Smart CAPs for Smart Its - Context Detection for Mobile Users

FLORIAN MICHAHELLES[1], MICHAEL SAMULOWITZ[2]

[1]*Institute for Scientific Computing, ETH, CH-8092 Zuerich*

Florian.Michahelles@inf.ethz.ch
Phone ++41 (0)1 63 20434
Fax ++41 (0)1 63 21596

[2]*Corporate Technology Siemens AG; Otto-Hahn-Ring 6, D-81730 Munich*

Michael.Samulowitz@mchp.siemens.de
Phone ++49 (0)89 636 50564

ABSTRACT

Context detection for mobile users plays a major role for enabling novel, human-centric interfaces. For this, we introduce a context detection scheme for disseminated, computer empowered sensors, referred to as *Smart-Its* [7]. Context-detection takes place without requiring any central point of control, and supports push as well as pull modes. Our solution is based on an in-network composition approach relying on so-called *smart context-aware packets* (sCAPs). sCAPs travel thru a sensor network governed by an enclosed *retrieving plan*, specifying which sensors to visit for gaining a specific piece of context information. For enhanced flexibility, the retrieving plan itself may be dynamically altered in accordance to past sensor readings.

Keywords: mobile computing, context-aware computing, context representation, data-centric networking, sensor fusion

1. Introduction

Emerging mobile computing devices provide their users with a variety of services. In such environments powerful computing services can be accessed through small devices. Interaction with mobile devices involves a number of research challenges. One challenge is discovering computing services and resources available in the user's current environment. This lead to discovery protocols such as JINI [1], IETF SLP [2] or SDP[3]. Another challenge is the selection and execution of services respecting the user's context. As shown in [4] context impacts mobile HCI, [5] states the shift from explicit to implicit HCI by reverting to context. In [6] we argued that just offering services is not enough, services should be aligned to a user's task which is constrained by the user's context. How can it be detected? Accordingly, a third challenge is to reveal the user's current context. In this paper we describe an approach of exploiting manifold sensors available in a future environment for revealing the user's context. We will present a uniform communication scheme that allows explicit context detection requests, initiated by the mobile user, (pull) as well as event-triggered context evaluation, initiated at by sensor's percepts (pull). Our document-based approach – *smart context-aware packets* (sCAPs) - allows to capturing feature data in self-organized sensor networks without the need of central points of control. Further, sCAPs provide adaptive sensor fusion based on an online changeable execution plan and parallel retrieving of various sensors.

2. Context Detection

A mobile user is given access to services available in his surrounding environment. In our approach we want to gain the user's context from sensors available in his current environment for implicit human computer interaction [5]. Our work mainly focuses on the communication scheme for retrieving sensors.

2.1 Requirements

We envision an environment with disseminated computing empowered sensors, so-called Smart-Its [7]. These Smart-Its are attached to day-to-day devices such as cups, tables, chairs etc.; they can be equipped with various sensors for perceiving temperature, light, audio, co-location, movement and so on. Further, our tiny devices are supplied with a wireless communication such as RF or Bluetooth [8]. An on board micro-controller provides computing power and enables simple feature calculation from the sensor's inputs. This features, e.g. loudness, brightness, speed, temperature etc., are described by discrete number values.

Additionally, the envisioned Smart-Its operate autonomously with no central point of control. Accordingly, there is no directory service giving information about the sensors available in the current environment. Instead, each device knows about its own sensing capabilities and can report this, if inquired, to its neighbors. Further, each Smart It unit has a global unique identifier, its capabilities, respectively the features it is capable to deliver, are also distinguishable by unique identifiers. The neighborhood of a single Smart It is constrained by the reach of wireless communication.

Further we assume local awareness of each single Smart It as proposed in [9] and time synchronization [10].

2.2 The Concept of Smart Context-Aware Packets (sCAPs)

In this section we introduce our concept of *smart context-aware packets* (sCAPs) that act as an interchanging format sensor-features among Smart-Its. The concept of sCAP shares some similarities with *context-aware packets* (CAP) [6] based on an document-based approach, because it also uses context for implicit addressing the packet's receiver. The sCAPs can be understood as prepared containers for collecting features from various sensors available in the environment. After creation a sCAP get injected into the local sensor network. Each Smart-It receiving a sCAP contributes to the required sensor features and forwards it to another Smart-It devices in its neighborhood. Accordingly, the sCAP gets filled with sensed information on its way through the environment. Combining the gained features stored in the sCAP allows each Smart It to make an assumption about the current context. Based on the knowledge it can forward this sCAP to an appropriate sensor for further investigation of the context. Thus, there is a permanent in network recalculation of the context, which allows continuous refinement of the assumption and adaptation of the sCAP routing plan.

As Figure 1 depicts sCAP document is organized into three parts: *retrieving plan*, *probable context*, *retrieving path*. The **retrieving plan** embodies the execution plan determining which sensors should be when involved into the context detection. It describes which types of sensors have to be queried for achieving the current context. Due to single sensors percepts this retrieving plan can be continuously refined at each receiving sensor unit, such that the detection process can adapt to the actual sensor inputs. The representation of the retrieving plan will be discussed in more detail in the subsequent section.

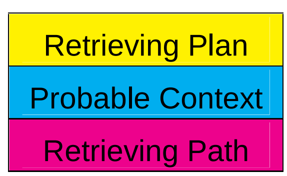

Figure 1 Smart Context-Aware Packet

As already mentioned, the **probable context** is an accumulation of feature values retrieved from several sensors. At each receiving unit the probable context can be recalculated by taking the retrieved features and new sensor input at the current unit into account. The probable context is simple represented by a list of feature already perceived. Here, each feature is described by following entries: *Feature ID, Feature value, Sensor type ID, Smart It ID, Sensor location, timestamp*. *Feature ID* determines the type of the feature, e.g. whether it is loudness, temperature, brightness etc.; *Feature value* is an actual number value*; Sensor type ID* defines the type of sensor the feature was gained from; *Smart It ID* is the uniform identifier for the platform the feature was sensed at; *Sensor location* and *timestamp* give the physical location and time of the feature was perceived at. The context results from these sensor measurements.

The **retrieving path** section of sCAPs simply provides a time ordered list of visited sensors. This aims to prevent loops, revisiting sensor units twice and to provide knowledge on the network topology for transmitting the context to the inquirer (mobile user).

2.3 Retrieving Plan

This section details the semantics and representation of the retrieving plan contained in sCAPs.

As mentioned above the retrieving plan defines the sensors to be visited for revealing the user's current context. Initially, the retrieving plan is given by an initial sCAP template setup for different context queries. However, what makes our approach promising is the possibility of continuous alteration and adaption and of the retrieving plan to single sensor measurements.

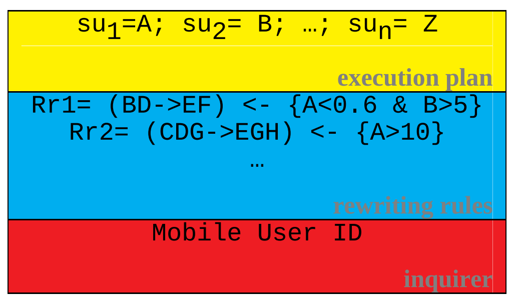

Figure 2 Retrieving Plan

As depicted in Figure 2 the retrieving plan is represented by three parts: *execution plan*, *rewriting rules* and *inquirer*. The **execution plan** contains the list of sensor units (su) to be involved into the context detection process. In particular the execution plan maintains a list of types of sensors, it does not identify certain sensors specifically. As the user is mobile, the environment always can change. Accordingly, it seems to be more reasonable to abstract from the actual sensor units but rather focus on sensor types.

Applying **rewriting rules** the execution plan can be altered at any Smart It the sCAP gets directed to. The rewriting rules incorporate the probable context currently available in this sCAP. Consequently, significant sensor inputs and combinations of those trigger rewriting rules to alter replace, add or delete sensor units from the execution plan. In our current prototype we do not allow changes of the rewriting rules themselves but this is feasible for future work.

The **Inquirer** field finally identifies the mobile user the context has to be delivered to.

2.4 Interaction schemes: Pull & Push

In our current design there are two ways of applying sCAPs for interacting with sensors. This section outlines these two interaction schemes: *pull* and *push*.

Pull embodies the active intervention of the user, such that the user, respectively a software agent acting on the user’s behalf, explicitly inquires a context update. The user initiates the context detection process by injecting an emtpy sCAP template with a default retrieving plan. Accordingly the sCAP gets round routed through sensor network by forwarding it through Smart Its’ neighbourhoods. The sCAP collects data from various sensors according to its retrieving plan. Meanwhile, all visited sensors’ identifiers are stored in the retrieving path section of the sCAP. Finally, the context detection process terminates as soon as the retrieving plan is performed; the detected context is reported to the inquirer (specified in the sCAP).

In contrast, **push** empowers sensors to collect data autonomously. Here, the sensors‘ percepts are used to trigger the context detection process. As soon as a sensor encounters a certain signal pattern it can initiate the context detection by injecting a sCAP to the sensor network by itself. After that, the detection process elapses as described in the last paragraph. As soon as the context has revealed the mobile user is informed on a publish-subscriber basis. Initially, certain sensors are empowered for autonomous sCAP creation by initial configuration. In our current prototype sensors can be enabled on timely or condition triggered basis. This configuration currently is set once during booting the system, online reconfiguration is postponed for future work.

Whereas context pull is used for determining the user’s context when the user explicitly needs the current context information, the push scheme is more appropriate for emerging events and interrupts, such as detecting rapid changes in the environments, states of emergencies or other sporadic coincidences.

So far, several sensors actually can perceive different inputs stemming from the same coincidence. Accordingly several sCAPs can be created at the same time detecting the same context. Up to now, we do not try to consolidate parallel context detection processes. Accordingly, the user might receive several sCAPs reporting the same context. This should be interpreted as very strong evidence for the reported contexts. However, future work should focus on these issue and merging strategies of sCAPs have to be found to consolidate related detection processes.

3. Conclusions

Envisioning interconnected future environments several challenges have to overcome. In order to provide convenient human computer interaction in changing environments the user's context should be taken into account and detected automatically by the system. In this work we focussed on context detection for mobile users in sensor monitored environments. Our work was heavily influenced by the Smart-Its project [7] assuming unobtrusively interconnected everyday objects with embedded sensors. We proposed *smart context-aware packets* as an intercommunication format among Smart Its for revealing mobile users' contexts. Our approach promotes sensor information exchange in sensor networks without central points of control. Further, our sCAPs documents foster adaptive fusion of feature inputs of manifold sensors.

4. Future Work

Future work will focus on the following six issues: (1) online reconfiguration of sensors for different push behaviors, (2) merging strategies for related sCAPs, (3) spreading strategies of sCAPs for parallelizing the context detection process, (4) online alterable rewriting rules, (5) semantical domains for sCAPs and finally tackling communication breakdowns [11].

References

[1] Jini(TM), 1998. http://java.sun.com/products/jini

[2] C. Perkins. Service Location Protocol White Paper, May 1997.

[3] S. Czerwinski, B. Zhao, T. Hodes, A. Joseph, and R. Katz. An Architecture for a Secure Service Discovery Service. In Proceedings of MobiCom '99, Seattle, WA, August 1999.

[4] Tom Rodden, Keith Cheverst, Nigel Davies and Alan Dix, *Exploiting Context in HCI Design for Mobile Systems,Proceedings of the First Workshop on Human Computer Interaction for Mobile Devices,* Department of Computing Science, University of Glasgow, U.K., 21-23 May 1998, *pp12-17.*

[5] Schmidt, *Implicit human-computer interaction through context*, 2nd Workshop on Human Computer Interaction with Mobile Devices. Edinburgh, Scotland, August 1999.

[6] M. Samulowitz, F. Michahelles, C. Linnhoff-Popien, *Adaptive Interaction for Enabling Pervasive Services,* MobiDE01, Santa Barbara, California, USA, May 2001.

[7] Smart-Its. http://www.smart-its.org

[8] The Bluetooth SIG. http://www.bluetooth.com

[9] Andreas Savvides, Chih-Chien Han, Mani Srivastava , *Dynamic Fine-Grained Localization in Ad-Hoc Networks of Sensors, Mobicom 2001*

[10] Jeremy Elson and Deborah Estrin , *Time Synchronization for Wireless Sensor Networks, Workshop on Parallel and Distributed Computing Issues in Wireless and Mobile Computing, San Francisco, California, USA. April 2001*.

[11] Tom Goff . Nael Abu-Ghazaleh, Dhananjay Phatak, Ridvan Kahvecioglu: *Preemptive Route Maintenance in Ad Hoc Networks, Mobicom 2001*