

3<sup>rd</sup> March 2011, a detailed paper will be appearing in CACM soon.



# MobiCon: Next Generation Mobile and Ubiquitous Platforms

## A Mobile Context Monitoring Platform for Sensor-rich Dynamic Environments

Seungwoo Kang, S.S. Iyengar, Youngki Lee, Jinwon Lee, Chulhong Min, Younghyun Ju, Taiwoo Park, Yunseok Rhee, Junehwa Song.

Korean Advanced Institute of Science and Technology (KAIST) and Louisiana State University (LSU)

### TECHNOLOGY

The research driven technology presents MobiCon, a middleware platform for developing human activity applications in an energy efficient manner. Such application infer the user's patterns of activity by processing measurements streams collected by sensors placed on or around the user's body which are connected by a personal area network (PAN).

The main insight behind MobiCon is that instead of sensing all the sensors all the time, the system determines the sensors that are required by the union of applications that are currently active.

MobiCon supports writing human activity applications in the form of predicates (e.g. "location == 'library'") connected by logic operators that are supposed to run for a specified duration of time. MobiCon decomposes these high-level programs into a set of requests to acquire sensor streams whose 'values' are periodically re-evaluated.

MobiCon is an initial attempt to provide an active resource orchestration system, recognizing the PAN-scale sensor-rich mobile platform as a common underlying computing platform. Recently, many systems have been proposed for effective resource management of mobile devices [3][4][5][6][7] and sensors [8][9][10] comprising PAN. They are mostly designed to manage resources, especially battery in most cases, for applications on a single computing device. Such device-centric resource management, however, can hardly be utilized in our target environment, in which multiple sensors and a mobile device cooperatively serve multiple applications.

### RATIONALE

The challenge in this new environment is that the platform should simultaneously support a number of

applications with highly scarce and dynamic resources. Greedy and injudicious resource use would significantly aggravate contention among multiple applications and accelerate skewed use of some specific devices. This can also lead to reduced overall system capacity. Especially, we note that most of the devices have highly constrained resources, often less than the capacity required for a context monitoring. More challenging, availability of such resources dynamically changes due to their wearable forms and mobility of users. Sensors join or leave the platform frequently as users may take off a sensor-equipped smart watch [13], or enter a new sensor-embedded smart office. Also, resource usage by running applications or environmental factors such as interference continuously affects the resource availability.



Sensor-rich mobile environment for mobile context computing

It is almost impossible for each application to address the challenges without system-level supports. First, context monitoring entails very complex processing, and often

3<sup>rd</sup> March 2011, a detailed paper will be appearing in CACM soon.

involves burdensome device-level exploration [11][12]. Second, it is more challenging to ensure applications' steady running under highly scarce and dynamic resource availability. To achieve this, each application should efficiently share the constrained resources with other applications. Without system-level support, however, each individual application has an extremely limited view of the existence or resource uses of other applications, and further cannot negotiate with the concurrent applications for coordinated and efficient resource utilization. Thus, system-level support is compelling for the effective and coordinated use of scarce computing resources among multiple applications.

In this technology, we propose a novel mobile platform, *MobiCon* for the PAN (Personal Sensor Network) -scale sensor-rich mobile computing platform. Actively interplaying between context-aware applications and scarce dynamic resources, *MobiCon* enables the mobile platform to host a number of applications stably, exploiting its full resource capacity in a holistic manner. More specifically, it helps applications efficiently share resources and processing as much as possible. It also seamlessly adapts the applications to dynamic resource availability by resolving resource contention between applications or selecting the best processing plan according to the resource availability at that time. In addition, *MobiCon* can apply a system-level policy such as energy optimization with an integrated view of the total resource uses and status. With such system supports, applications become capable of providing mobile users with seamless, long-running high-quality service under dynamic circumstances with scarce resources. This semantics is different from conventional context recognition, which identifies the current context. Once a change is identified, it is not necessary to recognize the context redundantly as long as it remains unchanged. The context monitoring often involves multi-step complex operations such as feature extraction and context recognition distributed across the mobile and sensor devices at the same time. However, it is quite difficult for individual applications to perform complicated context monitoring process on their own, especially over the shared devices with highly limited resources. To effectively support context monitoring, a common monitoring platform is essential.



Figure 2: Smartphones and sensor devices deployed for the prototype system

## DESIGN SCHEME

To design effective resource orchestration, we take an active resource use orchestration approach that actively finds out the best combination of resources for requested context monitoring under the current status of resources and applications. This is substantially different from an existing approach, passive resource use management [3][4][7][10]. A key feature of the active approach is to decouple application's logical resource demand from physical resource allocation. It enables postponing resource selection and binding until resources' availability is sufficiently explored. Under this active approach, applications do nothing but turn in high-level context specifications to the system, and comply with the system's decision on resource allocation. Most of mobile and sensor systems, on the other hand, have taken passive resource use management approaches based on application-driven decision in resource selection and binding. In such approaches, applications should explicitly specify the type and amount of resources as in resource ticket [10] and resource descriptor [3]. A system simply allocates the requested resources if available. If not, applications would change their resource use plans according to predefined code blocks, e.g., by trading off data or functional fidelity. However, these approaches impose huge burdens on the programmer, and their flexibility cannot be fully utilized in practice since it is almost impossible for applications to estimate dynamic resource status and prepare well adapted code blocks for all the cases.

*MobiCon* realizes the proposed approach as follows. First, it prepares multiple alternative resource use plans, each of which can process a high-level context from applications. Such alternatives result from the diversity of semantic translation. That is, a context can be derived by utilizing many different sensing modalities as well as feature extraction and classification methods. For instance, when a context quality required by an application is conditionally tolerable under a particular situation, a 'running' context is monitored with diverse methods, e.g., utilizing DC and energy features from acceleration data [12], or statistical features from GPS location data. Second, at runtime, it dynamically adapts a processing plan to reflect resource availability, running applications' requests, system-level policies in a holistic manner. Such flexibility and adaptation enable *MobiCon* to support multiple context-aware applications, extending their running time while balancing their resource usage, in environments with highly limited and dynamic resources.

## HARDWARE FEATURES

We deployed the prototype system on various types of mobile devices and sensors. First, we have deployed the prototype on two different mobile devices, (1) Ultra Mobile PC (UMPC), SONY VAIO UX27LN with Intel® U1500 1.33 GHz CPU and 1GB RAM, and (2) a smart phone, Nokia N96 with Dual ARM9 264MHz processor and 128MB RAM. Second, we have incorporated various wireless sensors that have been widely adopted for context-aware applications (See Fig. 10 for sensor details). Considering the wearability and controllability, we mainly use eight of USS-2400 sensor

3<sup>rd</sup> March 2011, a detailed paper will be appearing in CACM soon.

Sensor	Sensor location (sensor ID)	Sampling rate	Feature	Feature generation rate	Context type (# of possible values)	Context value examples
Four 2-axis accelerometer	Right(3) wrist, Right thigh(4), Left wrist(5), Waist (6)	50Hz x 8	DC, Energy, RMS, MAD, Percentile	0.78Hz x 8	Activity (4)	Run, Sit, Walk, Stand
Two light Sensors	Body(2), Space(102)	0.78Hz x 2	Illumination	0.78Hz x 2	Light (7)	Dark, Bright
Two temperature Sensors	Body(1), Space(101)	0.78Hz x 2	Temperature	0.78Hz x 2	Temperature(8)	Cool, Hot
Two humidity sensors	Body(1), Space(101)	0.78Hz x 2	Humidity	0.78Hz x 2	Humidity (6)	Dry, Humid

Figure 4. Sensor, feature, context profile used in the prototype

nodes (MicaZ clone), i.e., four 2-axis accelerometers, two light, and two temperature/humidity sensors. They are equipped with Atmega 128L MCU, CC2420 RF transceiver supporting 2.4GHz band ZigBee protocol, and TinyOS as an operating system. To provide communication between the mobile device and sensors, we attach one base sensor node to the mobile device. The node receives sensor data from other sensor nodes and forwards the data to a mobile device. Also, it transmits control messages to the sensor nodes on behalf of the mobile device.



Figure 3. Hardware setup

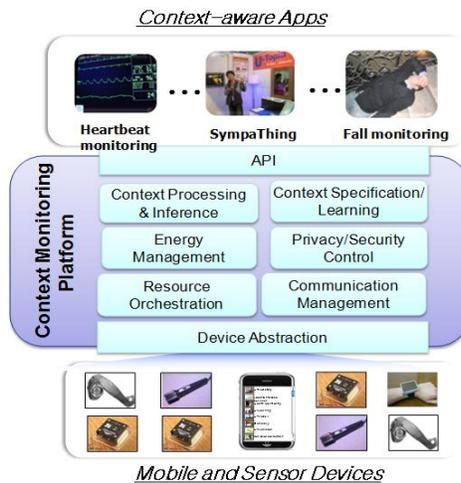


Figure 5 : Context Aware Applications

## PROTOTYPE

We have implemented the MobiCon architecture as a prototype system. First, we implemented the architecture of a mobile device in two platforms: (1) standard C/C++ over Linux, (2) Open C/C++ over S60 SDK and Symbian OS. Their total lines are about 13,000. Second, the sensor architecture is implemented in NesC on top of TinyOS 1.1.11. The total lines are about 2,300.

Currently, the plan processor in the mobile-side architecture includes eight feature extraction modules (See Fig. 10 for feature details). We used kiss\_fft [18], a Fast

Fourier Transform library, to derive frequency-domain features. It also provides a recognition module implementing a decision tree algorithm. To generate diverse plans to monitor activity contexts, we combine diverse feature and sensor sets. As feature sets, we use frequency-domain features (e.g., DC, Energy) and time-domain statistical features (e.g., RMS, PRC, MAD). For sensor sets, we use all combinations of sensors on the left wrist, right wrist, right thigh, and waist. We trained the activity contexts via annotation-based learning [12]. The learning was done with C4.5 decision tree by Weka, a Java-based open source machine learning tool [14]. We implemented feature extractors on sensor nodes to offload feature extraction tasks. We used a highly optimized avr-fft library which is written in an assembly language for FFT computation on sensors.

## Performance Results

Experimental results on the prototype show that the system achieves a superior level of scalability and energy efficiency. The processing time can be reduced by orders of magnitude compared to what the present technology offers, and data transmission can be reduced by the order of 50 percent.

## References

- [1] E. Miluzzo, N.D. Lane, K. Fodor, R.A. Peterson, H. Lu, M. Musolesi, S.B. Eisenman, X. Zheng, A.T. Campbell, "Sensing Meets Mobile Social Networks: The Design Implementation and Evaluation of the CenceMe Application," Proc. of SenSys, 2007.
- [2] S.B. Eisenman, E. Miluzzo, N.D. Lane, R.A. Peterson, G.-S. Ahn, A.T. Campbell, "The BikeNet Mobile Sensing System for Cyclist Experience Mapping," Proc. of SenSys, 2007.
- [3] B.D. Noble, M. Satyanarayanan, D. Narayanan, J.E. Tilton, J. Flinn, and K.R. Walker, "Agile Application-Aware Adaptation for Mobility," Proc. of SOSP, 1997.
- [4] J. Flinn and M. Satyanarayanan, "Energy-aware Adaptation for Mobile Applications," Proc. of SOSP, 1999.
- [5] D. Narayanan and M. Satyanarayanan, "Predictive Resource Management for Wearable Computing," Proc. of MobiSys, 2003.
- [6] H. Zeng, X. Fan, C. S. Ellis, A. Lebeck, and A. Vahdat, "ECOSystem: Managing Energy as a First Class Operating System Resource," Proc. of ASPLOS, 2002.
- [7] X. Liu, P. Shenoy, and M.D. Corner, "Chameleon: Application-Level Power Management," IEEE Trans. on Mobile Computing, vol. 7, no. 8, Aug. 2008.
- [8] A. Lachenmann, P. J. Marrón, D. Minder, and K. Rothermel, "Meeting Lifetime Goals with Energy Levels," Proc. of SenSys, 2007.
- [9] J. Sorber, A. Kostadinov, M. Garber, M. Brennan, M.D. Corner, E.D. Berger, "Eon: A Language and Runtime System for Perpetual Systems," Proc. of SenSys, 2007.
- [10] K. Lorincz, B. Chen, J. Waterman, G. W. Allen, and M. Welsh, "Resource Aware Programming in the Pixie OS," Proc. of SenSys, 2007.
- [11] S. Kang, J. Lee, H. Jang, H. Lee, Y. Lee, S. Park, T. Park, and J. Song, "SeeMon: Scalable and Energy-efficient Context Monitoring Framework for Sensor-rich Mobile Environments," Proc. of MobiSys, 2008.
- [12] L. Bao and S.S. Intille, "Activity recognition from user-annotated acceleration data," Proc. of Pervasive, 2004.
- [13] U. Maurer, A. Smailagic, D.P. Siewiorek, and M. Deisher, "Activity Recognition and Monitoring Using Multiple Sensors on Different Body Positions," Proc. of BSN, 2006.
- [14] Weka 3: Data Mining Software in Java. <http://www.cs.waikato.ac.nz/~ml/weka/>

3<sup>rd</sup> March 2011, a detailed paper will be appearing in CACM soon.

- [15] P. Korpipää, J. Mäntyjärvi, J. Kela, H. Keränen, and E.-J. Malm, "Managing Context Information in Mobile Devices," IEEE Pervasive Computing, 2003.
- [16] A. Haag, S. Goronzy, P. Schaich, and J. Williams, "Emotion Recognition Using Bio-sensors: First Step towards an Automatic Systems," LNCS 3068, 2004.
- [17] V. Kostov and S. Fukuda, "Emotion in User Interface, Voice Interaction System," Proc. of IEEE International Conference on Systems, Man, and Cybernetics, 2000.
- [18] Kiss FFT. <http://kissfft.sourceforge.net/>
- [19] C. Becker, G. Schiele, H. Gubbels, and K. Rothermel, "BASE: a Micro-Broker-Bases Middleware for Pervasive Computing," Proc. of PerCom, 2003.
- [20] C. Becker, G. Schiele, H. Gubbels, and K. Rothermel, "PCOM – a Component System for Pervasive Computing," Proc. of PerCom, 2004.
- [21] J. Mazzola Paluska, H. Pham, U. Saif, G. Chau, C. Terman, and S. Ward, "Structured Decomposition of Adaptive Applications," Proc. of PerCom, 2008.
- [22] H. Pham, J. Mazzola Paluska, U. Saif, C. Stawarz, C. Terman, and S. Ward, "A Dynamic Platform for Runtime Adaptation," Proc. of PerCom, 2009.