HiCon: A Hierarchical Context Monitoring and Composition Framework for Next-Generation Context-Aware Services

Kyungmin Cho, Inseok Hwang, Seungwoo Kang, Byoungjip Kim, Jinwon Lee, SangJeong Lee, Souneil Park, and Junehwa Song, Korea Advanced Institute of Science and Technology
Yunseok Rhee, Hankuk University of Foreign Studies

Abstract

This article presents a hierarchical context monitoring and composition framework that effectively supports next-generation context-aware services. The upcoming ubiquitous space will be covered with innumerable sensors and tiny devices, which ceaselessly pump out a huge volume of data. This data gives us an opportunity for numerous proactive and intelligent services. The services require extensive understanding of rich and comprehensive contexts in real time. The framework provides three hierarchical abstractions: PocketMon (personal), HiperMon (regional), and EGI (global). The framework provides effective approaches to combining context from each level, thereby allowing us to create a rich set of applications, not possible otherwise. It deals with an extensively broad spectrum of contexts, from personal to worldwide in terms of scale, and from crude to highly processed in terms of complexity. It also facilitates efficient context monitoring and addresses the performance issues, achieving a high level of scalability. We have prototyped the proposed framework and several applications running on top of it in order to demonstrate its effectiveness.

The upcoming ubiquitous space, fully instrumented with innumerable sensors and tiny devices, will see sensors ceaselessly pumping out a huge volume of dynamic data including biomedical status, location, and activity of individuals, traffic status of metropolitan areas, and environmental status such as weather and air pollution. The evolution and convergence of future networks will enable us to access and share the data anytime and anywhere. They will also make it possible to generate more value-added information through flexible and dynamic data composition. These events will facilitate the advent of new context-aware services running in a proactive and intelligent manner. Next-generation context-aware services will require extensive and real-time understanding of rich and comprehensive contexts on individuals, communities, businesses, and so on. However, the space is too chaotic to be effectively explored and utilized for the services. To handle such a grand challenge, we envision that infrastructural support is the most feasible and economic solution. The complexity and scale of the space and services make it almost impossible for individual developers to build and operate their services in their own independent ways.

In this article we present *HiCon*, a novel framework for hierarchical context monitoring and composition that supports advanced context-aware services requiring scalable monitoring and composition of dynamic contexts. By carefully investigating the semantics and structures of contexts, services, and underlying networks, we identify three major challenges in developing the proposed framework. First, the framework should support dynamic composition of diverse contexts covering various scales such as personal, local, city-wide, and global. Moreover, context composition and monitoring should be able to be performed in horizontal and vertical manners. Second, the framework should well reflect the hierarchical nature of networks mapping to the ubiquitous space with different coverage (e.g., BAN/PAN, MAN, and WAN). Finally and most important, the framework should handle the massive scale of computation for context monitoring and composition. The number of objects to be monitored will be huge (e.g., tens of millions or more). The number of clients would reach up to the entire population of a metropolis, a nation, and even the globe. To address these challenges, we classify the compound semantics of contexts into three levels, *personal*, *regional*, and *global* contexts, and design a hierarchical framework of three layers. The framework effectively deals with a broad spectrum of contexts, from personal to worldwide in terms of scale, and from crude to highly processed in terms of complexity.

To the best of our knowledge, our work is the first attempt to develop a large-scale infrastructure for context monitoring and composition. The proposed hierarchical framework consists of three context processing layers, corresponding to the personal, regional, and global context levels. *PocketMon* is a personal context monitoring platform running on a mobile device inside individuals' pockets. *HiperMon* is a high-performance system for massive context data processing at the regional context level. *EGI*, standing for efficient global (context delivery) infrastructure, is designed for effective wide-area context data delivery. The system components perform horizontal context composition at each level and cooperate with each other for vertical context composition. In addition, the hierarchical design facilitates context abstraction at the lowest possible level of the hierarchy and reduces costly data transmission to the higher level. Additionally, at each layer we address several system issues and technical challenges such as computing resource and battery limitation, massive data processing and delivery, network bandwidth overhead, data selectivity, and latency. We have prototyped the proposed framework and several applications such as Running Bomber, SympaThings, Ubiquitous taxiCab, and U-BattleWatcher. These applications drive the framework architecture, and the prototype validates our framework.

The remainder of the article is organized as follows. We present an overview of the proposed framework. We explain existing research on context-aware services and highlight the contribution of our work. We describe the system components in detail, and briefly discuss implementation and evaluation results. We conclude the article with a summary.

Hierarchical Context Monitoring and Composition Framework

Contexts and Composition

As the first step in designing the proposed framework, we carefully investigate promising contexts and classify them. Since contexts could be an abstraction of one or more types of sensor data, classifying them also means finding out abstraction scopes where contexts are very likely to be found. In this work we mainly look into the types of services in terms of contexts, network connectivity and coverage, relevance between activities, and the scope involved in context composition. Based on such features, contexts are categorized into three different levels: personal, regional, and global.

Personal context represents the status of individuals mainly related to everyday life. It includes a wide spectrum of information, from rather static information such as preferences to highly dynamic information such as activities and physiological status. Those vary from simple sensed data such as location, temperature, and illumination up to quite complicated information such as actions, activities, and physiological and emotional status. Personal contexts are usually collected from a number of sensors deployed in business/personal area networks (BANs/PANs).

Regional context is a kind of collective information over wide regions such as metropolitan areas. It is closely related to a large number of individual objects and their aggregated features. Regional contexts include population, traffic situation, status of buildings and bridges, weather, air pollution, water pollution, and electric supply in the region of interest. Regional contexts are usually induced from a large number of individual contexts (e.g., locations of people and vehicles to figure out pedestrians' distribution and traffic status). Metropolitan area networks (MANs) or wide-area networks (WANs) covering regional areas are used to collect the corresponding regional contexts.

Global context is the higher-level information that aggregates data throughout a global area beyond a single regional area, ultimately worldwide. The common examples are global weather, worldwide hot news, and financial and economic data such as business-related data of multinational enterprises. Such global contexts from worldwide sources will be delivered through the Internet to possibly millions of subscribers.

Deriving more abstract and higher level of contexts requires various compositions of delicate and lower-level contexts. Such compositions can be performed across multiple context levels as well as within a single context level. We regard context composition involving multiple levels of contexts as vertical composition. On the other hand, composition performed within a level is horizontal composition. For example, the emotional status of a person can be derived from heart rate, voice tone, body temperature, and location; it is a horizontally composed context in the personal context level. As an example for vertical composition, we can consider a personal healthcare service that helps a patient be delivered to the closest and most appropriate hospital in an emergency. The service needs vertical context composition utilizing personal contexts such as his/her physical status and regional contexts such as traffic conditions and medical staff availability in a city. Regional or personal contexts can be also used to compose global contexts. For example, we can expect a global planning service that helps business owners make agile and proper business decisions by providing collective and abstract context data based on a lot of worldwide data (e.g., stock trading, wholesale prices, supply chain, regional weather, and hot news).

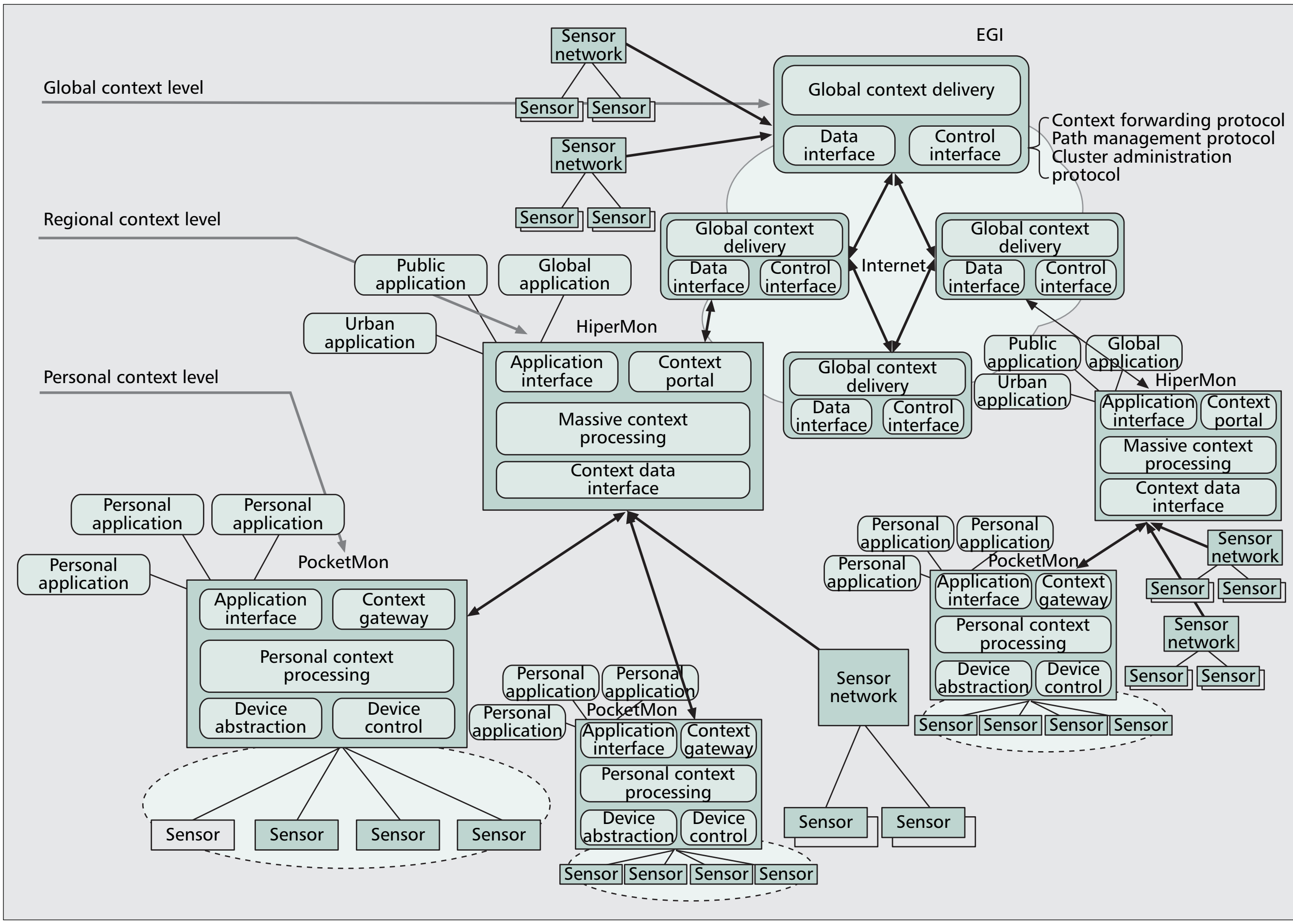
Framework Design

In the early stage of deployment of context-aware applications and services, each application will implement its own monitoring scheme. As the demand for context monitoring increases, the advent of a dedicated framework providing infrastructural support for the applications will be expedited. The framework will contain the common functions of context monitoring. Once established, applications will delegate most processing to the framework. Application developers concentrate on developing core service logics and do not spend time implementing raw monitoring systems. Moreover, the framework will efficiently utilize sharable resources as well as greatly reduce maintenance costs. Eventually, the framework can trigger the proliferation of advanced context-aware services.

Developing such a framework inherently involves several challenging issues due to its scale and sophisticated functionalities. Among them, this article mainly focuses on two issues; dynamic context composition and monitoring scalability.

Our framework is designed to have a hierarchical structure that facilitates flexible and dynamic context composition, and achieves a high level of scalability. Figure 1 shows the architecture of the framework and its components corresponding to the three context levels. We develop three layers responsible for horizontal context composition and interlayer communication interfaces for vertical composition. The hierarchical framework enables context abstraction at the lowest possible level of the hierarchy for processing efficiency and security. The framework also exploits a localization of context processing and consists of the following (the details of the components are discussed):

- *PocketMon* at the personal context level takes charge of personal context monitoring for individuals, running on mobile devices. PocketMon manages numerous sensors of diverse types in BANs/PANs and composes delicate personal contexts. If needed, regional and global contexts processed in HiperMon can also be fed into PocketMon as a data source. PocketMon provides a carefully designed context monitoring interface for running applications. It also provides a gateway to serve the higher layers for vertical composition.
- *HiperMon* at the regional context level monitors collective contexts about a group of people as well as static or dynamic objects in a certain regional area. HiperMon has a lot of connections to PocketMons existing in the region and collects necessary data from them. It also connects to widely



■ Figure 1. *HiCon: A hierarchical context monitoring and composition framework.*

deployed sensor networks regarding vehicles, buildings, bridges, streets, and so on. Various kinds of context-aware applications (urban, public, and global) can run atop HiperMon using its application interface.
- *EGI* at the global context level is an efficient data delivery infrastructure for global context composition and monitoring. Together with processing, composing global contexts requires a large volume of data delivered from geographically distant data sources. Public sensor networks or HiperMons open to the public can also be data sources. EGI is not directly involved in processing context data, but plays a key role in building and managing a global context delivery overlay connecting HiperMon nodes.

Review of Related Work

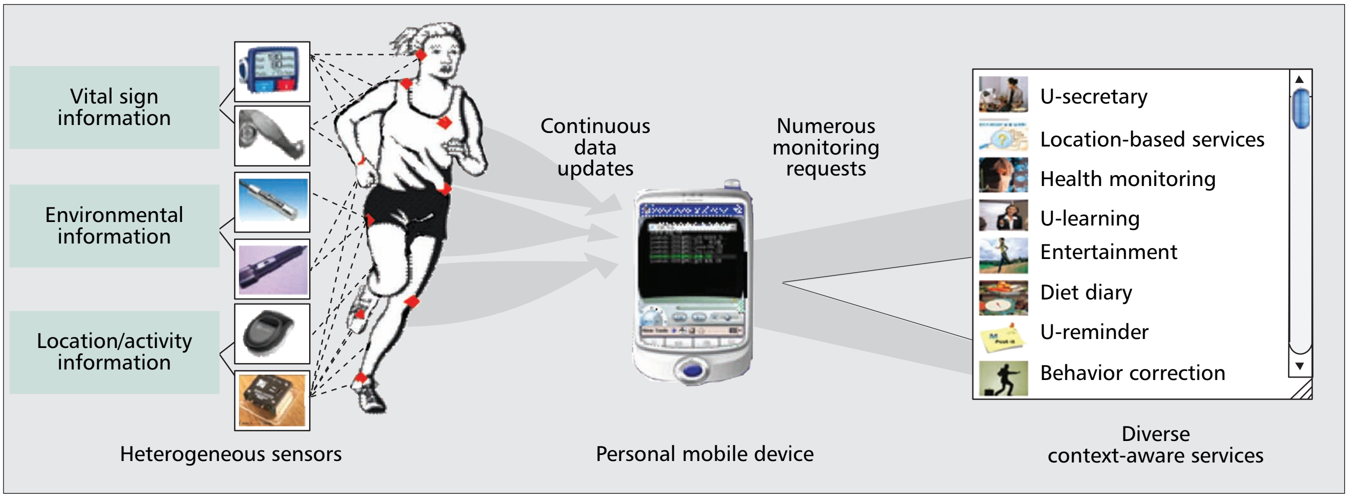
The proposed framework is a new and holistic approach for large-scale context monitoring and composition. Most context-aware services and systems mainly deal with a specific type of context [1–9]. Some handle multiple contexts, but they only provide limited context composition within a single context level [10–16]. Furthermore, only few systems support context monitoring on dynamic contexts. Google Earth [17] can be considered a commercial service based on several static contexts covering personal, regional, and global (e.g., user tags, street views, and satellite images). If Google includes richer and more dynamic contexts, and provides more advanced and proactive services that cover sensed data, Google Earth would evolve into a novel service requiring large-scale context monitoring and composition. In what follows, we introduce existing work for each context level and compare it with our framework.

Personal Context-Aware Applications and Systems

Availing of personal contexts, some applications and application-specific systems have been proposed (e.g., healthcare/medical [7], personal assistant [8], and activity/emotion-aware applications [9]). Each system mainly utilizes a specific personal context such as location, activity, and body signals, and concentrates on improving context recognition quality. Some middleware platforms [14–16] have been proposed to support multiple personal context-aware applications. They aim to hide the complicated issues related to context recognition, such as sensor data preprocessing, feature extraction, pattern recognition, context modeling, and specification. A key problem in real-world applications is that continuous context monitoring is required beyond one-time context recognition. Often, practically useful personal contexts are derived by continuously observing recognized contexts, their changes, and even complex patterns. PocketMon [18] takes an efficient context monitoring approach that continuously captures contexts to notice their changes. By utilizing continuity of contexts, it significantly reduces the overhead of repetitive context recognitions, thereby significantly improving the processing performance and energy efficiency.

Regional Context-Aware Services and Systems

In the literature of mobile computing, some regional location/traffic-aware systems have been proposed. Location-aware systems provide regional information to mobile users based on their current locations. for example, restaurant recommendations in i-channel [1], and taxi calling in NaviCall [2] and Google Ride Finder [3]. Traffic-aware systems [4] support shortest path navigation and intelligent tolling.



■ Figure 2. *Personal context monitoring environment.*

Several sensor networks have been proposed for specific application domains, such as air pollution [5], fire [6] and structural health monitoring, industrial process monitoring, and retail management [19]. To realize these sensor networks, several technical requirements have arisen (e.g., scalable routing protocol, efficient in-network processing mechanism, and quality of information). Most existing work has tried to address the above requirements, but the composition of diverse regional contexts has hardly been dealt with. The composition will greatly enhance the richness and comprehensiveness of contexts, thereby spawning a lot of advanced services. It will be enabled by HiperMon incorporated with plentiful sensor networks.

To provide a general framework for regional context-aware services, several middleware platforms have been developed. They mainly addressed technical challenges such as context abstraction in ContextSphere [10], service development in MISSA [11], context data archival in DCF [12], and context browsing in SenseWeb [13]. On the other hand, HiperMon focuses on providing context monitoring and composition, which are very essential to enable *proactive* and *enriched* context-aware services. The continuous monitoring and flexible composition of HiperMon enable understanding of dynamic and sophisticated behaviors of people, traffic patterns, and environmental status of regions.

Conventional database management systems (DBMSs) can hardly support data processing and management for massive regional contexts. They employ a *store-and-evaluation approach*, storing data in disks or memory before processing. Due to huge volume and rapid updates of context data, DBMSs would consume a significant amount of time to store the data, easily failing to provide real-time responses. Recently developed data stream management systems (DSMSs) such as Aurora [4] and TelegraphCQ [19] support high-performance monitoring on data streams generated from sensors. DSMSs mainly deal with relational semantics such as join and aggregation. However, other types of complex semantics such as event detection and composition are simultaneously required for rich and comprehensive context monitoring. In HiperMon these semantics are newly identified and effectively supported by high-performance processing mechanisms.

Global Context-Aware Services and Systems

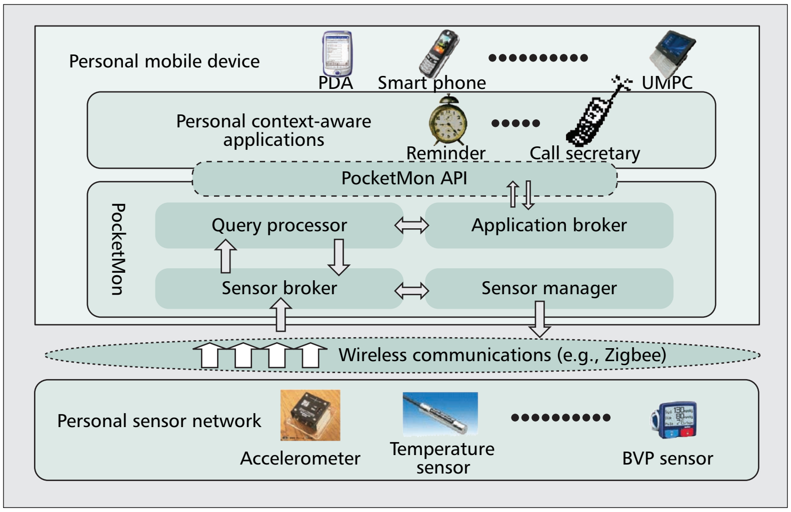
Distributed data stream processing systems (DSPSs) [1] have proposed relational query language and processing mechanisms that potentially can be used for global context processing over the Internet. DSPSs utilize distributed data processing techniques while considering network awareness. Since DSPSs process each requested query separately, we can hardly expect shared data processing, which could potentially reduce redundant data delivery. Application-level multicast (ALM) [20, 21] has been proposed for scalable data delivery over the Internet. It can greatly reduce redundant data delivery through shared delivery. In ALM all streaming data are delivered to numerous consumers. While such delivery is necessary in their target application, video on demand (VOD), selective data delivery is required in most global context-aware services; consumers want to receive only the context data of interest. Content-based publish/subscribe systems (CPSs) [22] are proposed to enable selective data delivery. All subscribers can receive only data in which they are interested. In particular, CPSs support in-network data selection by filtering out unwanted data in forwarding nodes. Thus, this approach does not impose heavy processing and delivery overhead to edge nodes. Unfortunately, CPSs have not been considered to support other delivery requirements such as end-to-end network delay, whereas EGI did. EGI is designed to provide a novel overlay architecture well suited to diverse delivery requirements. Such a feature is crucial since global context-aware services commonly require timely acquisition of global contexts for proactive and agile decision making.

System Components

PocketMon

Personal contexts reflect diverse detailed situations different individuals may face. To capture such situations, heterogeneous wireless sensors using different protocols and data formats will be deployed around individuals, and a large amount of data from the sensors will be used. Moreover, complex and intelligent processing will be performed to analyze raw sensor data and compose personal contexts. Various data cleansing and feature extraction techniques such as fast Fourier transform (FFT) will be applied on raw sensor data. Sophisticated context recognition methods such as HMM, decision tree, and Bayesian classifiers will be further executed on feature data.

Seamless monitoring of such delicate personal contexts requires careful consideration of privacy and mobility issues. Opening sensitive private data to the public is not preferred. The data should be isolated in a user's mobile device and accessed only by the user or authorized people (e.g., granting access rights to patients' health monitoring data for doctors only). In terms of mobility, services should be stably provided to mobile users regardless of their weak connectivity conditions or network failures. Additionally, continuous network connection and bandwidth consumption may cost high net-



■ Figure 3. *Architecture of PocketMon.*

```
CONTEXT (location = library)
AND (activity = sleeping)
AND (time = evening)
ALARM F -> T
DURATION 120 days
```

■ Figure 4. *An example of personal context monitoring query (PCMQ): "alarm when a user starts to sleep in a library in the evening."*

work service fees. A server-based approach [14] that collects and processes all private data continuously cannot address these issues adequately.

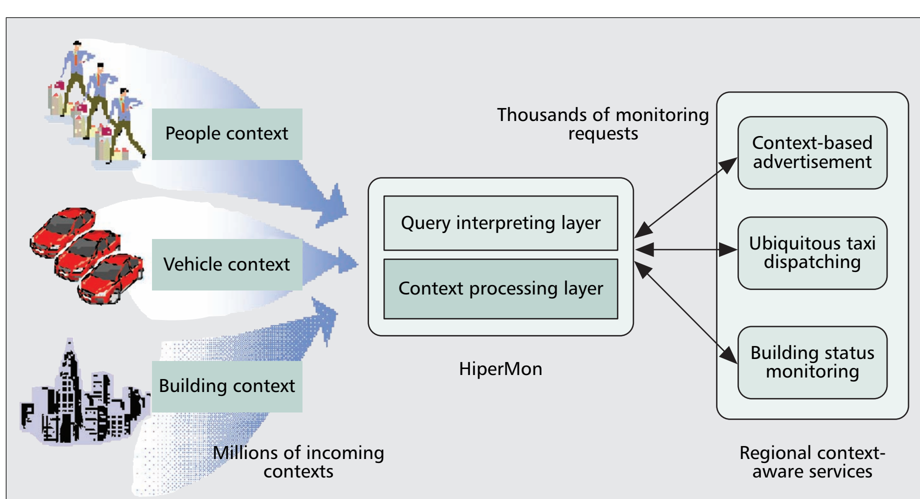
We develop PocketMon [18], a processing- and energy-efficient personal context monitoring platform running atop resource-limited mobile devices. The personal mobile devices will provide a number of context-aware services using data received from diverse and numerous sensors (Fig. 2). PocketMon enables developers to create such services by simply describing their contexts of interest without concern for the details of runtime environments. Figure 3 presents the overall architecture of PocketMon. Developers can intuitively express their requirements using a declarative query language, Personal Context Monitoring Query (PCMQ), shown in Fig. 4. They do not have to consider difficult issues of managing heterogeneous sensors and deriving contexts from sensor data. More important, PocketMon frees developers from performance and energy considerations. PocketMon significantly reduces computational overhead by exploiting the fact that the contexts of an individual often remain the same for a while. It avoids unnecessary context recognition logic and effectively accelerates repetitive computation. To reduce energy consumption, it dynamically identifies a small set of sensors sufficient to evaluate PCMQs and deactivates others to avoid unnecessary wireless data transmission.

HiperMon

We envision that regional contexts will be used as essential materials for public services, regional businesses, and even personal services. The services based on regional contexts give rise to new and critical challenges. First and foremost, we will confront the immense scale of regional contexts. Consider crowd movement monitoring in metropolitan cities. There would be up to ten million location contexts updated every a few seconds. Second, we should make provision for a huge number of monitoring requests. Consider a context-aware advertisement as an example. If each restaurant in a city issues a monitoring request, 10,000 monitoring requests would be executed concurrently and continuously. Third, real-time response will be crucial in most monitoring requests. In many cases the value of derived contexts rapidly deteriorates with increasing delay. Finally, regional contexts are generally complex, unstructured, and rarely studied. Thus, it is hard to easily describe the contexts. It is imperative to investigate well structured monitoring semantics for regional contexts. In the end, a well suited platform will be necessary to cope with such a large scale and complex semantics. It will greatly help service developers focus on developing their own service logics.

To address above challenges, we propose HiperMon [23] (Fig. 5), a high-performance massive context monitoring system. It is responsible for collecting numerous personal contexts, composing and monitoring the collected contexts in real time, thereby providing the contexts of interest to services effectively. HiperMon consists of two layers; the query interpreting layer is responsible for analyzing and decomposing diverse monitoring requests, while these requests are efficiently processed in the underlying context processing layer.

More important, HiperMon supports *rich monitoring semantics* and *high-performance massive context processing*. First of all, we devise Regional Context Monitoring Query (RCMQ), an SQL-like query language for regional context monitoring. It provides rich and comprehensive views on regional contexts, and enables intuitive monitoring expressions



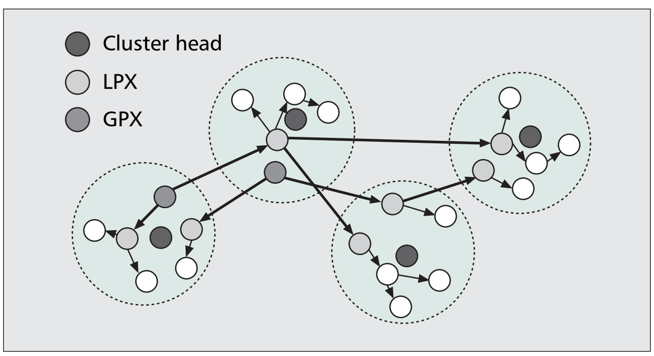
■ Figure 5. *Architecture of HiperMon.*

```
Q1: Find people who enter a given advertisement region and whose ages are under 20.

SELECT People.ID
FROM People, People-Info
WHERE (People.xlocation, People.ylocation) inward (xl, xh, yl, yh)
AND (People.ID = People-Info.ID)
AND (10 <= People-Info.age <20)
DURING infinite

Location stream schema: People<ID, xlocation, ylocation, timestamp>
Profile table schema: People-Info<ID, age, gender>
(xl, xh, yl, yh) is a given rectangular advertisement region.
Keyword inward expresses the situation where people are coming into a region.
DURING clause specifies how long the query will execute.
```

■ Figure 6. *An example of a regional context monitoring query (RCMQ).*



■ Figure 7. *EGI structure.*

on the contexts. We believe that the monitoring semantics will be widely used for regional context-aware services. Figure 6 shows an example RCMQ that continuously monitors people to send out restaurant advertisements.

For high-performance massive context processing, HiperMon is equipped with several well crafted processing techniques. Basically, HiperMon employs a dataflow-like evaluation method, which overcomes the inefficiency of the existing store-and-evaluation method. Furthermore, we significantly accelerate successive query evaluations by developing a *stateful query index* [24]. The index enables shared and incremental processing of a large number of RCMQs. Once the index is built on registered RCMQs, only relevant queries are quickly searched for without unnecessary access to irrelevant queries. In addition, the lightweight feature of the index structure enables in-memory processing.

EGI

Composing global contexts inherently requires wide-area delivery of continuously generated context data. We observe that the delivery involves spatial and temporal constraints on context data. Consumers will specify data sources and value ranges of their interests (i.e., spatial constraints) to avoid unnecessary data transmission and processing. They will also specify data rates and delay requirements (i.e., temporal constraints). As context-aware services get popular, context sources may serve a large number of context consumers widely dispersed.

We propose EGI [25], a scalable delivery infrastructure to facilitate global context composition. It composes an overlay multicast network of EGI nodes widely deployed on the Internet. EGI nodes provide interfaces to data sources or consumers including HiperMon. They handle data delivery requests issued by global context-aware applications. In designing such a delivery infrastructure, a key challenge is to construct efficient delivery paths satisfying diverse requirements of numerous consumers in a scalable manner.

EGI provides a scalable path construction scheme based on clusters of EGI nodes; with clustering, the path construction process performs in two levels, within a cluster and between clusters. It effectively reduces the costs of network probing and path selection. EGI further employs the concept of a selective forwarding table (SFT) to deal with complicated spatial and temporal constraints. SFT in each node abstracts the complex delivery requirements of its child nodes on the paths down to consumers. It enables EGI nodes to efficiently filter incoming context data and maintain the overall delivery network to be efficient. Figure 7 briefly illustrates the structure of EGI. Nodes are grouped into clusters (dotted circles in the figure) based on their network proximity, where the global proxy (GPX), local proxy (LPX), and cluster head are responsible for inter- and intracluster path construction and cluster management, respectively.

Implementation and Applications

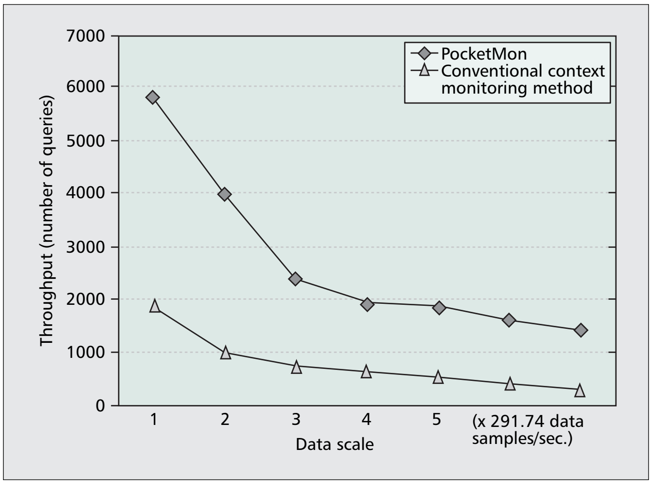
PocketMon

We have implemented a PocketMon prototype system running on a SONY UMPC with an Intel U1500 processor and a wearable device with a Marvell PXA270 processor both running Linux. The former targets powerful future mobile devices and the latter targets relatively resource-limited current mobile devices. Application developers used the prototype system and considered it effective for personal context composition and monitoring. Figure 8 and Fig. 9 show the demo of the developed applications. First, SympaThings in Fig. 8 controls nearby smart objects to adapt to a person's emotional contexts. For example, picture frames change pictures inside and lighting fixtures adjust their color. Second, Running Bomber in Fig. 9 is a ubiquitous game developed to give additional fun to treadmill runners. The runners wear small acceleration sensors on their wrist and pass a time bomb to others by shaking the hands.

We have also conducted experiments to evaluate the processing and energy efficiency of PocketMon. We collected raw

■ Figure 8. *SympaThings demo at Nextcom Show 2007.*

■ Figure 9. *Running Bomber demo.*



■ Figure 10. *Throughput measurement results.*

sensor data from the daily activities of a person. A student carried a UMPC for 12 hours in the campus with nine sensors presented in Table 1. The total data rate was 291.74 Hz. We synthetically generated PCMQs reflecting various monitoring conditions on different personal contexts. Assuming a resource-limited mobile environment such as smart phones, we scaled down the CPU frequency of a UMPC to 200MHz.

Figure 10 demonstrates the processing efficiency of PocketMon compared to a conventional context monitoring method, which repetitively performs context recognition upon data arrivals. Throughput is the maximum number of queries that can be handled without causing system overload. We synthetically increase the size of data workload by replicating the collected sensor data. At data scale k, the data rate becomes k times larger than data scale 1. Even with data scale 7, PocketMon can process about 1400 queries stably, i.e., 4.6 times larger than that of the conventional context monitoring method. It is a reasonably large throughput given the device's limited resources and the high rate of sensor data (2100 samples/s).

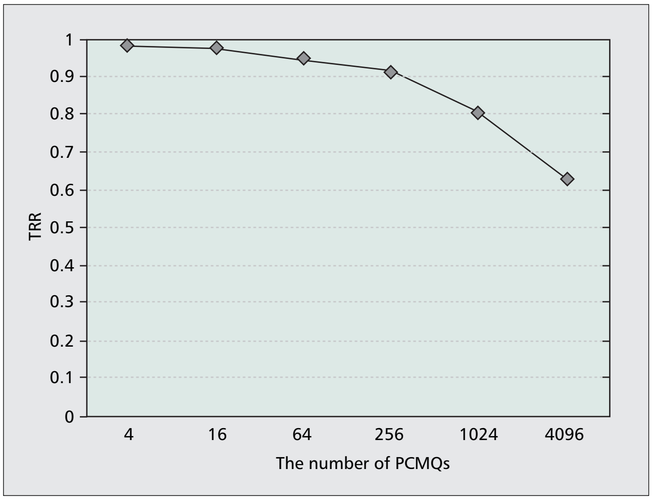
We also observed the energy efficiency in terms of transmission reduction ratio (TRR). TRR quantifies the amount of reduction in wireless data transmission, which is the main factor of sensors' energy consumption. Figure 11 shows that PocketMon reduces more than 90 percent of data transmissions when the number of PCMQs is fewer than 256. Even with 4096 queries, more than 60 percent of data transmissions are eliminated.

HiperMon

We have developed several regional context-aware services based on HiperMon. Figure 12 shows Ubiquitous taxiCab (UbiCab), an intelligent taxi dispatching service. UbiCab monitors stationary patterns of people and taxies in a city and dispatches available taxies to potential taxi-waiting passengers. Also, passengers can explicitly request a taxi satisfying their preferences such as closest distance, vehicle class, or driver's gender. It was demonstrated at the Mobile Media Research Center (MMRC) workshop held by SK Telecom and KAIST

Sensor	Sampling rate	Feature	Feature generation rate	Context type (no. of possible values)	Context value examples
Light sensor	0.72 Hz	Illumination	0.72 Hz	Light (7)	Dark, bright
Temperature sensor	0.36 Hz	Temperature	0.36 Hz	Temperature (8)	Cool, hot
Humidity sensor	0.18 Hz	Humidity	0.18 Hz	Humidity (6)	Dry, humid
Three 2-axial acceleration sensors	48.08 Hz × 6	DC	4.808 Hz × 6	Activity (12)	Running, sitting
		Energy	4.808 Hz × 6		
GPS sensor	2 Hz	Longitude	2 Hz	Outdoor location (9)	CS building, east restaurant
		Latitude	2 Hz		
		Speed	2 Hz	Speed (5)	Walking, bicycling
		Direction	2 Hz	Direction (8)	North, west
S/W sensor (timer)	—	Time	0.1 Hz	Time (8)	Dawn, noon
S/W sensor (indoor location)	Manual input	Indoor location	1 Hz	Indoor location (12)	1st floor lobby, room 2432

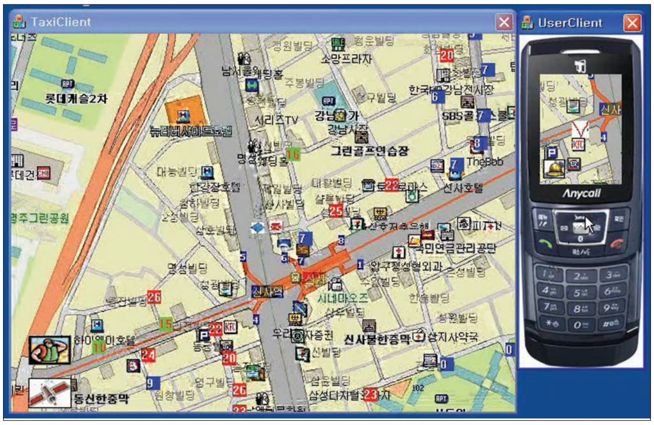
■ Table 1. *Sensor and context profiles.*



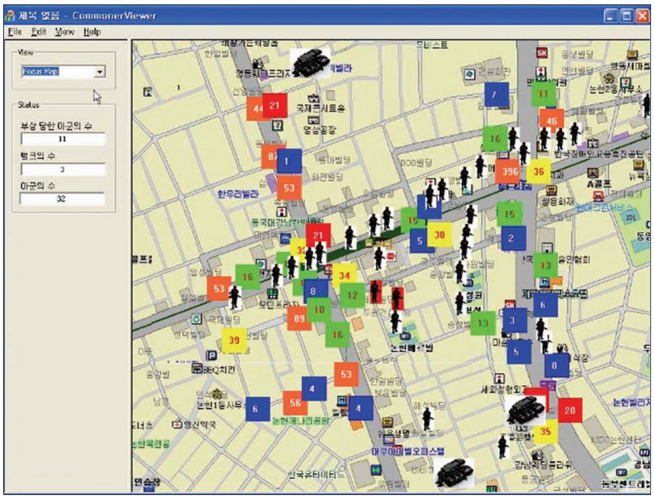
■ Figure 11. *TRR measurement results.*

Parameter	Value
Road network size	30 km × 30 km
No. of nodes in a road network	2400
People's speed	7 km/h
Taxi's speed	20 km/h
People's pause time	180 s
People's context reporting period	30 s
Taxi's context reporting period	10 s

■ Table 2. *Parameters for context stream generation.*



■ Figure 12. *UbiCab service.*



■ Figure 13. *U-BattleWatcher service.*

in February 2007. Figure 13 shows U-BattleWatcher, an integrated commander support service in battlefields. It collects diverse context from battlefields and carefully monitors collected battlefield contexts (e.g., our soldier's location and physical status, as well as brief distribution of enemy forces). Furthermore, it recommends the most strategic and appropriate commands in given situations. It was demonstrated at a Defense Software Research Center (DSRC) workshop held by KAIST and Agency for Defense Development (ADD) in Korea in November 2007.

We demonstrated the performance benefits of HiperMon through extensive performance studies. For the experiment, we developed a context stream generator and an RCMQ generator according to the UbiCab scenario. Table 2 shows parameters of context stream generation. In order to test our prototype under diverse and realistic city environments, we synthesized three different workload classes: heavy, normal, and light traffic conditions. Each workload represents the traffic at rush hour, day time, and night time, respectively. Table 3 shows the details of each workload class in a unit area of 1 km^2. The experiment was performed on a Fedora Core 5 server with 1.4 GHz Core Duo CPU and 4 GB memory.

Table 4 shows the maximum throughput for each workload class. In terms of covered area, HiperMon can cover up to about 190 km^2 under light traffic conditions. This shows that HiperMon is scalable enough to monitor a mid-size city. Note that HiperMon outperforms existing DSMS [4] by an order of magnitude. That is, HiperMon processes about 33,000 tuples/s,

Workload parameters	Traffic condition (per 1 km^2)		
	Heavy	Normal	Light
No. of taxis	100	100	100
No. of people	20,000	10,000	5000
No. of taxi calls/min	8	2	1.5

■ Table 3. *Parameters of three workload classes.*

	Light	Normal	Heavy
No. of taxis	18,769	9604	5041
No. of people	938,450	960,400	1,008,200
No. of taxi calls/min	281	192	403
Coverage (km^2)	187.7	96.0	50.4
Avg. no. of streams/s	31,900	32,300	33,800

■ Table 4. *Maximum throughput of HiperMon.*

whereas the DSMS processes only 2000 tuples/s. Moreover, HiperMon consumes only a small amount of main memory (less than 50 Mbytes) by exploiting lightweight data structures.

EGI

We have prototyped core functions of EGI such as cluster and path management. Each EGI node can be configured with processing capacity, available bandwidth, and clustering and path construction policies. We studied the performance of EGI in terms of bandwidth consumption, control message overhead, latency, and delivery hop counts, and observed that our scheme has a certain level of scalability and performance. In particular, EGI has shown significant reduction in network bandwidth consumption compared to application-level multicast.

Conclusion

We present the hierarchical context monitoring and composition framework for future context-aware services. The scale and complexity of the services are too challenging to successfully realize them. We notice the increasing importance of architectural supports to handle the challenge effectively. As the first exploration, we designed the framework for context monitoring and composition and developed the architectural components. The framework addresses critical requirements (i.e., context composition and monitoring scalability). The hierarchical structure and corresponding components fully support horizontal and vertical context composition. Moreover, highly scalable monitoring is achieved by layered context processing, localization of the processing overhead, and processing optimization in each layer. We prototyped core components and interesting context-aware applications, and demonstrated them in public exhibitions and research workshops.

Acknowledgment

This work was supported in part by the Ubiquitous Fashionable Computer (UFC) project, SoC Initiative for Ubiquity and Mobility (SoCium), Defense Software Research Center (DSRC), and Mobile Media Research Center (MMRC) at KAIST. We also thank the anonymous reviewers for their valuable comments. Our special thanks go to our shepherd, Dr. Lakshman Krishnamurthy, for his valuable suggestions to improve the quality of this article.

References

[1] T. Nakatsuru, K. Murakami, and H. Sakai, "Context-Aware Information Provision to The Mobile Phone Standby Screen," *Proc. MDM 2006*.
[2] NaviCall, http://navicall.co.kr/navi_call/navicall.jsp
[3] Google Ride Finder, http://labs.google.com/ridefinder
[4] D. J. Abadi *et al.*, "Aurora: A New Model and Architecture for Data Stream Management," *VLDB J.*, vol. 12, no. 2, Aug. 2003, pp. 120–39.
[5] Vanderbilt University, Mobile Air Quality Monitoring Network (MAQUMON) project, http://www.isis.vanderbilt.edu/maqumon/
[6] I. Hwang, Q. Han, and A. Misra, "MASTAQ: Middleware Architecture for Sensor Applications with Statistical Quality Constraints," *Proc. PERCOMW 2005*.
[7] J. E. Bardram, "Applications of Context-Aware Computing in Hospital Work Examples and Design Principles," Proc. SAC 2004.
[8] T. Sohn *et al.*, "Place-Its: A Study of Location-Based Reminders on Mobile Phones," *Proc. UbiComp 2005*.
[9] L. Bao and S. S. Intille, "Activity Recognition from User-Annotated Acceleration Data," *Proc. Pervasive 2004*, P. Pietzuch *et al.*, "Network-Aware Operator Placement for Stream-Processing Systems," *Proc. ICDE 2006*.
[10] N. Cohen *et al.*, "iQueue: A Pervasive Data-Composition Framework," Proc. MDM 2002 (http://www.research.ibm.com/cxs/)
[11] S. Kang *et al.*, "MISSA: Design and Implementation of a Middleware for Easy Development and Provision of Stream-based Services" KAIST Technical Report CS-TR-2006-253, 2006.
[12] K. Cho *et al.*, "DCF: An Efficient Data Stream Clustering Framework for Streaming Applications," *Proc. DEXA 2006*.
[13] Microsoft Research, SensorWeb project, http://research.microsoft.com/nec/senseweb/
[14] P. Fahy and S. Clarke, "CASS — a Middleware for Mobile Context-Aware Applications," *Proc. MobiSysW 2004*.
[15] D. Salber, A. K. Dey, and G. D. Abowd, "The Context Toolkit: Aiding the Development of Context-Enabled Applications," *Proc. CHI 1999*.
[16] O. Riva, "Contory: A Middleware for the Provisioning of Context Information on Smart Phones," *Proc. Middleware 2006*.
[17] Google, Google Earth, http://earth.google.com/
[18] S. Kang *et al.*, "SeeMon: Scalable and Energy-efficient Context Monitoring Framework for Sensor-rich Mobile Environments," *Proc. MobiSys 2008*.
[19] S. R. Madden *et al.*, "Continuously Adaptive Continuous Queries over Streams," *Proc. SIGMOD 2002*.
[20] CMU, End System Multicast (ESM) project, http://esm.cs.cmu.edu/
[21] J. Jannotti *et al.*, "Overcast: Reliable Multicasting with an Overlay Network," *Proc. OSDI 2000*.
[22] Univ. CO, SIENA project, http://serl.cs.colorado.edu/~carzanig/siena/
[23] B. Kim *et al.*, "HiperMon: A High Performance Massive Context Monitoring Framework for Wide-Area Context-Aware Services," KAIST tech. rep. CS-TR-2008-284, 2008.
[24] J. Lee *et al.*, "BMQ-Index: Shared and Incremental Processing of Border Monitoring Queries over Data Streams," *Proc. MDM 2006*.
[25] S. Jo *et al.*, "SATI: Scalable and Traffic Efficient Data Dissemination Infrastructure for Sensor-Based Distributed Information Services," KAIST tech. rep. CS-TR-2006-265, 2006.

Biographies

KYUNGMIN CHO (kmcho@nclab.kaist.ac.kr) is a Ph.D. student at the Korea Advanced Institute of Science and Technology (KAIST), Daejeon. He received a B.S. degree in electrical engineering from KAIST in 2001, and an M.S. degree in computer science from KAIST in 2003. His research interests lie in ubiquitous computing, data stream processing systems, distributed systems, and overlay networks.

INSEOK HWANG (saber@nclab.kaist.ac.kr) is a Ph.D. student at KAIST. He received B.S. and M.S. degrees in electrical engineering from KAIST in 2001 and 2002, respectively. His research interests lies in large-scale ubiquitous computing systems and distributed Internet computing systems.

SEUNGWOO KANG (swkang@nclab.kaist.ac.kr) is a Ph.D. student in computer science at KAIST. He received a B.S. degree in electrical engineering from Yonsei University, Seoul, Korea, in 2000, and an M.S. degree in electrical engineering from KAIST in 2002. His research interests include mobile and ubiquitous computing such as system architecture for context-aware applications, context-aware service and interaction design, and a mobile context monitoring framework.

BYOUNGJIP KIM (bjkim@nclab.kaist.ac.kr) is a Ph.D. student at KAIST. He received B.S. and M.S. degrees in electrical engineering and computer science from KAIST in 2002 and 2004, respectively. His research interests include mobile and ubiquitous computing, Internet computing, and data stream processing.

JINWON LEE (jcircle@nclab.kaist.ac.kr) is a Ph.D. candidate at KAIST. He received a B.S. degree in electrical engineering from KAIST in 2001 and an M.S. degree in computer science from KAIST in 2003. His research interests include mobile/ubiquitous computing, data stream processing systems, peer-to-peer overlay networks, and high-performance Internet cache systems. He received the best paper award at the 7th International Conference on Mobile Data Management, 2006.

SANGJEONG LEE (peterlee@nclab.kaist.ac.kr) is a Ph.D. candidate at KAIST. He received a B.S. degree in physics from KAIST in 1999 and an M.S. degree in electrical engineering from KAIST in 2001. His research interests lie in Internet technologies, mobile and ubiquitous applications and systems, and system support for context awareness and future games.

SOUNEIL PARK (spark@nclab.kaist.ac.kr) is a Ph.D. student at KAIST. He received a B.S. degree in computer science from Soongsil University, Seoul, Korea, in 2004, and an M.S. degree in computer science from KAIST in 2007. His research interests lie in Internet service and technologies, and ubiquitous computing systems.

YUNSEOK RHEE (rheeys@hufs.ac.kr) is an associate professor, School of Electronics and Information Engineering, Hankuk University of Foreign Studies, Yongin, Korea. Before joining the faculty he worked at Systems Engineering Research Institute, Korea, as a research staff member from 1988 to 1994. He received a B.S. degree in computer science and statistics from Seoul National University in 1988, and his Ph.D. in computer science from KAIST in 1999. His research interests include distributed computing, embedded systems, and Internet technologies.

JUNEHWA SONG (junesong@nclab.kaist.ac.kr) is an associate professor, Department of Computer Science, KAIST. Before joining KAIST, he worked at IBM T. J. Watson Research Center, Yorktown Heights, New York, as a research staff member from 1997 to 2000. He received his Ph.D. in computer science from the University of Maryland at College Park in 1997. His research interests lie in mobile and ubiquitous computing systems Internet technologies such as intermediary devices, high-performance Web serving, electronic commerce, and distributed multimedia systems.