Contory: a Smart Phone Middleware Supporting Multiple Context Provisioning Strategies

Oriana Riva
Helsinki Institute for Information Technology
Cristiano di Flora
Nokia Research Center, Tampere

DYNAMOS Project (http://virtual.vtt.fi/virtual/proj2/dynamos/)

Ubiquity of Smart Phones

- We are 6.5 billion people
- First quarter of 2006 total mobile phone shipments reached 226.7 million units
- In 2006, **800 million** mobile phones will be sold
  - one person in eight on the planet
- 4 times PC or TV sale units

Advances in Smart Phones

- Greater on-board intelligence (software and hardware)
- Larger on-device data storage
- Better connectiviy to networks and other devices
- Larger, clearer, colourful screens
- Easier programming of the on-board intelligence
- While retaining their original attributes
  - Pocketable, reliable, inexpensive, simple
  - Mobile voice communications

Smart Phones in the Pervasive World

- "Anywhere, anytime connected"
- Many usages
  - Information service end point
  - Remoter controller
  - Hub to the Internet
  - ...
- One common requirement: context-awareness
  - Shift in focus: "user-centric model"
  - Device's execution context

Context Provisioning on Smart Phones

■ It is a challenging task:
- Energy-consuming function for energy-constrained mobile devices
- Integration of sensors in devices should not compromise the portability, usability, lifetime of everyday devices
- Device cost should be contained
- Some sensors are operative only in some specific environments (e.g., GPS)

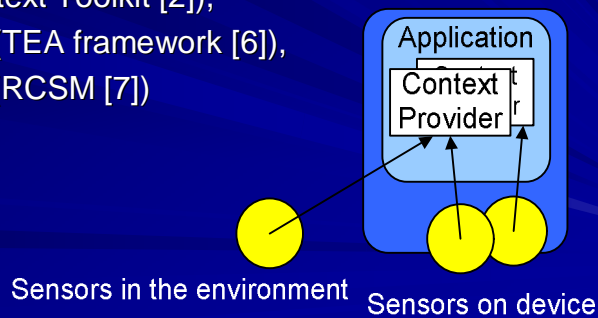**How to effectively support context provisioning on smart phones?**

Outline

■ Motivations

■ Context provisioning strategies

■ Contory Middleware

■ Evaluation

■ Current Status and Future Works
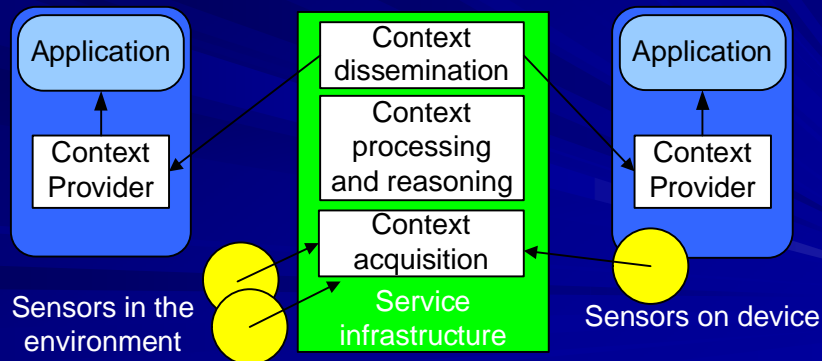
Internal Context Provisioning

Context providers:

- software components in charge of providing (i.e., sensing and processing) context information
- directly integrated into applications or organized in toolkits (Context Toolkit [2]), frameworks (TEA framework [6]), middleware (RCSM [7])

Application

Context Provider

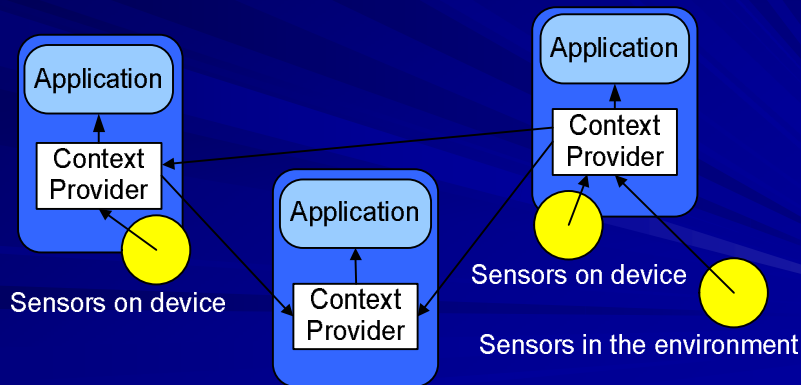Sensors in the environment   Sensors on device

External context provisioning: infrastructure-based

- Shared context servers (TCoS [4])
- Service infrastructures (Confab [5], JCAF [1])

Application

Context Provider

Context dissemination

Context processing and reasoning

Context acquisition

Service infrastructure

Application

Context Provider

Sensors in the environment   Sensors on device

External Distributed Context Provisioning: Infrastructure-less

- Abstraction of a distributed database
- Data are provided by context providers located on the nodes of a MANET

Application

Context Provider

Sensors on device

Application

Context Provider

Application

Context Provider

Sensors on device

Sensors in the environment

Outline

- Motivations
- Context provisioning strategies
- Contory Middleware
- Evaluation
- Current Status and Future Works

Contory Design Principles

- Transparency
- Flexible and reliable context provisioning
- Common querying interface
- Push and pull access mode
- Modularity and extensibility

- Key ideas:
  - Incorporation of multiple context provisioning strategies
  - Exploit ubiquitous connectivity
  - SQL-like querying interface

Context Item

Type	Value	Time hhmmss.sss	Lifetime	Source*	Metadata*

Location item

Type	Value	Time	Lifetime	Source*	Metadata*
location	latitude = 2425.6170 longitude = 6007.5084	092204.999	100000ms	GPS nokiaE234	Validity: A status Precision: HDOP 24.4

* = optional field

Context Query Model

```
SELECT <context type>
FROM <source>
WHERE <predicate clause>
FRESHNESS <time>
DURATION <duration>
EVERY <time>
EVENT <predicate clause>
```
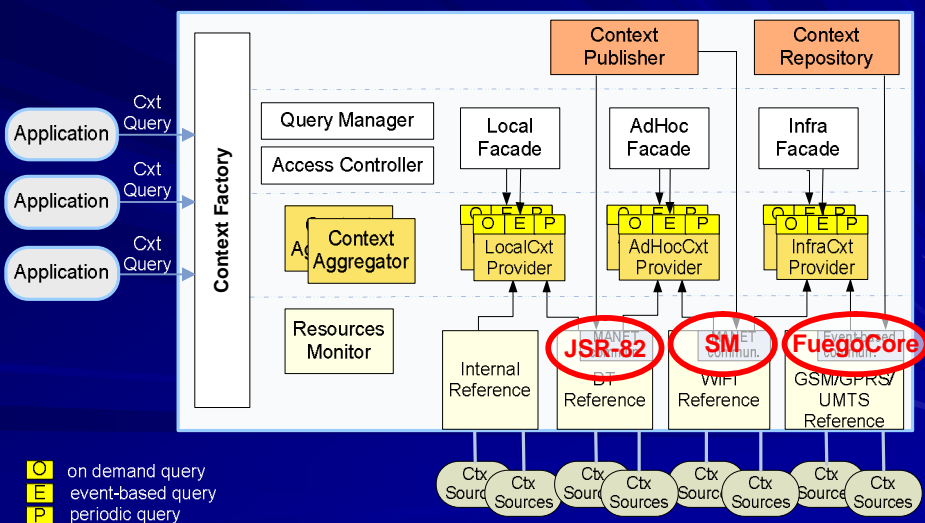
Example:
```
SELECT temperature
FROM adHocNetwork(first3,3)
WHERE accuracy = 0.2
FRESHNESS 30sec
DURATION 1hour
EVENT AVG(temperature)>25
```

Contory Software Architecture

O	on demand query
E	event-based query
P	periodic query

Implementation

- Implemented in Java
  - Java 2 Micro-Edition (J2ME) with CLDC 1.1 and MIDP 2.0
  - J2ME with CDC
- Development using Nokia Series 60 phones and communicators Nokia 9500
- Event-based platform: **Fuego middleware** (developed at HIIT)
- Ad Hoc Network Programming Support: **Smart Messages** computing platform (developed at Rutgers University)

Smart Messages at a Glance

- User-defined distributed applications similar to mobile agents
- Composed of code, data, and execution state
- Execute on nodes of interest named by properties (tags)
- Migrate between nodes of interest
  - Self-route at every node in the path during migrations
  - Use geographical routing and content-based routing
- Two Smart Messages implementations:
  - modified KVM, runs on iPAQs
  - Java VM, J2ME CDC, runs on smart phones

Outline

- Motivations

- Context provisioning strategies

- Contory Middleware

- Evaluation

- Current Status and Future Works

Ongoing Evaluation

- Middleware Performance
  - Latency and memory experiments
  - Energy consumption
- Application prototypes
  - DYNAMOS Sailing application
  - Traffic information application
- Some preliminary results...

Outline

- Motivations
- Context provisioning strategies
- Contory Middleware
- Evaluation
- Current Status and Future Works

Current Status and Future Works

- Prototype running on Nokia Series 60 and Communicators Nokia 9500
- Ongoing implementation
  - Multi-hop ad hoc context providers
  - Long-running queries processing
- Ongoing evaluation
  - More accurate energy consumption measurements
  - Applications development using Contory

References

[1] J. E. Bardram. The Java Context Awareness Framework (JCAF) - A Service Infrastructure and Programming Framework for Context-Aware Applications. In *Proceedings of the 3rd International Conference on Pervasive Computing (Pervasive 2005)*, 2005.

[2] A. K. Dey, D. Salber, and G. Abowd. A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications. *Human-Computer Interaction,* 16(2-4):97–166, 2001.

[3] H.W. Gellersen, A. Schmidt, and M. Beigl, *Multi-sensor context-awareness in mobile devices and smart artefacts*, Mobile Networks and Applications (2002), 341–351.

[4] F. Hohl, L. Mehrmann, and A. Hamdan. A context system for a mobile service platform. In *ARCS '02: Proceedings of the International Conference on Architecture of Computing Systems*, pages 21–33, London, UK, 2002. Springer-Verlag.

[5] J. Hong and J. Landay. An Architecture for Privacy-Sensitive Ubiquitous Computing. In *Proceedings of The Second International Conference on Mobile Systems, Applications, and Services (Mobisys'04)*, pages 177–189, Boston, MA, 2004.

[6] A. Schmidt, K. A. Adoo, A. Takaluoma, U. Tuomela, K. V. Laerhoven, and W. V. de Velde. Advanced Interaction in Context. In *Proceedings of the First Symposium on Handheld and Ubiquitous Computing (HUC'99)*, pages 89–101, Karlsruhe, Germany, September 1999.

[7] S. Yau and F. Karim. A context-sensitive middleware for dynamic integration of mobile devices with network infrastructures. *Journal Parallel Distributed Computing*, 64(2):301–317, February 2004.

Thank you!

oriana.riva@hiit.fi