GlobeCon – A Scalable Framework for Context Aware Computing

Kaiyuan Lu, Doron Nussbaum, and Jörg-Rüdiger Sack

School of Computer Science, Carleton University,
1125 Colonel By Drive, Ottawa, Ontario, Canada K1S 5B6
{klu2,nussbaum,sack}@scs.carleton.ca

Abstract. In this paper, we propose a context framework reference model that aims at supporting distributed context aggregation, processing, provision and usage in ubiquitous computing environments. Based on this reference model, we introduce our design of GlobeCon, a scalable context management framework that can be embedded into context aware systems operating in local as well as wide area networks. One of the key aspects of GlobeCon is that it aggregates context information from largely distributed, heterogeneous and disparate sources in a hierarchical manner and presents context aware systems with a unified context access interface. GlobeCon is scalable in numerical, geographical and administrative dimensions, which is achieved by three means: (i) organizing context aggregation hierarchically, (ii) effectively distributing the load of context collection and context processing (i.e., interpreting and reasoning) among local context managers, and (iii) providing a dynamic federation of the distributed directory servers.

1 Introduction

One of the main characteristics of ubiquitous computing is user-centricity where technology automatically moves into the background and adapts to people [17]. Consequently, services developed for ubiquitous computing need to be user-tailored and proactively adapt to users' needs. For this adaptation, services must have sufficient information about users and their immediate and/or possibly future contexts. As a consequence, the notion of context awareness plays an integral role in designing services and systems in the ubiquitous computing paradigm.

The term "context" has been interpreted in many different ways in the literature since it first appeared in [18]. There context was defined as location, identities of nearby people and objects and changes to those objects. One of the most widely accepted definitions, also used in this paper, is given by Dey [7], according to which "Context is any information that can be used to characterize the situation of entities that are considered relevant to the interaction between a user and an application, including the user and the application themselves". Most common entities refer to persons, places or objects.

Early context aware systems, including [1][6][18][21], were aimed at demonstrating the potential of context awareness by prototyping applications. These

G. Kortuem et al. (Eds.): EuroSSC 2007, LNCS 4793, pp. 190–206, 2007.

systems though were domain specific and tightly coupled context acquisition and management with application logic. Lately, researchers have been focusing on the architecture design of context aware systems to support the entire context processing flow [3][7][8][12]. This improves programming abstractions and aids in the development of context aware applications.

Throughout this paper, we attempt to contribute to this development by proposing a scalable context management framework for ubiquitous computing, which we call *GlobeCon*. "Scalable" in this paradigm has three dimensions as specified in [2]: numerical (i.e., the increase of workload of a single resource), geographical (i.e., the expansion of a system from concentration in a local area to a more distributed geographic pattern), and administrative (i.e., the coexistence of a variety of software/hardware platforms in a system).

1.1 Motivation

In previous work [9][10], we have designed and implemented an agent-based framework, referred to as Intelligent Map Agents (IMA), which aimed at providing access to, and support for manipulation of, spatial information. The IMA architecture is user-centric. As such, the IMA provides each user with a personalized set of functionality capabilities, interaction patterns and visualization. Hence, context plays an integral role in our IMA system. However, the usage of context in the current IMA is, at present, somewhat limited in terms of the types of context used (primarily location-based), their capabilities, and the design of context manager (embedded into personal agents). This sparked our interest and provided initial motivation for the research presented here which is our contribution towards the design of a scalable context management framework, GlobeCon. It is intended to be embedded into, or support rapid development of, context aware systems and applications operating in local as well as wide area networks.

1.2 Related Work

Growing rapidly since the Active Badge project [21], research on context aware computing has attracted a lot of attention as witnessed by a large body of literature. For comprehensive discussions, we refer the interested reader to recent survey articles [5][15][19]. We classify work in this research area as belonging to one (or more) of four main fields: designing techniques and sensors with desirable context sensing capabilities; devising context models to facilitate efficient context usage, inference and storage; building frameworks or middleware to support context aware computing; and developing context dependent applications to demonstrate the effects of context. In this paper, we concentrate our work and literature review on the third stream.

Context Toolkit [7] was one of first and influential projects that emphasized the isolation of application logic from the details of context sensing and processing. A conceptual context handling framework, based on a centralized component (i.e., the discoverer), was proposed. This framework provides context gathering, interpretation, and aggregation support for applications. CASS [8] uses a middleware approach to support context awareness for hand-held computers. The middleware resides on a centralized resource-rich server. CASS supports context acquisition, interpretation, and reasoning. CMF [14] is a stand-alone software framework designed for mobile terminals. It supports acquisition and processing of contexts in a user's surroundings, and gives the contexts to applications residing on the user's mobile device. CoBrA [3] is a broker centric agent-based infrastructure supporting context aware computing in dynamic smart spaces such as an intelligent meeting room. It has a central context broker which provides four functional modules: context acquisition, context knowledge base, context reasoning engine and context privacy control.

The above mentioned frameworks have in common a centralized component in their design, which may become a bottleneck when the number of context sources (i.e., sensors) and context consumers increases dramatically. Moreover, they do not scale well in the geographical dimension. SOCAM [12] is service-oriented context aware middleware architecture, which addresses the scalability problem with a global-wide Service Locating Service (SLS)[13]. SOCAM supports context acquisition, discovery, interpretation and context access through a set of independent services, which advertise themselves with the SLS. SCI [11], a generalized context framework, tries to address the scalability problem with a two-layer infrastructure, where the upper layer is a network overlay of partially connected nodes (named Ranges), and the lower layer concerns the contents of each Range in the overlay network. A Range provides context handling services within an area. However, the authors did not mention how to create an overlay network to support the interactions among Ranges, which in our opinion is one of the key challenges in solving the geographical scalability. Similarly, Solar [4] addresses the scalability issue using overlay networks, but does not provide a way to construct such an overlay network. In CoCo [2], the authors argued to utilize the existing infrastructures (i.e., grids, peer-to-peer networks, and Content Delivery Networks) to support scalability in a context management framework. Based on this assumption, CoCo provides an integration layer, which interfaces with heterogeneous context sources, maps the retrieved context to a standard information model, and provides it to context consumers in a unified way.

While recently some research is being carried out on the design of scalable context frameworks, a number of interesting research challenges remain to be solved, e.g., how to orchestrate the sheer mass of contexts from heterogeneous sources, how to support the efficient discovery of appropriate context information, how and where to store the vast amount of contexts, how to maintain the consistency, security and privacy of contexts in large-scale distributed environments, etc. In this paper, we attempt to answer some of the questions.

1.3 Contributions

In this paper, we present our contribution towards supporting context awareness in ubiquitous computing environments. Firstly, inspired by the philosophy of the OSI reference model [20], we propose a context framework reference model that aims at supporting distributed context aggregation, processing, provision and usage. Secondly, based on the reference model, we introduce our design of GlobeCon, a scalable context management framework that supports context aware systems operating in local as well as wide area networks. GlobeCon is different from existing context frameworks [3][8][12][14], which either operate in restricted spaces or have limited capability in network scaling. GlobeCon achieves scalability in numerical, geographical and administrative dimensions by three means: (i) organizes context aggregation in a hierarchical manner, (ii) effectively distributes the load of context collection and context processing (i.e., interpreting and reasoning) among Local Context Managers (LoCoM), and (iii) provides a dynamic federation of the distributed directory servers. GlobeCon also exhibits two additional features: (a) Dual Resource Discovery: a proximity-based discovery of raw context sources and a Universal Discovery Service (UDS) to discover the largely distributed LoCoMs which provide raw contexts as well as processed high-level contexts. This feature promotes the efficiency of discovering context sources and allows flexible context queries; (b) Context Classification: GlobeCon distinguishes *common context* and *personal context*, and applies different storage schemes on the two classes of context, which exhibit different acquisition patterns and require different levels of security and privacy protection.

1.4 Paper Outline

The remainder of this paper is organized as follows. In Section 2, we present design considerations of GlobeCon and our context classification, and propose a context framework reference model. In Section 3, we introduce our architectural design of GlobeCon. In Section 4, we discuss how to integrate GlobeCon with the IMA system and describe a health care application of GlobeCon and the IMA. In Section 5, we conclude the paper and point out our future work.

2 Scalable Context Management Framework

A context management system can be designed and implemented in a variety of ways depending on different requirements and deployment environments. GlobeCon aims at support context awareness. Hence, it needs to provide several fundamental function modules centered around context handling, so that we can relieve developers of context aware applications from the burden of having to deal with the entire context processing flow. The functional modules include:

- *Context acquisition:* collects and updates raw context from largely distributed, dynamic, heterogeneous context sources.
- *Context representation:* provides a unified data model to explicitly represent context semantically, and thus enhance context sharing.
- *Context processing:* processes the context and provides context reasoning (e.g., by combining multiple raw contexts to generate higher level contexts which usually can not be detected by sensors directly).
- *Context storage:* maintains a persistent historical context storage for later retrieval and analysis.
- *Context accessing:* supports efficient processing of plain as well as expressive queries for context information.

In addition to the functional modules, we identify a number of design considerations/issues that are of particular relevance for GlobeCon. Fig. 1 shows the concepts and approaches used in GlobeCon to address the issues.

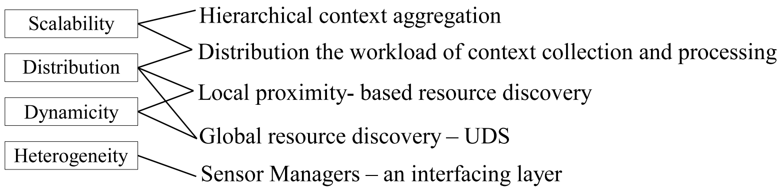

Fig. 1. Key concepts and approaches used to address the design issues

– *Scalability (C.Sca):* exists in three dimensions: numerical, geographical and administrative, as discussed in Section 1. In order to be a scalable framework, GlobeCon needs to have the ability to handle the growing demand in any of the three dimensions in a graceful manner without degrading its performance.

– *Distribution (C.Dis):* refers to the large number of raw context sources and context processing resources which are spread over large geographic areas. GlobeCon should shield the distribution from the end users.

– *Dynamicity (C.Dyn):* lies in two aspects: (i) context sources and users may be mobile and changing their states, e.g., a sensor moves from place to place and switches its status between active and inactive; (ii) the pervasive computing environment itself is highly dynamic, e.g., the traffic condition of a road network.

– *Heterogeneity (C.Htg):* arises in any system dealing with multiple data sources. GlobeCon should handle context data residing at heterogeneous and disparate sources (i.e., context sources with different interfaces and raw context data in different formats) and needs to provide these context with a unified view.

Privacy/security and fault tolerance are certainly important issues for systems like GlobeCon. However, they are not the main focus of our current work.

2.1 Our View of Context

In this section, we present our view of context, which classifies context as common and personal. *Common context* describes common places or spaces, and objects that are shared among all users or a group of users. Common contexts include physical context (e.g., light density, noise level, temperature, air pressure etc.), logical context (e.g., business processes, price, etc.), and technical context of common computing devices (e.g., processing power, memory capacity, bandwidth, latency etc.). *Personal context* describes a user's profile, physical, social and emotional situation, and objects/ entities owned by the user. Personal contexts can be further divided into domain specific context which are sets of information applied respectively to individual application domains, and general personal context which applies to most application domains (e.g., user profile, preference, the technical aspects of user's personal computing devices, etc.). As to the classification of application domains, the North American Industry Classification System (NAICS) [16] provides a guideline as a starting point.

2.2 Context Framework Reference Model

In ubiquitous computing environments the amount of context information is vast and the number of context sources, which are usually widely dispersed throughout the globe, can be very large too. To make it easier for applications to find context sources which can provide the required context, we need a solution that manages and indexes distributed context sources in a way that required context can be discovered efficiently and without (or with minimal) user interactions. Inspired by the philosophy of the OSI (Open Systems Interconnection) Reference Model [20] for computer networks, we are proposing a layered model for scalable context frameworks that provide distributed context handling support for context aware systems operating in local as well as wide area networks.

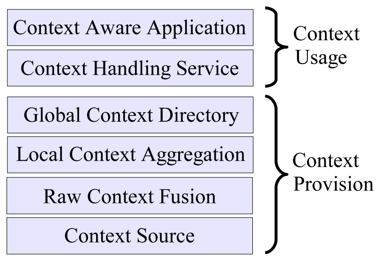

Fig. 2. Context Framework Reference Model

Our model has six layers as shown in Fig. 2: (a) context source layer, (b) raw context fusion layer, (c) local context aggregation layer, (d) global context directory layer, (e) context handling service layer, and (f) context aware application layer. Each layer built upon the one below it. The six layers together provide support for context aggregation, processing, provision and usage. At the lower "*context provision*" hierarchy, each layer offers a well defined set of context provision services. With each level of the hierarchy, users are exposed to and able to access more sophisticated contextual features.

– The context source layer comprises the ground level context sources, i.e., sensors. A context source is responsible for observing raw contexts within its sensing range and capability(ies), and reporting them to the layer above it.
– The raw context fusion layer has the tasks of interacting and gathering raw context from a number of adjacent context sources with heterogeneous APIs. Another task of this layer is to map the raw context obtained in various formats into a unified context representation, which should be compliant with a pre-specified context model.
– The local context aggregation layer collects context from raw context fusion layer, therefore its local context knowledge base covers a larger geographical

and logical area. This layer also preforms reasoning on its local context to reach high level context which is more meaningful for high level applications. The reasoning is usually based on pre-defined domain knowledge.
- The global context directory layer exists as a global registry and a matching tool, which enable context consumers to search within the context provision hierarchy for context providers having the needed context information.

The upper "*context usage*" portion of this model is concerned with the usage of context provided by the lower context provision hierarchy.

- The context aware application layer contains a variety of ubiquitous services that are context dependent and aimed at end-users. Each of such customizable service is usually developed for one particular application domain, e.g., health care, tourism, ubiquitous GIS and etc.
- The context handling service layer is a middleware layer residing between the application layer and the context provision hierarchy. It provides application domain specific, large-scale context handling functions (e.g., reasoning and aggregation) that are requested sufficiently often by services in the application layer, for example, a context inference service for a ubiquitous health care system, which performs reasoning on the context gathered from the lower context provision hierarchy, based on health care domain knowledge.

The key advantage of this reference model is that the aggregation of largely distributed context information is accomplished in a hierarchical way, thereby, each layer only needs to focus on its own tasks. The complexity of other tasks is shielded in other layers, and queries for context of a particular level of detail can be addressed directly to the corresponding layer.

3 Architectural Design of GlobeCon

Based on the reference model of Section 2.2, we introduce our architectural design of GlobeCon, a scalable context management framework. Fig. 3 depicts an overview of the multi-layered GlobeCon architecture, and gives an explicit mapping between GlobeCon and the reference model. The lowest level encompasses a variety of heterogeneous context sensors. The second level is comprised of Sensor Managers (SM) which manage their own set of sensors within their geographical vicinities. The third level contains Local Context Managers (LoCoM). Each LoCoM manages a number of SMs within their own physical or logical vicinities. A Universal Directory Service (UDS), which resides on the fourth level, provides a global registry for LoCoMs. This is followed by the context usage hierarchy. Context handling services provides larger scale context handling functionalities (e.g., reasoning and aggregation) beyond the scope of individual LoCoMs. Context aware applications are end-user oriented context dependent services. Each layer represents a fairly complex subsystem focusing on a different phase of the entire context processing flow. These layers are seamlessly integrated together to support distributed and scalable context aggregation, processing, provision and usage. In the following subsection, we elaborate on the role of each layers.

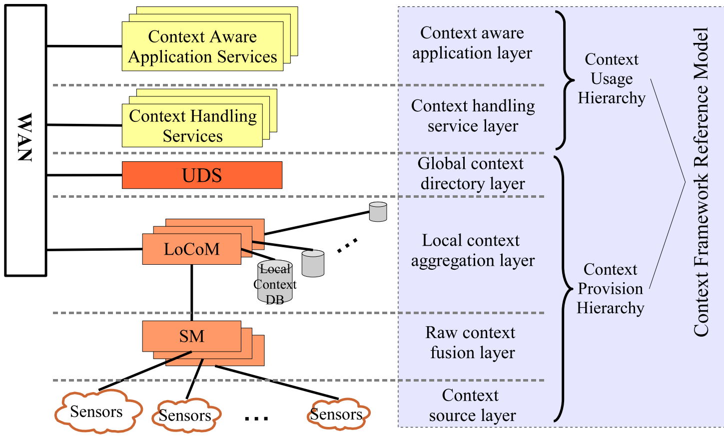

Fig. 3. An overview of GlobeCon architecture

3.1 Sensor Level

Sensors are the ground level context sources. A sensor in GlobeCon is responsible for acquiring raw context data within its sensing range and capability, connecting with and reporting the obtained context to its SM(s) via wired or wireless communication channel. In GlobeCon, a sensor proactively register itself with a SM after deployment. This is achievable by two alternative ways: (i) a sensor has pre-knowledge of which SM it should be connected with (e.g., preconfigure a sensor with this information before its deployment); and (ii) a sensor can discover available SMs by listening "*SM-advertisement*" messages within its communication range. For both of the two approaches, authentication and authorization process may necessary to ensure privacy and security.

A sensor can be either stationary (e.g., sensors mounted on the walls) or mobile (e.g., in-vehicle sensors and biosensors attached on a person). In terms of form, sensors in GlobeCon consist of software sensors and hardware sensors. A software sensor captures context directly by accessing and reading from existing software applications or services, such as obtaining CPU usage from the OS of a laptop, or accessing a user's schedule using APIs of the e-calendar on his PDA. Sophisticated software sensors are able to apply domain logic to deduce higher level context, e.g., identifying a user's availability by calculating the percentage of free time in his e-calendar. A hardware sensor is a physical device usually of small size, measuring physical aspects of context, e.g., light, noise and temperature. Modern hardware sensors are more advanced since they are typically equipped with small processor and disk spaces to perform simple on-sensor processing.

In some cases, depending on communication resources, a number of sensors can collaborate with each other to form a network and a data sink collects context from these sensors via multi-hop communication. In this scenario, the entire sensor network is considered as one logic sensor and the data sink acts as a portal to SM(s). With the networking of sensors, sensing coverage is effectively enlarged and sensing capabilities are effectively federated.

3.2 Sensor Manager

A SM manages its own set of sensors within its vicinity. A sensor-SM relationship is initiated by the sensor. The aim of SMs is to relieve a LoCoM's burden of aggregating context from any number of low level heterogeneous sensors. Acting as a gateway between sensors and LoCoMs, SMs can support multiple network interfaces (e.g., infrared, RF and phone-line) and cater to the heterogeneity of sensors. In order to accomplish this, a SM performs the following tasks, which consequently addresses the design considerations of *C.Dyn*, *C.Htg* and *C.Dis*:

- Publishing its availability by periodically broadcasting "*SM-advertisement*" messages over its wired/wireless LAN, so that local sensors can discover it.
- Initializing, configuring and synchronizing its sensors, e.g., set data sampling rate, specify data acquisition method (on-demand or event-driven).
- Monitoring and controlling the operations of its sensors, e.g., detect malfunctioning sensors and take appropriate actions (i.e., reset or de-register a sensor) to avoid inaccurate or stale context information.
- Gathering raw contexts from sensors in two possible transmission modes: push-mode in cases of periodical stream-based transmission, and pull-mode in cases of irregular requests on raw sensory readings.
- Associating obtained context with subject and sampling time, and converting the raw context in heterogeneous formats into a unified context representation in accordance with pre-specified context model.
- Detecting and resolving potential context information correlation and inconsistency existed in the obtained contexts.
- Discovering and registering with the available LoCoMs by listening "*LoCoM-advertisement*" messages within its communication range. If there are more than one available LoCoMs, a SM selects one as its manager and marks the rest as context subscribers.

Within the same physical space, there may be several SMs where each SM works for a particular application domain. For example in a hospital cafeteria where visitors come to eat, there is one SM collecting health care related contexts for visitors (e.g., temperature, humidity and heart activity), while another SM gathering dining related contexts (e.g., user's food preference). In such cases, a sensor with multiple sensing capabilities may report its context data to more than one SMs[1], as shown in Fig. 4. GlobeCon has two types of SMs: stationary sensor manger (SSM) and mobile sensor manager (MMS). Due to its mobility, a mobile SM may be out of the range of any LoCoMs. In this case, alternative schemes are devised to avoid context information loss/unavailability. For example, instead of tossing the local context away, a MSM may temporarily cache the obtained

[1] A sensor may report the same piece of context to multiple SMs to achieve high degree of availability of context information with the penalty of data redundancy.

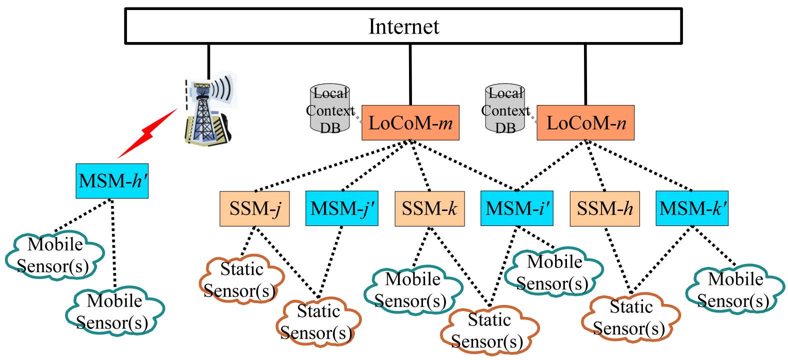

Fig. 4. An example illustrating the subsystem of LoCoM, SMs and Sensors

context locally (which requires large local disk space) and report its cache to a LoCoM when a LoCoM is available later, or a MSM may compress the context data, or it may tune sensors' sampling rate to reduce the volume of context data.

3.3 Local Context Manager

A LoCoM is responsible for aggregating context from SMs distributed within its local coverage. "Local" here indicates a logic relationship between a LoCoM and its SMs, and is not necessary limited to the proximity at the geographical level. It may imply proximity at the network level (e.g., within the same WAN) or proximity at the level of application domain(e.g., health care or tourism). LoCoMs are designed to perform computation and communication intensive functionalities (e.g., conducting context reasoning and processing potentially large numbers of context queries) within the context management processing flow, therefore, they are usually hosted by stationary device(s) with high performance computing, and high bandwidth network connections. One stream of the main tasks of a LoCoM is related to SMs management:

- Periodically broadcasting "*LoCoM-advertisement*" messages over its wired and wireless WAN so that SMs within its communication range are able to be aware of its availability.
- Handling registration requests from local SMs, and maintaining a list of registered SMs. There are two types of SM registration, where a LoCoM plays different roles respectively: SM manager or SM subscriber.
- Detecting and de-registering inactive SMs due to failure or out-of-connection. As a consequence of de-registration, a LoCoM needs to update accordingly its self-description (refer to the last paragraph in this subsection).
- Gathering context information from the registered active SMs.

As a SM manager, a LoCoM provides a local context database to store the *common context* [2] retrieved from the SMs under its management. Historical *Personal*

[2] For definition of *common context* and *personal context*, please refer to Section 2.1.

context may not be stored at LoCoMs for two reasons: (i) users are mobile, which implies that a user may interact with several LoCoMs from time to time. Storing users' personal context at LoCoMs makes the retrieval of their personal historical contexts difficult. In worst case, a query needs to visit all LoCoMs to collect a user's historical personal context; (ii) personal context is usually of high sensitive nature and requires higher level of privacy protection. Storing personal context at distributed LoCoMs poses more challenges on security and privacy control.

LoCoM provides a context inference engine, which is able to deduce high-level context that is not directly observable from sensors. Inference is usually based on domain knowledge that is often organized as sets of rules. A LoCoM conducts reasoning only on its local context which may span multiply application domains. At a higher level of GlobeCon, there are application domain specific context reasoning services, which conduct larger scale inference based on context information gathered from multiple LoCoMs. The main reason of having two levels of context reasoning is to effectively balance the computational load of context inference which, as demonstrated in [12], is a high computation intensive task, and becomes a bottleneck if loads are not distributed strategically.

In order for the local context to be global-wide retrievable by the services in the *context usage hierarchy*, LoCoMs register with the Universal Directory Service (UDS) and provide external access to their local context knowledge base via the Internet. First, a LoCoM generates a self-description on what kind of context information it can provide in accordance with its list of SMs. It then submits the self-description and its communication interfaces to the UDS. After knowing a LoCoM via the UDS, a context consumer (i.e., the services in the context usage hierarchy) can issue context queries directly to the LoCoM.

3.4 Universal Directory Service

The universal directory service (UDS) is at the heart of the GlobeCon framework. It contains all required information about the available LoCoMs, which can be viewed as services by UDS, and the context handling services (refer to Section 3.3.5), and how to reach and make use of them. Thus, higher level applications/services can be aware of the availability of these services via the UDS. The UDS can be viewed as one logical registry, while in real implementation and deployment, it comprises many distributed UDS servers that are federated together. The federation of UDS servers enables GlobeCon to operate in dynamic and distributed environments that address local as well as global needs.

The UDS constructs a hierarchy of the registered services. Fig. 5 shows an example of such a hierarchical service space. The top level of the hierarchy consists of LoCoMs, 3rd party information servers (e.g., an existing weather information server) and context handling services. The service space hierarchy can be extended to include context aware application services at the top layer of GlobeCon. While drilling down the hierarchy further, the classifications used by the UDS can be either locations, types of provided context, or application domains. In order to not overload any single UDS server, the tree-like hierarchical structure is divided into non-overlapping zones. The division is dynamic to

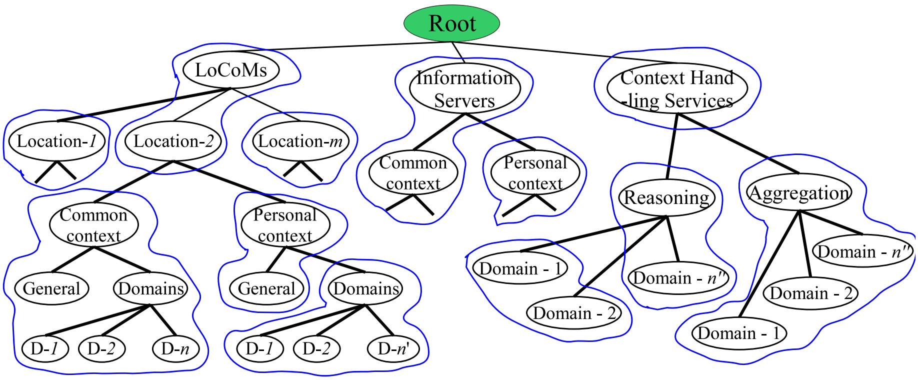

Fig. 5. Part of the hierarchical UDS service space with the division into zones

further balance the loads in cases of new service registrations and de-registrations of existing services. The advantages of this hierarchical structure are two fold. Firstly, it allows the UDS to efficiently manage the large numbers of services with a variety of types. Secondly, it certainly speeds up lookup processing for location-based and application domain based services. Because in order to find a match, a lookup request traverses only the UDS servers along one (or a few, in case of more complex queries) branch in the tree structure, instead of traversing all of the UDS servers. Therefore, this UDS design particularly addresses the design considerations of *C.Sca*, *C.Htg*, *C.Dis* and *C.Dyn*.

3.5 Context Usage Layers

The upper two layers in Fig. 3 map directly to the *context usage* hierarchy of the context framework reference model. The two layers include two groups of services: context handling services and context aware application services. The first group provides large scale, domain specific context handling services (e.g., context reasoning and context aggregation) that are beyond the service scope of individual LoCoMs. For example, they can be context inference services which perform context reasoning on context information gathered from multiple LoCoMs. One such context reasoning service usually focuses on one particular application domain. Another example can be context aggregation services, which collect context of certain type (e.g., location) or of certain event (e.g., traffic jams) from multiple LoCoMs and store the gathered context at one logic location. With the help of context aggregation services, the access to the largely distributed context sources can be transparent to the higher level applications. This group of context handling services will register themselves with the UDS.

The other group of services is a variety of ubiquitous context aware services at the application level, aiming at end users. Each of these services utilizes the context information acquired from the lower *context provision hierarchy* and provides user-oriented context aware services on a particular application domain. Even though it is not mandatory for them to register with the UDS, these services may utilize the UDS to search, discover and make use of the appropriate services from the previous group registered with UDS. Examples of such context aware specialized services are in health care, tourism, ubiquitous GIS, road emergency help, target tracking, etc. For each type of service from either of the two service groups, there may be many of them available on the Internet. They may appear as competing services and can provide the same service with different QoS parameters and different pricing.

4 Embedding GlobeCon with the IMA

GlobeCon is designed to be general purpose, we plan to show the validity of the concept by integrating GlobeCon with the IMA. The IMA, Intelligent Map Agents, is an agent-based user-centric framework. It aims at providing access to, and support for manipulation of, spatial information via a set of personalized services. In this setting, context plays an integral role in supporting user-tailored and user-adaptive features of the IMA system. A layered view of the IMA architecture is depicted in Fig. 6. By embedding GlobeCon within the IMA, we can seamlessly integrate the IMA system with the underlying sensor-rich environments, which capture surrounding contexts silently. Furthermore, GlobeCon facilitates IMA components to easily access the context acquired by the GlobeCon context aggregation hierarchy. Depending on context types, two extra components are devised to facilitate the interaction between GlobeCon and the IMA: *Personal Context Manager* and *Common Context Proxy*, as shown in Fig. 6.

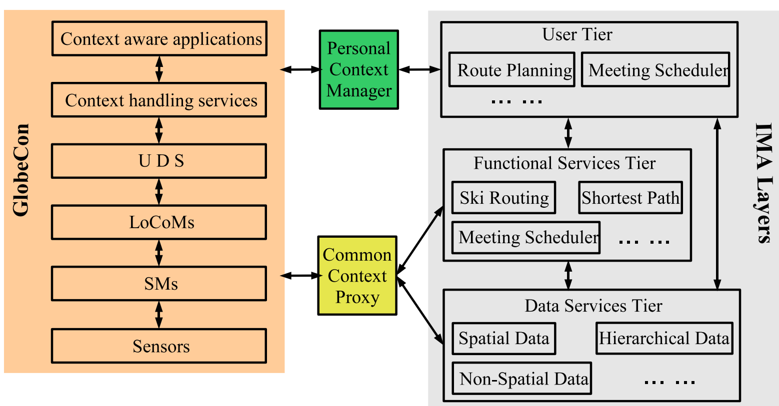

Fig. 6. Embedding GlobeCon with IMA by adding two interfacing components

Personal Context Manager (PCM) represents private and intimate knowledge about a user, and allows IMA personal agents to obtain a user's personal context. It can be viewed as services and mapped to the context handling service layer in the GlobeCon. There is one PCM for a user or a group of users depending on applications. The personal PCM for a user (or a group of users) is the subscriber for all his/her (or their) personal contexts that are captured and processed by the LoCoM, SMs and sensors subsystem. For example, in this scenario, a mobile user John, wearing a detectable ID tag (e.g., RFID tag) and other possible wearable sensors, is moving around in potentially very large geographical areas. When John is within the range of a LoCoM, the MSM, which resides on his PDA and manages his RFID and other wearable sensors, will register with the LoCoM. The LoCoM will then be aware of John's PCM (either notified by the connection request initiated by the PCM, or told by this MSM) and relay all of John's local personal context[3] to his PCM. If this MSM is out of the range of any LoCoMs, it may take alternative actions described in Section 3.3.2 and ultimately relays John's personal context captured by its sensors to John's PCM. Personal contexts that are relayed and/or stored at a PCM can then be queried and utilized by the IMA services to support customization and personalization.

Common Context Proxy (CCP) is used by the IMA to fetch common context from the GlobeCon system. For the usage of common context, the IMA services can be mapped directly to the top layer of the GlobeCon architecture shown in Fig. 3. That is, a IMA service may issue lookup queries to the UDS searching for appropriate LoCoMs or/and context handling services that hold the required common context, and subscribes context directly from them. However, letting heterogeneous IMA services to interact directly with LoCoMs and context handling services, may involve re-engineering their communication interfaces. Therefore, employing the concept of separation of concerns, CCP comes in place to provide an integration layer which bridges the smooth interaction between IMA services and the GlobeCon. In this sense, CCP can be implemented as a special IMA service that accepts context requests from other IMA services and acts upon these services to query the GlobeCon for required common context.

4.1 A Use Case in an Intelligent Health Care Application

In this section, we present a use case of the GlobeCon architecture. It is an intelligent context aware health care application as shown in Fig. 7. The following scenario demonstrates the context data/processing flow of GlobeCon, as well as the communication and interaction between the distributed services.

This application involves a number of context aware services. The services provided by the IMA system include: the *patient monitoring service* (PM) which gathers real-time physical data of patients, examines the data, determines critical conditions based on which it either issues recommendations to the patients or calls other health care related services; the *medical service* (MD) which is responsible for assigning patients to appropriate physician regardless of patients' location; the *emergency/ambulance service* (EM) that can find and dispatch the nearest available ambulance to patients' location; the *shortest path service* (SP)

[3] Note that the local personal context provided by a LoCoM can be provided by either John's personal wearable sensors or the stationary sensors under the management of the LoCoM (e.g., a video camera mounted on the entrance door of a grocery store).

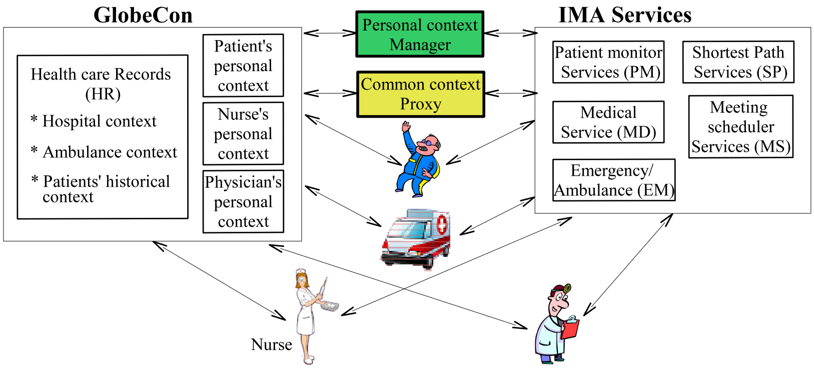

Fig. 7. An intelligent health care application using the GlobeCon framework

which takes points of a road network as input and computes shortest cost path (available in our IMA) between the given points where cost can vary e.g., from distance to travel time; and the *meeting scheduler service* (MS) which explores a number of ways for coordinating a meeting between users [9]. The GlobeCon provides the *health records service* (HR) which collects and maintains historical health care related records. The GlobeCon context provision hierarchy captures and aggregates the personal context of the patient, the physician and the nurse.

Consider a patient wearing a number of physical sensors which monitor his heart rate, blood pressure, body temperature and body humidity etc. Those sensors are managed by a MSM residing on the patient's PDA. By noticing the patient's appearance via the UDS, the PM and HR subscribe his physical context from the MSM via LoCoMs. Noticed that, whenever applicable, these physical context can be delivered to the PM and HR through LoCoMs which usually have high bandwidth connections. When the PM detects a critical condition based on the patient's current context, let's say abnormal blood pressure, which may cause serious illness if not treated properly, the PM will invoke the MD which fetches the patient's historical health record from the HR and his current physical context. Based on the patient's current context and historical health record, the MD assigns a physician to him using a pre-defined assignment algorithm. The MS will then be invoked by the MD to compute a feasible meeting time/location for the patient and the assigned physician, which takes into account the context of their meeting preference and available time acquired via the GlobeCon system. During the meeting calculation, the MS calls the SP to get shortest cost paths for the patient and the physician.

5 Conclusions and Future Work

One of the major challenges in providing scalable context management is flexibility and mobility, since the sensors, computing resources and end users are dynamic and mobile in ubiquitous computing environment. Therefore, in this paper, we propose a context framework reference model aimed at supporting distributed context aggregation, processing, provision and usage. Based on this reference model, we introduce our design of GlobeCon, a scalable context management framework that can be embedded into context aware systems operating in local as well as wide area networks.

We currently focus on the initial phase of GlobeCon prototype implementation. Then, we plan to embed GlobeCon with our existing implementation of the IMA system. By providing context aggregation and provision supports to the IMA applications, we can further demonstrate the feasibility and performance of GlobeCon from experimental point of view.

Although privacy and security are an integral part of any context management frameworks, we do not focus on these in our work. It is partially because privacy and security are self-contained research topics. Further, we plan to integrate, in the future, existing privacy and security components and policies to our framework. However, we do plan to conduct research on the privacy/security issue which in particular related to context, namely the threat of potential information leakage, which refers to unintentional disclosure of private information through knowledge mining process based on public information.

References

1. Abowd, G., Atkeson, C., Hong, J., Long, S., Kooper, R., Pinkerton, M.: Cyberguide: A mobile context-aware tour guide. ACM Wireless Networks 3, 421–433 (1997)
2. Buchholz, T., Linnhoff-Popien, C.: Towards realizing global scalability in context-Aware systems. In: 1st Int'l. Workshop on Location- and Context-Awareness (2005)
3. Chen, H., Finin, T., Joshi, A.: Semantic web in the context broker architecture. In: Proc. of IEEE Int'l. Conference on Pervasive Computing and Communications, IEEE Computer Society Press, Los Alamitos (2004)
4. Chen, G., Kotz, D.: Solar: An open platform for context-aware mobile applications. In: Proc. of 1st Int'l. Conference on Pervasive Computing, pp. 41–47 (2002)
5. Chen, G., Kotz, D.: A survey of context-aware mobile computing research. Dartmouth College Technical Report TR2000-381 (November 2000)
6. Cheverst, K., Davies, N., Mitchell, K., Friday, A.: Experiences of developing and deploying a context-aware tourist guide: the GUIDE project. In: Proc. of 6th Annual Int'l. Conference on Mobile Computing and Networking, pp. 20–31 (2000)
7. Dey, A.K., Salber, D., Abowd, G.G.: A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. Journal of Human-Computer Interaction (HCI) 16(2-4), 97–166 (2001)
8. Fahy, P., Clarke, S.: CASS: a middleware for mobile context-aware applications. In: Workshop on Context Awareness, MobiSys (2004)
9. Gervais, E., Liu, H., Nussbaum, D., Roh, Y.-S., Sack, J.-R., Yi, J.: Intelligent Map Agents - A ubiquitous personalized GIS. Accepted by ISPRS Journal of Photogrammetry and Remote Sensing, special issue on Distributed Geoinformatics (2007)
10. Gervais, E., Nussbaum, D., Sack, J.-R.: DynaMap: a context aware dynamic map application. presented at GISPlanet, Estoril, Lisbon, Portugal (May 2005)

11. Glassey, R., Stevenson, G., Richmond, M., Nixon, P., Terzis, S., Wang, F., Ferguson, R.I.: Towards a middleware for generalised context management. In: 1st Int'l. Workshop on Middleware for Pervasive and Ad Hoc Computing (2003)
12. Gu, T., Pung, H.K., Zhang, D.Q.: A service-oriented middleware for building context aware services. Network and Computer Applications 28(1), 1–18 (2005)
13. Gu, T., Qian, H.C., Yao, J.K., Pung, H.K.: An architecture for flexible service discovery in OCTOPUS. In: Proc. of the 12th Int'l. Conference on Computer Communications and Networks (ICCCN), Dallas, Texas (October 2003)
14. Korpipaa, P., Mantyjarvi, J., Kela, J., Keranen, H., Malm, K.J.: Managing context information in mobile devices. IEEE Pervasive Computing 2(3), 42–51 (2003)
15. Mitchell, K.: A survey of context-aware computing, Lancaster University, Technical Report (March 2002)
16. NAICS: North American Industry Classification System, http://www.naics.org
17. Satyanarayanan, M.: Pervasive computing: vision and challenges. IEEE Personal Communication Magazine 8(4), 110–117 (2001)
18. Schilit, B., Theimer, M.: Disseminating active map information to mobile hosts. IEEE Network. 8(5), 22–32 (1994)
19. Strang, T., Claudia, L.P.: A context modeling survey. In: Davies, N., Mynatt, E.D., Siio, I. (eds.) UbiComp 2004. LNCS, vol. 3205, pp. 34–41. Springer, Heidelberg (2004)
20. Tanenbaum, A.S.: Computer Networks, 3rd edn. Prentice Hall, Inc., Englewood Cliffs (1996)
21. Want, R., Hopper, A., Falcao, V., Gibbons, J.: The active badge location system. ACM Transactions on Information Systems 10(1), 91–102 (1992)