.

A15. C. Ververidis, S. Valavanis, M. Vazirgiannis, G.C. Polyzos, “An Architecture for Sharing, Discovering and Accessing Mobile Data and Services: Location and Mobility Issues”, Presented at: Lobster Workshop, LBS for accelerating the European-wide deployment of Services for the Mobile User and worker, Mykonos, Greece, 4-5 October, 2002, http://www.iit.demokritos.gr/lobster/lobster_mykonos2002.html

A16. A. S. Andreou, S. Mavromoustakos, C. Leonidou ,C. Chrysostomou, A. Pitsillides, G. Samaras, C. Schizas, “Key Issues for the Design and Development of Mobile Commerce Services and Applications”, International Journal of Mobile Communications (IJMC), to appear.

A17. S. Polyviou, G. Samaras and P. Evripidou, “A Relationally Complete Visual Query Language for Heterogeneous Data Sources and Pervasive Querying”, Proc. of the 19th International Conference on Data Engineering, Sponsored by the IEEE Computer Society, Tokyo, Japan, April 5-8, 2005.

A18. C. Panayiotou and G. Samaras, “Personalized Portals for the Wireless User: An Agent Approach”. Journal of ACM/Baltzer Mobile Networking and Applications (MONET), special issue on “Mobile Commerce", Volume 9, Issue 6, 663–677, 2004, Kluwer Academic Publishers.

A19. O. Papapetrou and G. Samaras, “Optimizing Distributed Web Crawling by Minimizing the Network Distances”, Proc. of the 9th IFCIS International Conference on Cooperative Information Systems (CoopIS΄2004), Agia Napa, Cyprus, October 2004.

A20. O. Papapetrou and G. Samaras, “IPMicra: Toward a distributed and adaptable location aware web crawler”, Proc. 8th East-European Conference on Advances in Databases and Information Systems (ADBIS 2004), 22-25 September, 2004, Budapest, Hungary.

A21. S. Polyviou, P. Evripidou and G. Samaras, “Query by Browsing: A Visual Query Language Based on the Relational Model and the Desktop User

Interface Paradigm", The 3rd Hellenic Symposium on Data Management, (HDMS04), Athens, Greece, 28-29 June 2004.

A22. S. Polyviou, P. Evripidou and G. Samaras, "PALLAS: A Querying Interface for Pervasive Computing Using Handheld Devices", The 2004 ACS/IEEE International Conference on Pervasive Services (ICPS'2004), American University of Beirut, Lebanon, July 19-23, 2004.

A23. O. Papapetrou and G. Samaras, "IPMicra: An IP-address Based Location Aware Distributed Web Crawler", The 2004 International Conference on Internet Computing (IC'04): June 21-24, 2004, Las Vegas, Nevada, USA.

A24. S. Polyviou, P. Evripidou and G. Samaras, "Context-Aware Queries using Query by Browsing and Chiromancer", 2nd International Conference on Pervasive Computing, Linz / Vienna, Austria, April 18-23, 2004. Publish in the "Advances in Pervasive Computing" book of the OCG (Vol 176 ISBN 3-85403-176-9)..

A25. O. Papapetrou and G. Samaras, "IPMicra: A Distributed Location-Aware Crawler", Poster Session, 13th International World Wide Web Conference (WWW2004), 19-22 May 2004, New York, New York, USA

A26. G. Samaras, "Mobile Agents: What about them? Did they deliver what they promised? Are they here to stay?" Proc. of the 2004 IEEE International Conference on Mobile Data Management (MDM2004), Berkeley, California, USA, January 19-22, 2004.

A27. S. Papastavrou, G. Samaras, P. Evripidou and P. K. Chrysanthis, "Fine-Grained Parallelism in Dynamic Web Content Generation: The Parse & Dispatch Approach", Proc. of the 9th IFCIS International Conference on Cooperative Information Systems (CoopIS΄2003), Italy, November 2003

A28. O. Papapetrou, S. Papastavrou, and G. Samaras, "Distributed Indexing of the Web Using Migrating Crawlers". 12th International World Wide Web Conference (W3) 20-24 May 2003, Budapest, Hungary. (poster)

A29. S. Papastavrou, G. Samaras, P. Evripidou and P. K. Chrysanthis, "Parse & Dispatch: Parallelizing the Generation of Dynamic Web Content", 12th International World Wide Web Conference (W3) 20-24 May 2003, Budapest, Hungary. (poster)

A30. C. Panayiotou, G. Samaras, "Personalized Portals for the Wireless User Based on Mobile Agents: Demonstration", 19th International Conference on Data Engineering, IEEE Computer Society, March 5 - March 8, 2003 - Bangalore, India. (demo)

A31. G. Samaras, C. Panayiotou, "A Flexible Personalization Architecture for Wireless Internet Based on Mobile Agents", Proc. 6th East-European Conference on Advances in Databases and Information Systems (ADBIS 2002), September 2002, Bratislava, Slovakia.

A32. C. Skouteli, G. Samaras and E. Pitoura, "Concept-Based Discovery of Mobile Services" Proc. of the 5th IEEE International Conference on Mobile Data Management (MDM2005), Agia Napa, Cyprus, May 9-13, 2005.

A33. K. Karenos, G. Samaras, E. Pitoura and P. K. Chrysanthis, "Mobile Agent-Based Services for View Materialization". The ACM Mobile Computing and Communications Review (MC2R), special issue on "Recent Advances in Mobile Data Management Issues", Volume 1, Number 2, December, 2004.

A34. C. Spyrou, G. Samaras, E. Pitoura and E. Paraskevas "Mobile Agents for Wireless Computing: The Convergence of Wireless Computational Models with Mobile-Agent Technologies", Journal of ACM/Baltzer Mobile Networking and Applications (MONET), special issue on "Mobility in Databases & Distributed Systems ", Volume 9, Issue 5 (Oct. 2004).

A35. O. Shigiltchoff, P.K. Chrysanthis and E. Pitoura. "Adaptive Multiversion Data Broadcast Organizations" Information Systems Journal, Volume 29, Issue 6, Pages 509-528, September 2004

A36. O. Shigiltchoff, P. K. Chrysanthis and E. Pitoura. Energy Efficient Access in Multiversion Broadcast Environments. 5th IEEE International Conference on Mobile Data Management, Berkeley, California, USA, January 19-22, 2004 (short paper)

A37. G. Samaras, C. Skouteli, C. Panayiotou and E.Pitoura, "Communities: Concept-Based Querying of Mobile Services", International Workshop on Global Computing 2004 (GC 2004). Published in the LNCS volume of the post-proceedings of the Global Computing 2004 workshop.

A38. G. Samaras, K. Karenos, E. Pitoura, and P. K. Chrysanthis, "ViSMA: Extendible, Mobile-Agent Based Services for the Materialization and Maintenance of Personalized and Shareable Web Views", Proc. of the 6th International Workshop "Mobility in Databases & Distributed Systems" DEXA´03, September 1-5, 2003, Prague, Czech Republic.

A39. G. Kastidou, E. Pitoura and G. Samaras, "A Scalable Mobile Agent Location Mechanism", Accepted for Publication 1st International Workshop on Mobile Distributed Computing (MDC'03), May 19, 2003, held in conjunction with the 23rd International Conference on Distributed Computing Systems (ICDCS'03).

A40. E. Pitoura and P. Chrysanthis. "Multiversion Data Broadcast", IEEE Transactions on Computers 51(10):1224-1230, October, 2002.

A41. G. Samaras, C. Spyrou, E. Pitoura, "View Generator (VG): A Mobile Agent Based System for the Creation and Maintenance of Web Views", 7th IEEE Symposium on Computers and Communications, Taormina, Italy July 2002.

A42. O. Shigiltchoff, P. Chrysanthis and E. Pitoura. “Multi-version Data Broadcast Organizations”. In Proceedings of the 6th East European Conference on Advances in Databases and Information Systems (ADBIS), September 2002, Bratislava, Slovakia.

A43. S. Abiteboul, O. Benjelloun, B. Cautis, I. Manolescu, T. Milo and N. Preda. “Lazy Query Evaluation for Active XML”. Proceedings of the 2004 ACM SIGMOD international conference on Management of Data, SIGMOD 2004. pp. 227-238, June 2004, Maison de la Chimie, Paris, France.

A44. S. Abiteboul. “Distributed Information Management with XML and Web Services”. European Joint Conferences on Theory and Practice of Software (ETAPS) March 2004, Barcelona, invited paper. Proceedings of FASE' 2004 published by Springer-Verlag in the Lecture Notes on Computer Science, Volume 2984/2004, pp 1-11, April 2004.

A45. T. Milo, S. Abiteboul, B. Amann, O. Benjelloun, F. D. Ngoc, “Exchanging Intentional XML Data”, Procs of the 2003 ACM SIGMOD International Conference on Management of Data, pp 289-300, San Diego, California, USA, June 9-12, 2003.

A46. S. Abiteboul, A. Bonifati, G. Cobena, I. Manolescu and T. Milo, “Dynamic XML Documents with Distribution and Replication”, Procs of the 2003 ACM SIGMOD International Conference on Management of Data, pp 527-538, San Diego, California, USA, June 9-12, 2003.

A47. S. Abiteboul, J. Baumgarten, A. Bonifati, G. Cobéna, C. Cremarenco, F. Dragan, I. Manolescu, T. Milo, N. Preda, “Managing Distributed Workspaces with Active XML”, Demo, Procs. of VLDB’03, Sept 2003 (demo)

A48. S. Abiteboul, B. Amann, J. Baumgarten, o. Benjelloun, F. D. Ngoc, and T, Milo, “Schema-Driven Customization of Web Services”, Demo, Procs. of VLDB’03, Sept 2003 (demo)

A49. The Active XML team. Active XML primer, INRIA report, (Gemo-Report-275) April 2003

A50. S. Abiteboul, O. Benjelloun, I. Manolescu, T. Milo and R. Weber, “Active XML: Peer-to- Peer Data and Web Services Integration (demo)”, Proceedings of the 28th VLDB Conference, Hong Kong, 2002.

A51. K. Stefanidis, E. Pitoura and P. Vassilaidis On Supporint Context-Aware Preferences in Relational Database Systems In proceedings of the MDM 2005 Workshop on "Managing Context Information in Mobile and Pervasive Environments" (MCMP05), Ayia Napa. Cyprus, May 9 2005

A52. O. Shigiltchoff, P. K. Chrysanthis and E. Pitoura. “Adaptive Multiversion Data Broadcast Organizations”. Information Systems Journal. 29(6): 509-528, 2004.

A53. G. Koloniari and E. Pitoura, Filters for XML-based Service Discovery in Pervasive Computing. Computer Journal, Special Issue on Mobile and Pervasive Computing, Volume 47, No 4, July 2004, pp 461-474

A54. G. Koloniari, E. Pitoura. "Content-based Routing of Path Queries in Peer-to-Peer Systems". EDBT 2004.

A55. I. Kyriazis and I. Fudos, "Thin Client Access to a Visualization Environment", Third International Workshop on Graphics and Geometric Modelling CGGM04, in conjunction with ICCS 2004.

A56. E. Leontiadis, V. V. Dimakopoulos and E. Pitoura, "Cache Updates in a Peer-to-Peer Network of Mobile Agents", 4th IEEE International Conference on Peer-to-Peer Computing (P2P 2004), Zurich, Switzerland, August 2004

A57. Y. Petrakis, G. Koloniari and E. Pitoura, "On Using Histograms as Routing Indexes in Peer-to-Peer System", 2nd International Workshop on Databases, Information Systems and Peer-to-Peer Computing, In conjunction with VLDB2004, August 29-30, Toronto, Canada

A58. Y. Petrakis and E. Pitoura, "On Constructing Small Worlds in Unstructured Peer-to-Peer Systems", In the EDBT International Workshop on Peer-to-Peer Computing and Databases, Heraklion, Crete, Greece, March 14, 2004

A59. O. Shigiltchoff, P. K. Chrysanthis, E. Pitoura, "Energy Efficient Access in Multiversion Broadcast Environment". Mobile Data Management 2004, page 168 (short paper)

A60. A. Zarras, P. Vassiliadis and V. Issarny. Model-Driven Dependability Analysis of Web Services . Procs of the 6th International Symposium on Distributed Objects and Applications (DOA'04), 2004.

A61. V. V. Dimakopoulos and E. Pitoura. "A Peer-to-Peer Approach to Resource Discovery in Multi-Agent Systems", 7th International Workshop Cooperative Information Agents, CIA 2003, Helsinki, Finland, August 27-29, 2003, pp 62-77

A62. G. Koloniari and E. Pitoura, "Bloom-based Filters for Hierarchical Data", 5th Workshop on Distributed Data and Structures (WDAS), June 2003- extended version to appear in the workshop post-proceeding published by Carleton Scientific.

A63. G. Koloniari, Y. Petrakis and E. Pitoura. "Content-Based Overlay Networks of XML Peers Based on Multi-Level Bloom Filters". International VLDB Workshop on Databases, Information Systems and Peer-to-Peer Computing, 2003

A64. E. Pitoura, P. K. Chrysanthis and K. Ramamritham. “Characterizing the Temporal and Semantic Coherency of Broadcast-based Data Dissemination”. Proc. of the International Conference on Database Theory, January 2003, Siena, Italy.

A65. A. Karakasidis and E. Pitoura, “DBGlobe: A Data-Centric Approach to Global Computing”. International Workshop on Smart Appliances and Wearable Computing (IWSAWC 2002) In conjunction with ICDCS 2002, Vienna, Austria, July 2002

A66. Th. Korakis and L. Tassioulas. “An Approach Towards QoS Provisioning in Wireless LAN Complying to the 802.11e Framework”. IEEE Lanman 2002 Workshop, August 2002, Stockholm, Sweden.

A67. P. Triantafillou, and A. Economides, “Subscription Summarization: A New Paradigm for Efficient Publish/Subscribe Systems”, IEEE International Conference on Distributed Computing Systems, ICDCS, 2004

A68. P. Triantafillou and I. Aekaterinides ProxyTeller: “A Proxy Placement Tool for Content Delivery under Performance Constraints”, 4th Web Information Systems Engineering Conference (IEEE), Dec 2003.

A69. P. Triantafillou and A. Economides, “Subscription Summaries for Scalability and efficiency in Publish/Subscribe Systems”, 1st Intl. IEEE Workshop on Distributed Event-based Systems, (DEBS02) July 2002.

A70. P. Triantafillou and I. Aekaterinides, “Web Proxy Cache Placement, Replacement and the Proxy Teller”, 2nd International IEEE Symposium on Network Computing and Applications April, 2003.

A71. P. Triantafillou, I. Aekaterinides, "ProxyTeller: A Tool Guiding Web Proxy Placement Decisions", 12th International World Wide Web Conference (WWW) May, 2003, (poster).

Also a significant number of invited papers has been produced as a result of the work in the DBGlobe project. These are:

I1. Evaggelia Pitoura, George Samaras, Can Turker, “Seamless Consistency”, Book Chapter in “Mobile Middleware”, edited by Paolo Bellavista, Antonio Corradi, to appear 2005.

I2. Evaggelia Pitoura (UoI) was (with P. K. Chrysanthis) the special issue editors for the topic “Data Engineering of Mobie and Wireless Access” for the Wireless Network Journal, Vol 10, Issue 6, 2004

I3. George Samaras (UCY) was a panelist organizer and moderator: "Mobile Agents: What about them? Did they deliver what they promised? Are they here to stay?” main panel of the 2004 IEEE International Conference on Mobile Data Management (MDM2004), Berkeley, California, USA, January 19-22, 2004.

I4. Serge Abiteboul: Distributed information management with XML and Web Services. European Joint Conferences on Theory and Practice of Software (ETAPS), March 2004, Barcelona. Invited talk. In proceedings of FASE'2004

I5. Evaggelia Pitoura, George Samaras, Georgia Kastidou, "Locating Mobile Objects", Book Chapter, "Handbook of Mobile Computing", (Imad Mahgoub, Editor), CRC Press, December 2004.

I6. E. Pitoura, S. Abiteboul, D. Pfoser, G. Samaras and M. Vazirgiannis, "DBGlobe: A Service-Oriented P2P System for Global Computing", Invited contribution, Sigmod Record 32(3), pp.77-82, September 2003

I7. M. A. Nascimento, D. Pfoser, and Y. Theodoridis, "Synthetic and Real Spatiotemporal Datasets", Invited contribution, IEEE Data Engineering Bulletin, June 2003.

I8. G. Samaras invited as Panel chair in the 5th Mobile Data Management (MDM) conference, Berkley, California, January 19-22.

I9. Serge Abiteboul: Managing an XML Warehouse in a P2P Context. The 15th International Conference on the Advanced Information Systems Engineering, (CAiSE 2003), Klagenfurt, Austria, June 16-18, 2003. Keynote talk.

These papers have been submitted for publication.

S1. D. Pfoser and N. Tryfona, "The Use of Ontologies in Location-Based Services: The Space and Time Ontology in Protégé-2000", under submission.

S2. P. Triantafillou and I. Aekaterinides, "Guiding Web Proxy and Server Placement for High Performance Internet Content Delivery"

S3. G. Samaras, C. Skouteli, C. Panayiotou, E. Pitoura, "Communities: Creating and Quering Ad-hoc Databases based on Concepts", submitted

S4. George Samaras, Maria Antreou, Christoforos Panayiotou, Andreas Pitsillides, "Time Based Personalization for the Moving User", submitted

6. Potential impact

DBGlobe innovation lies on:

- Service-orientation: data are wrapped into services
- Semantic service discovery: a service and a parameter ontology are used to select the appropriate service
- Community concept: data are clustered into communities based on context
- Mobility support
- Active XML language: a language that allows service calls inside XML documents

We provide a qualitative assessment of the work carried out per workpackage and the potential impact of related results.

6.1 System Architecture (WP1)

The main target in the first work package (WP1) of DBGlobe was to derive appropriate architectures for ad-hoc databases of mobile entities. Such architectures should be metadata driven in order to facilitate transparent interoperability among the involved entities, and to enhance location management, which is a critical aspect of a global computing system that relies mostly on mobile distributed entities.

In a nutshell, the technical results that were attained are:

- A service-oriented architecture in which data are encapsulated within services to support interoperability and extensibility
- An analysis of metadata and models for their representation and storage
- General research results on content caching
- A location-based distribution of DBGlobe servers that supports extensibility

In order to be able to assess the progress in this workpackage, we considered to what extent the objectives comprising the milestones of WP1 have been achieved. In particular, the following objectives with respect to the system architecture of the DBGlobe system have been met:

- We defined what is the appropriate metadata information to describe mobile entities; such metadata include information in various levels of detail and granularity,
- We derived an appropriate and universal metadata definition and manipulation language for the effective and efficient definition, management and maintenance of the scores of the available metadata,
- We designed distribution and replication protocols for the DataStores that are mainly storage devices and hold metadata among other things, and the main computational entities of the system (called DataHandlers in the initial architecture and CAS in the revised one).
- We designed our architecture to be dynamically configurable and extensible, in order to accommodate for possible expansions in the scale of the participating entities.
- Our distributed design achieves fault-tolerance and availability.

We used the simulation environment WP2 to verify that:

- every component of the system was examined with respect to the provision of the appropriate services or functionalities,
- the appropriateness of the interfaces among the various components.

6.2 Simulation Enviroment (WP2)

The aim of Workpackage 2 (WP2) is to design and implement a virtual testbed of the proposed DBGlobe architecture (WP1), which simulates a real-world dynamic environment of mobile entities able to co-operate with each other. This simulator will then be extended to a proof-of-concept prototype of DBGlobe (WP5). A detailed description of the simulator is provided in deliverable D6: “Initial simulator prototype and web page”.

Evaluation of the Simulator

The simulation system must be thoroughly tested according to widely approved testing techniques. Testing is not simply the process of performing tests to the completed system (i.e. the location-aware tourist information exchange application). The most common testing practice consists of 5 test-tasks that should be applied recursively for each subsystem if necessary to uncover errors related to the design, development and integration process.

(1) Unit Code-Test: focuses testing on each separate unit (or component) as implemented in source code, to verify that it functions properly. Units (PMO server module, PMO client module, registration, authentication, login, session management, handover, location service) are tested making heavy use of white-box techniques. If any problems occur testing is reapplied on the unit that appears to be generating errors.

(2) Integration (or Design) Test: is applied after the process of assembling the units and integrating them to produce the complete system. The focus of testing is now shifted to the design and the construction of the software architecture and the data-flows between the subsystems. Heavy use of black-box test-case design techniques perform post-integration tests on the units, to locate integration errors or communication problems.

(3) Requirements Validation Test: At this stage the requirements established during the early stages are validated against the characteristics of the system that has been constructed and assembled. Final assurance that the software meets all functional, behavioral and performance requirements, using exclusively black-box techniques.

(4) System Engineering Test: Software, hardware, framework and other system elements are tested as a whole combined with additional external entities that affect the use of the simulator (e.g. users, devices, external services). At this final stage the testing process will have to focus on correct extraction of metadata about the devices and ensure that the system tracks and interprets correctly the location and movement of pmo devices.

It is clear that this 5-step testing process will have to be reapplied after the completion of the proof-of-concept prototype and the simulation environment will be reassessed at that point as well.

The testing of the simulator is still in progress and the results presented in deliverable D7.

Evaluation of research results

A first evaluation cycle was performed by an internal committee, the second by a consortium committee that was appointed in the third meeting.

Interesting research results have been produced in the following areas:

- modeling the distribution, mobility and the data contents of mobile entities.
- expressing the interaction among the entities.
- forming dynamic communities of mobile objects that act as ad-hoc databases.
- running computations and integrating data on large collections of geographically dispersed computing resources.
- capturing mobility patterns of users.
- specifying a formal language that allows service description and discovery.
- mobility issues in cell based systems.

6.3 Data Delivery and Coordination (WP3)

Introduction

One important aspect in DBGlobe is how data are communicated among the components that participate in the system: (a) the mobile entities, (b) the DBGlobe servers and (c) the users. Current database servers support a pull-based mechanism of data delivery: data are sent to clients after having been explicitly requested. This request-response time of data delivery is not appropriate for all interactions in the DBGlobe system. Some of the mobile entities deliver information in a push-based fashion, that is without an explicit request. Both push and pull data delivery may be periodic or aperiodic. For instance, for push delivery, data may be sent, when a condition is met (e.g., the refrigerator notifies the user's PDA, that there is no milk) or periodically (a sensor reports measurements of the ozone level every hour). Finally, data may be received either by a single component or by a group of components. DBGlobe's data delivery mechanism must incorporate:

- Push and pull data
- Periodic and aperiodic
- Multicast and unicast.

In global computing, data are delivered from sources to clients in a variety of modes. In Deliverable D8 we have studied the properties of the various data delivery mechanisms and how they can be deployed in the overall DBGlobe framework.

We classified the various data delivery mechanisms based on the following four basic characteristics: (1) how is data delivery initiated (push vs. pull), (2) the frequency of the delivery (periodic vs. aperiodic), (3) the receivers of the data delivery (unicast vs. broadcast) and (4) the unit of interaction (data vs. query shipping). Various combinations along these four dimensions are possible. An interesting such combination is data diffusion in which an interest (query) for a data item is periodically injected (pulled) in the network and then broadcasted to the appropriate nodes for evaluation.

A central issue in data delivery is ensuring that the clients receive consistent data. We developed a formal framework for expressing the correctness properties of data delivery and present protocols that ensure both the temporal and the semantic coherency of data for the various data delivery mechanisms. By temporal coherency, we refer to the currency related properties of the data sets read by a client, e.g., when were the data items read by a client current at the server? Semantic coherency refers to transaction-based consistency properties, e.g., do the data items read by a client correspond to a "serializable" execution of the server transactions? Without both semantic and temporal coherency, clients may be seeing either outdated data sets or data sets that correspond to inconsistent database states.

We showed how such mechanisms fit into the DBGlobe architecture as defined in DBGlobe's "Deliverable D3: Overall System Architecture" and present directions regarding an adaptive deployment of each mechanism.
.

6.4 Information Discovery and Querying (WP4)

Introduction

The basic objective of WP4 is to derive methods and techniques for querying and discovering knowledge in environments consisting of autonomous, dynamic, and mobile entities. The first fundamental challenge is choosing an appropriate data model to describe the information available to, and offered by, these dynamic entities, and design a query language and a query execution paradigm for this context. This is part of tasks 4.1 and 4.2 in which we have made considerable progress.

Description of Work

As often when dealing with heterogeneous and autonomous information sources, our goal is to hide the heterogeneity and the data distribution details from the naive users, and enable them to manipulate and query the data as if it resides in a single source. The solution developed here is based on the following two main observations. XML, and emerging standards for web services, like SOAP and WSDL, normalize the way that distributed remote resources can be accessed, and more generally the way programs can be invoked over the web. De-centralized architectures, in which resources are shared by direct exchange between systems, propose a scalable solution for dynamic distributed environment with independent data sources. These two paradigms, combined together, form the basis for our scalable data integration platform. We propose Active XML (AXML in short), a language that leverages services for data integration and is put to work in a global computing architecture. The AXML framework is centered on AXML documents. These are XML documents that may contain calls to services. When calls included in an AXML document are fired, the document is enriched by the corresponding results. An AXML PMO or DBGlobe system component contains AXML documents and offers (AXML) services, which can be used to enrich the AXML documents of the same or of other components. In some sense, an AXML document can be seen as a (partially) materialized view integrating plain XML data and dynamic data obtained from service calls. As a simple example, consider an AXML document for the home-page of a local newspaper. It may contain some plain XML data, such as general information about the newspaper, and some dynamic fragments, e.g. one for the current temperature in the city, obtained from a weather forecast Web service, and a listing of current art exhibits, obtained from the local TimeOut guide service. While documents with embedded calls have been used before, to our knowledge, AXML is the first to actually turns calls to web services embedded in XML documents into a powerful tool for data integration.

Controlling Service Call Activation. The service calls inside AXML documents are represented by special XML elements carrying the tag <sc> (for service call) which are interpreted as calls to services. An <sc> element encodes the URL of the site providing the service, the service name, the particular functionality of the service that is invoked, and the appropriate parameters for the call. Particular attributes of the element allow to specify when a service call should be activated (e.g. when needed, every hour, etc.), and for how long its result should be considered valid. This simple mechanism allows capturing and combining different styles of data integration such as warehousing and mediation.

Continuous services. Simple services are similar to remote procedure calls. The service is called with some arguments, and eventually returns an answer. For the interesting class of continuous services, the interaction is more complex. Once a service call has been registered, a stream of answers is returned for this single service call. Streams of answers are encountered in many real-life applications: the stream of source updates for the maintenance of a data warehouse, the readings of a temperature sensor or surveillance system, answers returned by continuous queries or publish-subscribe systems. In the case of such continuous services, the activation of the service call encapsulates a service subscription, and all the results received subsequently from the service are integrated in the caller document.

Intensional Parameters and Results. The parameters and result of a service call may themselves be AXML documents. A system component receiving a call with parameter containing service calls may have to activate the calls it includes before actually performing the service. Similarly, a service result may contain further service calls, which have to be activated by the site receiving the result. Thus, service call activations entail exchanging intensional data, and lead to a form of distributed computation. The use of intensional parameters/results involves security issues. For instance, a malicious user may force a site to perform a dangerous action by calling one of the site's services with an intensional parameter that contains a call to a dangerous service. Besides security, the peer capabilities need also to be taken into consideration. For instance, if the receiving site is a non-AXML client (hence unable to activate the service calls embedded in the data) only fully extensional data must be sent. To address these issues we introduced a typing mechanism to control the execution of services. Just like for regular XML data, AXML schemas (ala XML Schema) are used to control the exchange of AXML data and, in particular, to determine which services should be called before sending an AXML document, and which should not. We formalized the problem and developed algorithms to solve it.

Defining Services. The AXML framework allows to use arbitrary existing services as well as to define new services on top of the enriched AXML documents. The definition of AXML services relies on parameterized XML queries expressed in XQuery, the standard language for querying XML data, extended with updates. AXML services may query and update AXML or regular XML documents. The AXML service specification allows, in particular, the definition of continuous services, and of services with intensional input/output. One should note that an AXML service, when called by a non-AXML client, will detect the limitations of its caller and in such case will return only extensional answers.

Distribution and Replication. By the sole presence of PMO- and DBGlobe- services, AXML documents already include inherently some form of distributed computation. A higher level of distribution that also allows (fragments of) AXML documents and services to be distributed and/or replicated over several sites is highly desirable in a dynamic mobile architecture: A single XML document may need to be distributed on more than one PMO if the PMO does not have enough storage; Replication of services, as well as the data they utilize, is needed, since a PMO will tend to use services "nearby". To address this, we extended the AXML data model with distribution and replication, and provided a location-aware extension to XQuery to handle distributed/replicated data and services. We developed a comprehensive cost model for query evaluation that applies to both user queries and service class. Finally, we designed an algorithm that (based on the above cost model) chooses data and services that DBGlobe should replicate to improve the efficiency of maintaining and querying its dynamic documents.

On the industrial side, it is quite clear that information management for heterogeneous mobile entities is in great demand. The technology that will possibly emerge from this research has thus the potential for a fast industrial implementation, given the increasing need for such support on the one hand and the fact we base our research on industrial standards such as XML, SOAP, and WSDL. We intend to get relevant feedback from industry by organizing meetings with such relevant parties, presenting and demonstrating the developed technology when the implemented prototype will be available.

6.5 Proof–of-Concept Prototype (WP5)

The aim of WP5 is to develop a proof of content prototype that verifies the various concepts developed within the DBGlobe project. In summary,

- To demonstrate through a concrete example the idea of ad-hoc databases and the various forms of querying
- To provide a concrete example of the workflow interaction model and the different data delivery mechanisms
- To capture, design and implement a location-aware system

We have further improved the overall design of our application by further enhancing and understanding the needs of the DBGlobe system and architecture. In doing we have focused on the needs of the moving user or entity.

6.6 Project Management and Dissemination of Results (WP6)

This work package is responsible for the co-ordination of the project in both administrative and technical terms aiming towards achieving effective operation of the project, as well as timely delivery of the results.

We have achieved:

- all project objectives within the work program time and resource constraints,
- deliverables with “publishable” research quality, as suggested by the publications resulted from the project

- the compliance with the reporting requirements outlined in the Contractual Agreement entered into the European Community,
- a number of meetings and the formation of working groups
- the wide distribution of the research results of the project by publishing at competitive journals and conferences and creating a web page for the project,
- the attendance of maximum visibility for the project in the research (and industry) community by presenting our work in many scientific and industrial forums,
- to disseminate the simulator (we have set up a page for distributing the simulator through the web, also we have submitted demos of the simulator to competitive forums),
- the continuous observation and assessment of areas of impact which provided our project with invaluable feedback.
- describe the activities related to the integration of DBGlobe project with the other project of the Global Computing project (both with projects inside the "Foundation of Networks and Large Distributed Systems" cluster and with related projects at other clusters).

6.7 Project's Achievement Fiche

Questions about project's outcomes	Number	Comments
1. Scientific and technological achievements of the project (and why are they so ?)		
Question 1.1. Which is the 'Breakthrough' or 'real' innovation achieved in the considered period	N/A	**Architecture** Much work has been invested on issues relating to caching and replication of documents at the large-scale fixed infrastructure (ie the Internet) assumed within the DBGlobe architecture. The real innovation of the group's work consists of: 1. A tool, coined ProxyTeller, that helps the system designers to make beneficial decisions about where to place replicas and caches of data items within the network infrastructure and whether it is of benefit and according to which metric to cache or replicate. The tool is available through the Internet to be used by the members of the community. Such a tool was very much lacking. 2. Assuming an event-based environment, as is the case for a very large class of existing and emerging applications, we concentrated on pub/sub systems. We offered a new paradigm (architecture, data structures, and routing and matching algorithms) for processing publications (events) and subscriptions in the system, that is, based on summarizations of subscriptions, which considerably improves the overall system performance. The above key results have resulted in 4 Conference publications and one journal submission. All these conferences are considered to be prestigious and very competitive with acceptance rates ranging from 17% to 25%. In the third year, we have put special emphasis on distributed architectures that are based on "content-based" clustering. In particular, we consider hierarchical clustering of information in global computing based on the unstructured super-peer architectures. Another innovation of our research is the introduction of the notion of "workload-aware" clustering that takes into consideration the type and frequency of queries in the construction of clusters.

Context Management

Research work in the DBGlobe project has focused on the various types of context information available in a global computing environment putting special emphasis on location. In the *third* and final year of the project, we concluded our work in the area and focused on context management within new paradigms of structuring distributed systems such as in peer-to-peer architectures and web service computing

Our approach to the basic problem of information discovery in a global computing environment is to use a distinguished context of the data, namely *space*, as a means to index information and querying. This idea was explored in the specific context of *location-based services* by means of an ontology characterizing the specific scenario.

Within the field of global computing initiative, mobile service discovery plays an eminent role. The contribution of this effort lies in improving the precision and recall of mobile service discovery, by introducing the use of implicit contextual information related both to the user and the web service. This is achieved by a novel proposal for a context-aware service directory that provides searching facilities for web services based on contextual parameters. Our approach is applicable on a peer-to-peer network of service directories that serve as the underlying infrastructure for supporting mobile service interactions in a global computing environment.

Although the findings in this project are merely theoretical, the upcoming new product of the Greek startup company *Talent SA* may prove the practical relevance of those concepts. The product, Cruiser – a Web portal specialized in the exchange and publishing of spatially-referenced information, is based on the concept of exploring information by navigating based on its spatial aspect.

Location-Aware Prototype
In terms of our prototype, during the third year we performed experiments to evaluate its salability. In addition, we experimented with small sensor devices. Innovation regarding our prototype includes:

1. We extend the notion of Location-aware querying over the space of concept based querying utilizing the notion of “communities” and provide rigorous performance evaluation demonstrating improved performance.
2. A real life demo substantiating and demonstrating the viability of the above concepts incorporating

<table>
<tr><td></td><td></td><td>Active XML, multilevel Bloom filters. This infrastructure is been used as a testbed of other innovative ideas, e.g., moving communities
3. An application environment to develop concept based applications

Querying
A main contribution of the project is the development of Active XML (AXML, for short), a declarative framework that harnesses Web services for data integration and is put to work in a global computing architecture.

This novel technology allows combining of stored information with data defined in an intentional manner as well as dynamic information. Thus, it offers a sound solution to global data management in the dynamic global computing context.

AXML is now in open source in the ObjectWeb industrial consortium:
http://forge.objectweb.org/projects/activexml/</td></tr>
<tr><td colspan="3">2. Impact on Science and Technology: Scientific Publications in scientific magazines</td></tr>
<tr><td>Question 2.1.

Scientific or technical publications on reviewed journals and conferences</td><td>71</td><td>Title and journals/conference and partners involved

D. Pfoser and C. S. Jensen. Trajectory Indexing Using Movement Constraints. GeoInformatica Journal, to appear, June, 2005.
Partner: CTI and Univ. of Aalborg

N. Tryfona and D. Pfoser. Data Semantics in Location-Based Services. Journal on Data Semantics, to appear, April 2005
Partner: CTI

S. Brakatsoulas, D.Pfoser, and Nectaria Tryfona. Practical Data Management Techniques for Vehicle Tracking Data. 21st International Conference on Data Engineering (ICDE), to appear, 2005
Partner: CTI</td></tr>
</table>

		S. Brakatsoulas, D. Pfoser, and N. Tryfona. Modelling Storing and Mining Moving Object Databases. *8th International Database Engineering & Applications Symposium (IDEAS)*, 2004 Partner: CTI D. Pfoser, N. Tryfona and V. Verykios, "Services-Based Data Management in a Global Computing Environment", *Third International Workshop on Web and Wireless Geographical Information Systems (W2GIS 2003)* Partner: CTI L. Speicys, C. S. Jensen, A Kligys, "Computational Data Modeling for Network-Constrained Moving Objects", in proceedings of the *11th ACM International Symposium on Advances in Geographical Information Systems* (ACM-GIS), 2003. Partner: Aalborg D. Pfoser, and C. S. Jensen, "Indexing of Network-Constrained Moving Objects", in proceedings of the *11th ACM International Symposium on Advances in Geographical Information Systems (*ACM-GIS), 2003. Partner: CTI and Aalborg M. A. Nascimento, D. Pfoser, and Y. Theodoridis, *"Synthetic and Real Spatiotemporal Datasets",* Invited contribution DEBulletin, June 2003. Partner: CTI D. Pfoser, E. Pitoura, and N. Tryfona. Metadata Modeling in a Global Computing Environment. Proceedings *of the 10th ACM International Symposium on Advances in Geographic Information Systems,* McLean, VA November 8-9, 2002. Partner: CTI & UoI C. Doulkeridis, V. Zafeiris and M. Vazirgiannis. "The Role of Caching and Context-Awareness in P2P Service Discovery", to appear in the *6th International Conference on Mobile Data Management (MDM'05)*, Ayia Napa, Cyprus, May 9-13, 2005. Partner: AUEB C. Doulkeridis and M. Vazirgiannis. "Querying and Updating a Context-Aware Service Directory in Mobile Environments", proceedings of the *2004 IEEE/WIC/ACM International Conference on Web Intelligence (WI'04)*, Beijing, China, pages 562-565, September 20-24, 2004.

		Partner: AUEB C. Doulkeridis and M. Vazirgiannis. "Updating a Context-Aware Service Directory for M-Services", in the *3rd Hellenic Data Management Symposium, HDMS'04* , Athens, Greece, June 28-29, 2004. Partner: AUEB E. Valavanis, C. Ververidis, M. Vazirgiannis, G.C. Polyzos, K. Nørvåg, "MobiShare: Sharing Context-Dependent Data and Services from Mobile Sources", Procs of the *2003 IEEE/WIC International Conference on Web Intelligence* WI 2003 October 13-17, 2003, Halifax, Canada Partner: AUEB C. Doulkeridis, E. Valavanis and M. Vazirgiannis. "Towards a Context-Aware Service Directory". Proceedings of the *4th VLDB Workshop on Technologies for E-Services* (TES'03), Sept.8 , 2003, Berlin, Germany. Proceedings published in the Springer Lecture Notes in Computer Science (LNCS) series. Partner: AUEB C. Ververidis, S. Valavanis, M. Vazirgiannis, G.C. Polyzos, "An Architecture for Sharing, Discovering and Accessing Mobile Data and Services: Location and Mobility Issues", Presented at: *Lobster Workshop*, LBS for accelerating the European-wide deployment of Services for the Mobile User and worker, Mykonos, Greece, 4-5 October, 2002, http://www.iit.demokritos.gr/lobster/lobster_mykonos2002.html Partner: AUEB A. S. Andreou, S. Mavromoustakos, C. Leonidou ,C. Chrysostomou, A. Pitsillides, G. Samaras, C. Schizas, "Key Issues for the Design and Development of Mobile Commerce Services and Applications", *International Journal of Mobile Communications (IJMC),* to appear. Partner: UCY S. Polyviou, G. Samaras and P. Evripidou, "A Relationally Complete Visual Query Language for Heterogeneous Data Sources and Pervasive Querying", Proc. of the *19th International Conference on Data Engineering*, Sponsored by the IEEE Computer Society, Tokyo, Japan, April 5-8, 2005. Partner: UCY C. Panayiotou and G. Samaras, "Personalized Portals for the Wireless User: An Agent Approach".

		Journal of ACM/Baltzer Mobile Networking and Applications (MONET), special issue on “Mobile Commerce", Volume 9, Issue 6, 663–677, 2004, Kluwer Academic Publishers. Partner: UCY O. Papapetrou and G. Samaras, “Optimizing Distributed Web Crawling by Minimizing the Network Distances”, Proc. of *the 9th IFCIS International Conference on Cooperative Information Systems (CoopIS´2004)*, Agia Napa, Cyprus, October 2004. Partner: UCY O. Papapetrou and G. Samaras, “IPMicra: Toward a distributed and adaptable location aware web crawler”, Proc. *8th East-European Conference on Advances in Databases and Information Systems (ADBIS 2004),* 22-25 September, 2004, Budapest, Hungary. Partner: UCY S. Polyviou, P. Evripidou and G. Samaras, “Query by Browsing: A Visual Query Language Based on the Relational Model and the Desktop User Interface Paradigm”, *The 3rd Hellenic Symposium on Data Management, (HDMS04)*, Athens, Greece, 28-29 June 2004. Partner: UCY S. Polyviou, P. Evripidou and G. Samaras, “PALLAS: A Querying Interface for Pervasive Computing Using Handheld Devices”, The *2004 ACS/IEEE International Conference on Pervasive Services (ICPS'2004)*, American University of Beirut, Lebanon, July 19-23, 2004. Partner: UCY O. Papapetrou and G. Samaras, “IPMicra: An IP-address Based Location Aware Distributed Web Crawler”, The *2004 International Conference on Internet Computing (IC'04)*: June 21-24, 2004, Las Vegas, Nevada, USA. Partner: UCY S. Polyviou, P. Evripidou and G. Samaras, “Context-Aware Queries using Query by Browsing and Chiromancer”, *2nd International Conference on Pervasive Computing*, Linz / Vienna, Austria, April 18-23, 2004. Publish in the "Advances in Pervasive Computing" book of the OCG (Vol 176 ISBN 3-85403-176-9).. Partner: UCY O. Papapetrou and G. Samaras, “IPMicra: A Distributed Location-Aware Crawler”, Poster

		Session, *13th International World Wide Web Conference (WWW2004),* 19-22 May 2004, New York, New York, USA Partner: UCY G. Samaras, "Mobile Agents: What about them? Did they deliver what they promised? Are they here to stay?" *Proc. of the 2004 IEEE International Conference on Mobile Data Management (MDM2004),* Berkeley, California, USA, January 19-22, 2004. Partner: UCY S. Papastavrou, G. Samaras, P. Evripidou and P. K. Chrysanthis, "Fine-Grained Parallelism in Dynamic Web Content Generation: The Parse & Dispatch Approach", *Proc. of the 9th IFCIS International Conference on Cooperative Information Systems (CoopIS´2003),* Italy, November 2003 Partner: UCY O. Papapetrou, S. Papastavrou, and G. Samaras, "Distributed Indexing of the Web Using Migrating Crawlers". *12th International World Wide Web Conference (W3)* 20-24 May 2003, Budapest, Hungary. (poster) Partner: UCY S. Papastavrou, G. Samaras, P. Evripidou and P. K. Chrysanthis, "Parse & Dispatch: Parallelizing the Generation of Dynamic Web Content", *12th International World Wide Web Conference* (W3) 20-24 May 2003, Budapest, Hungary. (poster) Partner: UCY C. Panayiotou, G. Samaras, "Personalized Portals for the Wireless User Based on Mobile Agents: Demonstration", *19th International Conference on Data Engineering*, IEEE Computer Society, March 5 - March 8, 2003 - Bangalore, India. (demo) Partner: UCY G. Samaras, C. Panayiotou, "A Flexible Personalization Architecture for Wireless Internet Based on Mobile Agents", Proc. *6th East-European Conference on Advances in Databases and Information Systems (ADBIS 2002),* September 2002, Bratislava, Slovakia. Partner: UCY C. Skouteli, G. Samaras and E. Pitoura, "Concept-Based Discovery of Mobile Services" Proc. of

		the *5th IEEE International Conference on Mobile Data Management (MDM2005*), Agia Napa, Cyprus, May 9-13, 2005. Partner: UCY and UoI K. Karenos, G. Samaras, E. Pitoura and P. K. Chrysanthis, “Mobile Agent-Based Services for View Materialization”. The *ACM Mobile Computing and Communications Review (MC2R),* special issue on “Recent Advances in Mobile Data Management Issues”, Volume 1, Number 2, December, 2004. Partner: UCY and UoI C. Spyrou, G. Samaras, E. Pitoura and E. Paraskevas “Mobile Agents for Wireless Computing: The Convergence of Wireless Computational Models with Mobile-Agent Technologies”, *Journal of ACM/Baltzer Mobile Networking and Applications (MONET),* special issue on “Mobility in Databases & Distributed Systems ", Volume 9, Issue 5 (Oct. 2004). Partner: UCY and UoI O. Shigiltchoff, P.K. Chrysanthis and E. Pitoura. “Adaptive Multiversion Data Broadcast Organizations” *Information Systems Journal*, Volume 29, Issue 6, Pages 509-528, September 2004 Partner: UCYand UoI O. Shigiltchoff, P. K. Chrysanthis and E. Pitoura. Energy Efficient Access in Multiversion Broadcast Environments. *5th IEEE International Conference on Mobile Data Management, Berkeley*, California, USA, January 19-22, 2004 (short paper) Partner: UCY and UoI G. Samaras, C. Skouteli, C. Panayiotou and E.Pitoura, “Communities: Concept-Based Querying of Mobile Services”, *International Workshop on Global Computing 2004 (GC 2004).* Published in the LNCS volume of the post-proceedings of the Global Computing 2004 workshop. Partner: UCY and UoI G. Samaras, K. Karenos, E. Pitoura, and P. K. Chrysanthis, “ViSMA: Extendible, Mobile-Agent Based Services for the Materialization and Maintenance of Personalized and Shareable Web Views”, Proc. of the *6th International Workshop “Mobility in Databases & Distributed Systems*” DEXA´03, September 1-5, 2003, Prague, Czech Republic. Partner: UCY and UoI

		G. Kastidou, E. Pitoura and G. Samaras, "A Scalable Mobile Agent Location Mechanism", Accepted for Publication *1st International Workshop on Mobile Distributed Computing* (MDC'03), May 19, 2003, held in conjunction with the 23rd International Conference on Distributed Computing Systems (ICDCS'03). Partner: UCY and UoI E. Pitoura and P. Chrysanthis. "Multiversion Data Broadcast"*, IEEE Transactions on Computers* 51(10):1224-1230, October, 2002. Partner: UCY and UoI G. Samaras, C. Spyrou, E. Pitoura, "View Generator (VG): A Mobile Agent Based System for the Creation and Maintenance of Web Views"*, 7th IEEE Symposium on Computers and Communications*, Taormina, Italy July 2002. Partner: UCY and UoI O. Shigiltchoff, P. Chrysanthis and E. Pitoura. "Multi-version Data Broadcast Organizations". In *Proceedings of the 6th East European Conference on Advances in Databases and Information Systems (ADBIS),* September 2002, Bratislava, Slovakia. Partner: UCY and UoI S. Abiteboul, O. Benjelloun, B. Cautis, I. Manolescu, T. Milo and N. Preda. "Lazy Query Evaluation for Active XML". *Proceedings of the 2004 ACM SIGMOD international conference on Management of Data*, *SIGMOD 2004*. pp. 227-238, June 2004, Maison de la Chimie, Paris, France. Partner: INRIA S. Abiteboul. "Distributed Information Management with XML and Web Services". *European Joint Conferences on Theory and Practice of Software (ETAPS)* March 2004, Barcelona, invited paper. Proceedings of FASE' 2004 published by Springer-Verlag in the Lecture Notes on Computer Science, Volume 2984/2004, pp 1-11, April 2004. Partner: INRIA T. Milo, S. Abiteboul, B. Amann, O. Benjelloun, F. D. Ngoc, "Exchanging Intentional XML Data", Procs of the 2003 *ACM SIGMOD International Conference on Management of Data*, pp 289-300, San Diego, California, USA, June 9-12, 2003.

		Partner: INRIA S. Abiteboul, A. Bonifati, G. Cobena, I. Manolescu and T. Milo, "Dynamic XML Documents with Distribution and Replication", Procs of the *2003 ACM SIGMOD International Conference on Management of Data*, pp 527-538, San Diego, California, USA, June 9-12, 2003. Partner: INRIA S. Abiteboul, J. Baumgarten, A. Bonifati, G. Cobéna, C. Cremarenco, F. Dragan, I. Manolescu, T. Milo, N. Preda, "Managing Distributed Workspaces with Active XML", *Demo, Procs. of VLDB'03*, Sept 2003 (demo) Partner: INRIA S. Abiteboul, B. Amann, J. Baumgarten, o. Benjelloun, F. D. Ngoc, and T, Milo, "Schema-Driven Customization of Web Services", *Demo, Procs. of VLDB'03*, Sept 2003 (demo) Partner: INRIA The Active XML team. Active XML primer, INRIA report, (Gemo-Report-275) April 2003 Partner: INRIA S. Abiteboul, O. Benjelloun, I. Manolescu, T. Milo and R. Weber, "Active XML: Peer-to- Peer Data and Web Services Integration (demo)", *Proceedings of the 28th VLDB Conference*, Hong Kong, 2002. Partner: INRIA K. Stefanidis, E. Pitoura and P. Vassilaidis On Supporint Context-Aware Preferences in Relational Database Systems In proceedings of the MDM 2005 Workshop on "*Managing Context Information in Mobile and Pervasive Environments*" (MCMP05), Ayia Napa. Cyprus, May 9 2005 Partner: UoI O. Shigiltchoff, P. K. Chrysanthis and E. Pitoura. "Adaptive Multiversion Data Broadcast Organizations". *Information Systems Journal*. 29(6): 509-528, 2004. Partner: UoI G. Koloniari and E. Pitoura, Filters for XML-based Service Discovery in Pervasive Computing. *Computer Journal*, Special Issue on Mobile and Pervasive Computing, Volume 47, No 4, July 2004, pp 461-474

		Partner: UoI G. Koloniari, E. Pitoura. "Content-based Routing of Path Queries in Peer-to-Peer Systems". *EDBT 2004*. Partner: UoI I. Kyriazis and I. Fudos, "Thin Client Access to a Visualization Environment", *Third International Workshop on Graphics and Geometric Modelling* CGGM04, in conjunction with ICCS 2004. Partner: UoI E. Leontiadis, V. V. Dimakopoulos and E. Pitoura, "Cache Updates in a Peer-to-Peer Network of Mobile Agents", *4th IEEE International Conference on Peer-to-Peer Computing* (P2P 2004), Zurich, Switzerland, August 2004 Partner: UoI Y. Petrakis, G. Koloniari and E. Pitoura, "On Using Histograms as Routing Indexes in Peer-to-Peer System", *2nd International Workshop on Databases, Information Systems and Peer-to-Peer Computing*, In conjunction with VLDB2004, August 29-30, Toronto, Canada Partner: UoI Y. Petrakis and E. Pitoura, "On Constructing Small Worlds in Unstructured Peer-to-Peer Systems", In the EDBT International Workshop on Peer-to-Peer Computing and Databases, Heraklion, Crete, Greece, March 14, 2004 Partner: UoI O. Shigiltchoff, P. K. Chrysanthis, E. Pitoura, "Energy Efficient Access in Multiversion Broadcast Environment". *Mobile Data Management 2004*, page 168 (short paper) Partner: UoI A. Zarras, P. Vassiliadis and V. Issarny. Model-Driven Dependability Analysis of Web Services . *Procs of the 6th International Symposium on Distributed Objects and Applications (DOA'04),* 2004. Partner: UoI V. V. Dimakopoulos and E. Pitoura. "A Peer-to-Peer Approach to Resource Discovery in Multi-

		Agent Systems", *7th International Workshop Cooperative Information Agents*, CIA 2003, Helsinki, Finland, August 27-29, 2003, pp 62-77 Partner: UoI G. Koloniari and E. Pitoura, "Bloom-based Filters for Hierarchical Data", *5th Workshop on Distributed Data and Structures* (WDAS), June 2003- extended version to appear in the workshop post-proceeding published by Carleton Scientific. Partner: UoI G. Koloniari, Y. Petrakis and E. Pitoura. "Content-Based Overlay Networks of XML Peers Based on Multi-Level Bloom Filters". *International VLDB Workshop on Databases, Information Systems and Peer-to-Peer Computing*, 2003 Partner: UoI E. Pitoura, P. K. Chrysanthis and K. Ramamritham. "Characterizing the Temporal and Semantic Coherency of Broadcast-based Data Dissemination". Proc. of the *International Conference on Database Theory*, January 2003, Siena, Italy. Partner: UoI A. Karakasidis and E. Pitoura, "DBGlobe: A Data-Centric Approach to Global Computing". *International Workshop on Smart Appliances and Wearable Computing (IWSAWC 2002*) In conjunction with ICDCS 2002, Vienna, Austria, July 2002 Partner: UoI Th. Korakis and L. Tassioulas. "An Approach Towards QoS Provisioning in Wireless LAN Complying to the 802.11e Framework". *IEEE Lanman 2002 Workshop*, August 2002, Stockholm, Sweden. Partner: UoI P. Triantafillou, and A. Economides, "Subscription Summarization: A New Paradigm for Efficient Publish/Subscribe Systems", *IEEE International Conference on Distributed Computing Systems*, ICDCS, 2004 Partner: TUC P. Triantafillou and I. Aekaterinides ProxyTeller: "A Proxy Placement Tool for Content Delivery under Performance Constraints", *4th Web Information Systems Engineering Conference (IEEE)*,

		Dec 2003. Partner: TUC P. Triantafillou and A. Economides, "Subscription Summaries for Scalability and efficiency in Publish/Subscribe Systems", *1st Intl. IEEE Workshop on Distributed Event-based Systems*, (DEBS02) July 2002. Partner: TUC P. Triantafillou and I. Aekaterinides, "Web Proxy Cache Placement, Replacement and the Proxy Teller", *2nd International IEEE Symposium on Network Computing and Applications* April, 2003. Partner: TUC P. Triantafillou, I. Aekaterinides, "ProxyTeller: A Tool Guiding Web Proxy Placement Decisions", *12th International World Wide Web Conference (WWW)* May, 2003, (poster). Partner: TUC
Question 2.2. Scientific or technical publications on non-reviewed journals and conferences	4	We list below submitted publications D. Pfoser and N. Tryfona, "The Use of Ontologies in Location-Based Services: The Space and Time Ontology in Protégé-2000", under submission. Partner: CTI P. Triantafillou and I. Aekaterinides, "Guiding Web Proxy and Server Placement for High Performance Internet Content Delivery" Partner: TUC G. Samaras, C. Skouteli, C. Panayiotou, E. Pitoura, "Communities: Creating and Quering Ad-hoc Databases based on Concepts", submitted Partner: UCY and UoI George Samaras, Maria Antreou, Christoforos Panayiotou, Andreas Pitsillides, "Time Based Personalization for the Moving User", submitted Partner: UCY

Question 2.3. Invited papers published in scientific or technical journal or conference.	9	Evaggelia Pitoura, George Samaras, Can Turker, "Seamless Consistency", Book Chapter in "Mobile Middleware", edited by Paolo Bellavista, Antonio Corradi, to appear 2005. Partner: UCY and UoI Evaggelia Pitoura (UoI) was (with P. K. Chrysanthis) the special issue editors for the topic "Data Engineering of Mobie and Wireless Access" for the Wireless Network Journal, Vol 10, Issue 6, 2004 George Samaras (UCY) was a panelist organizer and moderator: "Mobile Agents: What about them? Did they deliver what they promised? Are they here to stay?" main panel of the 2004 IEEE International Conference on Mobile Data Management (MDM2004), Berkeley, California, USA, January 19-22, 2004. Partner: UCY Serge Abiteboul: Distributed information management with XML and Web Services. European Joint Conferences on Theory and Practice of Software (ETAPS), March 2004, Barcelona. Invited talk. In proceedings of FASE'2004 Partner: INRIA Evaggelia Pitoura, George Samaras, Georgia Kastidou, "Locating Mobile Objects", Book Chapter, "Handbook of Mobile Computing", (Imad Mahgoub, Editor), CRC Press, December 2004. Partner: UCY and UoI E. Pitoura, S. Abiteboul, D. Pfoser, G. Samaras and M. Vazirgiannis, "DBGlobe: A Service-Oriented P2P System for Global Computing", ***Invited contribution***, *Sigmod Record* 32(3), pp.77-82, September 2003 Partners: All M. A. Nascimento, D. Pfoser, and Y. Theodoridis, "Synthetic and Real Spatiotemporal Datasets", **Invited contribution,** *IEEE Data Engineering Bulletin*, June 2003. Partner: CTI

		5th Mobile Data Management (MDM) G. Samaras invited as **Panel** chair in the 5th *Mobile Data Management* (MDM) conference, Berkley, California, January 19-22. Partner: UCY Serge Abiteboul: Managing an XML Warehouse in a P2P Context. *The 15th International Conference on the Advanced Information Systems Engineering*, (CAiSE 2003), Klagenfurt, Austria, June 16-18, 2003. **Keynote talk**. Partner: INRIA
3. Impact on Innovation and Micro-economy		
A – Patents		
Question 3.1. Patents filed and pending		
Question 3.2. Patents awarded		
Question 3.3. Patents sold		
Questions about project's outcomes	**Number**	**Comments or suggestions for further investigation**
B - Start-ups		
Question 3.4.	No	

Creation of start-up		
Question 3.5. Creation of new department of research (ie: organisational change)	No	
C – Technology transfer of project's results		
Question 3.6. Collaboration/ partnership with a company ?		Which partner : CTI Which partner : CTI Which company: Talent S.A., Greece What kind of collaboration? The product of Talent is a Web portal specialized in the exchange and publishing of spatially-referenced information. CTI expertise on spatiotemporal data modelling and data management will be incorporated in this product. Which partner : CTI Which company : Emphasis Telematics Solutions (Emphasis) What kind of collaboration ? CTI is working with Emphasis on the extension of their existing fleet-management solutions software. The

		contribution of CTI is the modeling of the data in this system and the definition of add-valued services such as location-based services (LBS) in the fleet management context. For the modeling of the data and as a basis of the location, we use an ontology modeling space and time as defined in the context of the work of CTI within the DBGlobe project. This specific ontology will be used as the basic means to model LBS data and to exchange information (common data model). Which partner : AUEB Which company: ERICSSON RESEARCH What kind of collaboration? Consulting in the context of MARIE CURIE Grants in the area of text mining for P2P computing. Which partner : INRIA Which companies : The French National Institute of Agronomy, ALCATEL, Mandrake software What kind of collaboration? AXML is used in a project with the French National Institute of Agronomy for the construction of a warehouse on food risk. Discussions for possible transfer of AXML are conducted with ACATEL for network configuration and with Mandrake Software for software distribution.
4. Other effects		
A - Participation to Conferences/Symposium/Workshops or other dissemination events		
Question 4.1.		

Active participation[1] to Conferences in EU Member states, Candidate countries / NAS. (specify if one partner or "collaborative" between partners)		Names/ Dates/ Subject area / Country: At least one member of the group (most often 2 or 3) participated in the conference where the publications listed in Question 2.1 and 2. 3 were presented In addition: Presentation C. Doulkeridis (AUEB) as 2nd winner of the "Michael Dertouzos Competition" on IT with a Human Face, in the *14th World Congress on Information Technology* (WCIT'04), Athens, Greece, May 2004. George Samaras (UCY) was invited as General chair of the *6th Mobile Data Management (MDM) conference*. The conference will be held in Cyprus, Agia Napa, 9-13 May, 2005. First time in Europe George Samaras (UCY) has co-organized a special track on "Ubiquitous Computing" at the Symposium on Applied Computing (SAC), Nicosia, Cyprus, March 14-17, 2004 Evaggelia Pitoura (UoI) is the global chair for the topic "Mobile and Ubiquitous Computing" of *EuroPar 2005* Evaggelia Pitoura (UoI) was one of the organizers of the *Dagstuhl seminar* on "Mobile Information Systems", 25-29 October 2004. She also gave an invited overview talk on global computing. Evaggelia Pitoura (UoI) was program chair of the 3rd Hellenic Database Conference. A special session on global computing was organized. One invited talks by S. Abiteboul (INRIA) at the European Joint Conferences on Theory and Practice of Software (ETAPS), in Barcelona, Spain) Also, there were two demos presented by members of the INRIA group (O. Benjelloun and T. Milo) at the VLDB 2003 conference, Berlin, Germany, 9-12 Sept. 2003. Two invited talks by S. Abiteboul (INRIA): the keynote address at 15th International Conference on the

[1] 'Active Participation' in the means of organising a workshop / session / stand / exhibition directly related to the project (apart from events presented in section 2).

		Advanced Information Systems Engineering, (CAiSE 2003), Klagenfurt, Austria and one scheduled for March 2004 at the European Joint Conferences on Theory and Practice of Software (ETAPS), in Barcelona, Spain) Presentation by UoI (E. Pitoura and I. Fudos) of the project at the 4th Agentcities information day (iD4) that was held in Helsinki, Finland on the 25th and 26th of August 2003
Question 4.2. Active participation to Conferences outside the above countries (specify if one partner or "collaborative" between partners)		Names/ Dates/ Subject area / Country: At least one member of the group (most often 2 or 3) participated in the conference where the publications listed in Question 2.1 and 2. 3 were presented In addition, DBGlobe members participated in the organization of the following conferences: Dieter Pfoser (CTI) was program committee chai, of the *12th Symposium on the Advances of Geographical Information Systems*, Washington DC, USA. Evaggelia Pitoura (UoI) was the *ICDE 2005 Program Vice Chair* for "Distributed, Parallel, Deep Web, and P2P Databases" George. Samaras (UCY) organized a panel in the *5th Mobile Data Management (MDM) Conference*, Berkley, California, January 19-22. Also G. Samaras (UCY) organized a panel in the 5th *Mobile Data Management* (MDM) conference, Berkley, California, January 19-22. One demo presented by C. Panayiotou (UCY) at the *19th* International Conference on Data Engineering, IEEE Computer Society, March 5 - March 8, 2003 - Bangalore, India
B – Training effect		
Question 4.3. Number of PhD students		In what field? AUEB Three Phd Students

hired for project's completion		C. Doulkeridis: Content Managements and P2P Semantic Overlay Network Generation E. Valavanis: Mobile Computing Architectures C. Ververidis: Ad hoc Networks INRIA Omar Benjelloun, PhD thesis on Active XML, thesis completed in 2004 Benjamin Nguyen, partly involved in the AXML project. New thesis on security and AXML started (Bogdan Cautis) UoI Georgia Koloniari and Yannis Petrakis on service discovery and querying in global computing and peer-to-peer systems also a number of MSc students: Thodoris Tsotos (on using histograms for resource discovery in global computing) Zisis Plitis (on integrating sensor devices in the DBGlobe prototype) Kostas Stefanidis (on context-aware preferences) Kostas Stamkopulos (on peer-to-peer models for web service discovery) UCY Three PhD students: Chara Skouteli and Christoforos Panayiotou in the area of Context awareness and Personalization. Stavros Polyviou on Pervasive Querying Also three graduate students as follows: Maria Andreou and George Ioakim on personalization and Odysseas Papapetrou on Location-aware Web Crawling.
Questions about project's outcomes	**Number**	**Comments or suggestions for further investigation**
C - Public Visibility		
Question 4.4. Media appearances		References: Press release: C.Doulkeridis 2nd winner of the Michael Dertouzos Awards in the 14th World Congress on Information Technology (WCIT'04), Athens, Greece, May 22, 2004.

and general publications (articles, press releases, etc.)	2	http://www.wcit2004.com/media/docs/tessera.pdf Press release: A half-page entry on the DBGlobe project appeared in the October 18th issue (issue No 190) of the Parliament Magazine
Question 4.5. Web-pages created or other web-site links related to the project	7	**Project web page** at http://softsys.cs.uoi.gr/dbglobe/ Local web pages of the project is hosted by AUEB at http://www.db-net.aueb.gr/dbglobe/ http://proxyteller.ceid.upatras.gr is the site where the *ProxyTeller* tool may be ran. Web page about the *simulator*: http://www.dbnet.aueb.gr/dbglobe/NetworkSimulation/DBGLOBE_NCTUNS_simulation_slides2_files/frame.htm Web page about the *testbed* http://www.db-net.aueb.gr/dbglobe/Testbed/ Testbed_files/frame.htm Web page of the *Proof of concept prototype* link http://cs458.cs.ucy.ac.cy:8080/dbglobe/jsp/login.jsp Web page of the Active XML http://www-rocq.inria.fr/gemo/Gemo/Projects/axml/
Question 4.6. Video		References: (Please attach relevant material)

produced or other dissemination material		
Question 4.7. Key pictures of results		References: (Please attach relevant material .jpeg or .gif)
D - Spill-over effects		
Question 4.8. Any spill-over to national programs	Yes	• Research program PYTHAGORAS II (supported by the Greek Ministry of National Education and Religious Affairs): ΔΙΑΧΕΩ on Database Management over Ad-Hoc Networks (UoI) • Bilateral cooperation Greece – Germany funded by the Greek General Secretariat of Research and Technology (GSRT) on traffic sensor data integration. (CTI) • “MHTIS: Pervasive Query language based on the relational model”, PENEK program aiming to support young scientist.(UCY) • A new project KadoP on combining AXML with distributed hash tables (DHTs) (INRIA) • AXML possibly be used in ACI Mase de Donnes MDP2P project on peer to peer data management. (INRIA) • Research program PYTHAGORAS (supported by the Greek Ministry of National Education and Religious Affairs): On extended pub/sub models for smart-space environments (TUC) • Research program PYTHAGORAS (supported by the Greek Ministry of National Education and Religious Affairs): Overlay Networks of Context-Aware Web Services: Management, Adaptability and Query Processing (UoI) • ENTER program of the Greek General Secretariat of Research and Technology (GSRT). Project

		ΙΧΝΗΛΑΤΗΣ on the development of services for traffic management systems.(CTI) • "Persona: Personalized Portals for the wireless user", PENEK program aiming to support young scientist.(UCY) • A bilateral project (Cyprus - Greece) funded by the Greek General Secretariat of Research and Technology (GSRT) and the Cyprus Research Promotion Foundation called "SemaNet: Semantic Overlay Networks of Services in Ubiquitous Computing". (UCY and UoI) • A bilateral project (Greece - Australia) funded by the Greek General Secretariat of Research and Technology (GSRT) on Data Engineering for Context-Aware Applications (UoI) • Active AXML is used in RNTL project E.DOT on food risk (INRIA) • Bilateral cooperation Greece – France funded by the Greek General Secretariat of Research and Technology (GSRT) MOBWS : Context Aware Web Services for Nomadic E-Business Systems, with INRIA-Rocquencourt - ARLES Group
<u>Question 4.9.</u> Any spill-over to another part of EU IST Programme	Yes	• MARIE CURIE Intra-European Fellowship - NGWeMiS - Next Generation Web Mining & Searching The project lies in the area of knowledge extraction and management from the massive and heterogeneous document collections on the World Wide Web. The main objective of the proposed project is the design guidelines and prototypes development for next generation web mining and searching techniques based on the P2P paradigm. of Web data. The innovation lies in i. usage of P2P paradigm in the various levels of web content management and searching, ii. the study and development of novel similarity measures among web pages that take into account multiple facets of web pages iii. clustering the web data and meta data taking into account their P2P organization paradigm. (AUEB and INRIA) • EDOS: Environent of the distribution of Open Source Software New european project (IST6) on the management of open source that may use ActiveXML for distribution. http://www.pps.jussieu.fr/~dicosmo/EDOS (INRIA) • Program FP6, IST/FET IP DELIS: On Peer-to-Peer Data Management (caching, replication, load

		balancing, etc.) • AEOLUS: project of the Global Computing II Initiative. The goal of this project is to investigate the principles and develop the algorithmic methods for building such an overlay computer that enables this efficient and transparent access to the resources of an Internet-based global computer.
Question 4.10. Are other team(s) involved in the same type of research as the one in your project ?	Yes	One focus in our work is to utilize space and time as a means to facilitate the structuring and querying of data in a distributed setting. This work is based on approaches that exist in the context of spatial and spatiotemporal databases. Specifically work on the handling of moving objects exists from several teams around the world. Examples include the University of California at Riversides (V. Tsotras) and Boston University (G. Kollios) on indexing moving objects, the Hong Kong University of Science and Technology (D. Papadias) on performance aspects in spatiotemporal databases and exploiting networks as constraining movement environments, and the University of Illinois at Chicago (O. Wolfson) on the management of moving objects in databases including aspects of uncertainty. Indexing distributed spatial information was investigated by the University of Thessaloniki (Y. Manolopoulos). We utilize specific ontologies to structure and exchange data. Related research to our work on defining an ontology for space and time includes work by Fraunhofer ISST (A. Voisard and N. Weissman) on the modeling of profile data using event notification techniques and on a modular ontology architecture to support ontologies and standards for Web services specifically for Olympia 2008, EPFL Lausanne (S. Spaccapietra) on the modelling of profile data using relational databases and work on scenarios and architectures based on ontologies to support share and autonomy in LBSes, and the University of Pittsburgh (F. Fonseca) on the integration of geographic information using ontologies. There are several groups involved in related P2P data management projects, e.g. A. Kemper's group in Germany (project Service Globe), the database group at Stanford led by H. Garcia-Molina (project Stanford Peers) and the database group at the U. Wisconsin in USA. Collaboration also exists with UC Dan Diego (sponsored by INRIA and NSF)

7. Future Outlook

DBGlobe addressed foundational aspects of data management and revisited most areas of database research including architectures, communication/co-ordination and querying under the perspective of a Global Computing Environment.

Research work in the project has resulted in many important research innovations that were published in major conferences and journals including SIGMOD, ICDE, VLDB ICDT and EDBT.

The key research directions that have been identified through DBGlobe and will be investigated by the research groups involved in the project are:

- *The relationship of peer-to-peer architectures with global computing.*

  Most DBGlobe partners have already started to pursue research in this area. This is demonstrated by their newest publications and a large number of new research projects at which they participate. The peer-to-peer model of autonomous computing nodes with dynamic membership in global computations shares many similarities to global computing. This may lead to cross-fertilization of both research areas. Advances in peer-to-peer research such as Distributed Hash Tables, small-world communities and trust management may have application to the vision of global computing. On the other hand, issues such mobility and ubiquity that have been central in global computing have not been yet addressed by the peer-to-peer community.

- *The importance of context information in global computing*.

  To deal with the huge amount of information available in global computing some form of personalization is necessary. Research work in DBGlobe has demonstrated that context information can be utilized to improve the performance of web service discovery.

The following newly-founded EU projects will perform research on issues that have been more or less originated for the partners work in DBGlobe:

- *AEOLUS* is a proposal that has been submitted to the 2nd Global Computing initiative and has been retained for negotiations. The goal of this project is to investigate the principles and develop the algorithmic methods for building such an overlay computer that enables this efficient and transparent access to the resources of an Internet-based global computer. Part of the continuation of the above research will be carried out under *AEOLUS*.

- MARIE CURIE Intra-European Fellowship - *NGWeMiS* - Next Generation Web Mining & Searching. The project lies in the area of knowledge extraction and management from the massive and heterogeneous document collections on the World Wide Web. The main objective of the proposed project is the design guidelines and prototypes development for next generation web mining and searching techniques based on the P2P paradigm. of Web data. The innovation lies in i. usage of P2P paradigm in the various levels of web content management and searching, ii. the study and development of novel similarity measures among web pages that take into account multiple facets of web pages iii. clustering the web data and meta data taking into account their P2P organization paradigm. (AUEB and INRIA)

- *EDOS*: Environent of the distribution of Open Source Software (http://www.pps.jussieu.fr/~dicosmo/EDOS). *EDOS* is a new european project (IST6) on the management of open source that may use ActiveXML for distribution. (INRIA)

Certain research directions that have resulted as an outcome of the DBGlobe project are being investigated in a number of national research projects:

- Research program PYTHAGORAS II (supported by the Greek Ministry of National Education and Religious Affairs): ΔΙΑΧΕΩ on Database Management over Ad-Hoc Networks (UoI).

- Bilateral cooperation Greece–Germany funded by the Greek General Secretariat of Research and Technology (GSRT) on traffic sensor data integration (CTI).

- "MHTIS: Pervasive Query language based on the relational model", PENEK program aiming to support young scientists. (UCY).

- A new project KadoP on combining AXML with distributed hash tables (DHTs) (INRIA).

- AXML possibly be used in ACI Mase de Donnes MDP2P project on peer to peer data management. (INRIA).

- Research program PYTHAGORAS (supported by the Greek Ministry of National Education and Religious Affairs): On extended pub/sub models for smart-space environments (TUC).

- Research program PYTHAGORAS (supported by the Greek Ministry of National Education and Religious Affairs): Overlay Networks of Context-Aware Web Services: Management, Adaptability and Query Processing (UoI).

- ENTER program of the Greek General Secretariat of Research and Technology (GSRT). Project ΙΧΝΗΛΑΤΗΣ on the development of services for traffic management systems (CTI).

- "Persona: Personalized Portals for the wireless user", PENEK program aiming to support young scientist.(UCY).

- A bilateral project (Cyprus - Greece) funded by the Greek General Secretariat of Research and Technology (GSRT) and the Cyprus Research Promotion

Foundation called "SemaNet: Semantic Overlay Networks of Services in Ubiquitous Computing". (UCY and UoI).

- A bilateral project (Greece - Australia) funded by the Greek General Secretariat of Research and Technology (GSRT) on Data Engineering for Context-Aware Applications (UoI).

- Active AXML is used in RNTL project E.DOT on food risk (INRIA).

- AXML possibly be used in ACI Mase de Donnes MDP2P project on peer to peer data management. (INRIA).

Finally, the DBGlobe project focuses exactly on fundamental research for inventing a framework for Global Computing which will provide in the future the means for designing and using highly distributed and massively networked software systems which will be robust, secure, user-friendly and powerful. There are several projects and experimental systems that provide the (software) infrastructure for running computations on large collections of geographically dispersed computing resources. DBGlobe provides a different data-centric perspective that focus on accessing and manipulating data.