Applying MUPE Context Producers in developing Location and Context Aware Applications

Kimmo Koskinen kimmo.m.koskinen@iki.fi
Kari Heikkinen kari.heikkinen@lut.fi
Jouni Ikonen jouni.ikonen@lut.fi

Lappeenranta University of Technology, Finland

May 23rd, 2006
ICEIS 2006
5th International Workshop on Wireless Information Systems

1. Introduction

2. Producing and mediating context with MUPE (Multi User Publishing Environment)

3. Context mediation usage example: Trix (Tribal Exhange)

4. MUPE analysis

5. Conclucions

Introduction

- Mobile terminals ideal for location and context-aware applications
  - Communication and sensor interfaces (e.g Bluetooth, GPRS...)
- However, application development is demanding (from the point of view of context-awareness)
  - Infrastructure
  - Testing
  - Simulating
- MUPE provides Context mediation: Server side context information provisioning and usage scheme
  - Instead of direct ad-hoc solutions, provide a reusable scheme for context information and application logic

Research in the field of context-awareness

- Ubicomp and Pervasive conferences
- Context computing focuses both on identifying/collecting the context and using the collected context information[1].
- Modelling of context information, semantics and ontologies
  - Bullet proof semantic modelling maybe a very ambiguous task
- Time and location the most used contexts
  - There are very many other, possibly interesting context sources to be used in application logic!
- Our choise: the MUPE (Multi User Publishing Environment) framework

[1] A.Ferscha, C.Holzmann and S.Oppl, Context Awareness for Group Interaction Support, Proceedings of MobiWac 2004

MUPE (Multi User Publishing Environment) overview

- Client:
  - J2ME MIDP(2) client
  - XML protocol to communicate with the server
  - XML UI scripts, rapid to develop (no compile time)
- Server:
  - Java application server
  - Middleware components for dividing tasks:

Client manager: Client connections
Context manager: Context information receiving and processing
World manager: Java application logic (the main thing in MUPE development)

  - Stores and sends XML UI scripts to the Client
  - Context scripts for creating events from context data

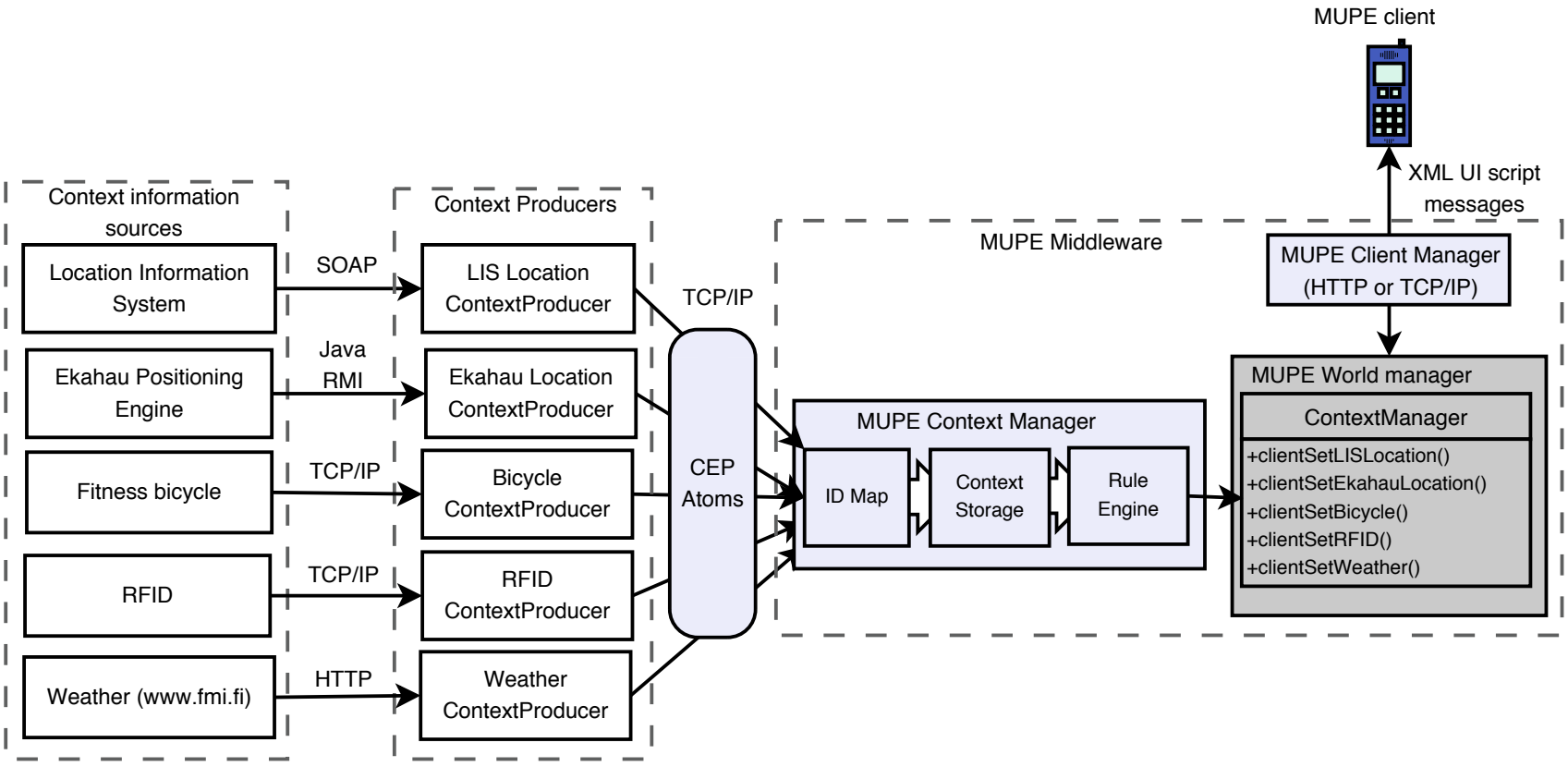
Producing and mediating context in MUPE

- Producing context information to applications can be complicated: many sources, many interfaces, many protocols...
- MUPE provides a pattern: context production
- A Context producer component is implemented for each context source
- This component wraps the context data into CEP (Context Exchange Protocol) format
- One Context producer can service any number of MUPE servers
  - Basically a simple TCP/IP server
- MUPE can then use the context data in a uniform way

Context sources implemented (overview)

- Location contexts
  - LIS (Location Information System, a cell -based WLAN positioning system)
  - Ekahau Positioning Engine (a commercial WLAN positioning system)
  - RFID tag reader
- Environmental context
  - Finnish Meteorological Institute weather web service
- Physiological context
  - Tunturi recumbent fitness bicycle (a bike where you sit while pedalling)

The big picture....



And explanation....

Location information with LIS (Location Information System)

- An implementation of a cell based location system
- Provides cell/description location information based on WLAN cells
- SOAP (Simple Object Access Protocol) interface for requesting location information and notifications for updates in location information (tracking)
- The context producer component followed specific WLAN devices and provided this information as CEP atoms for MUPE

Location Information with Ekahau Positioning Engine (EPE)

- Map/area/x,y location information
- Uses signal "fingerprinting"
  - Signal strengths are recorded in the positioning area to make a positioning model
  - After the positioning model is made, it is stored into EPE
  - Client devices (e.g. laptops) run a measuring software which sends signal samples to EPE
  - EPE then uses the signal samples to position the client with the model
- Provides a continuous "stream" of location estimates
- The Context producer registered to receive location estimates for given devices and provided the estimates as CEP atoms

Location information with RFID

- Access 7 RFID tag readers and passive tags
- The tag sightings are readable through are serial interface from the reader
- A dedicated C program was made to read tag sightings and deliver them to to the context producer through a TCP connection
- The context producer then provided the tag sighting information as CEP atoms to MUPE

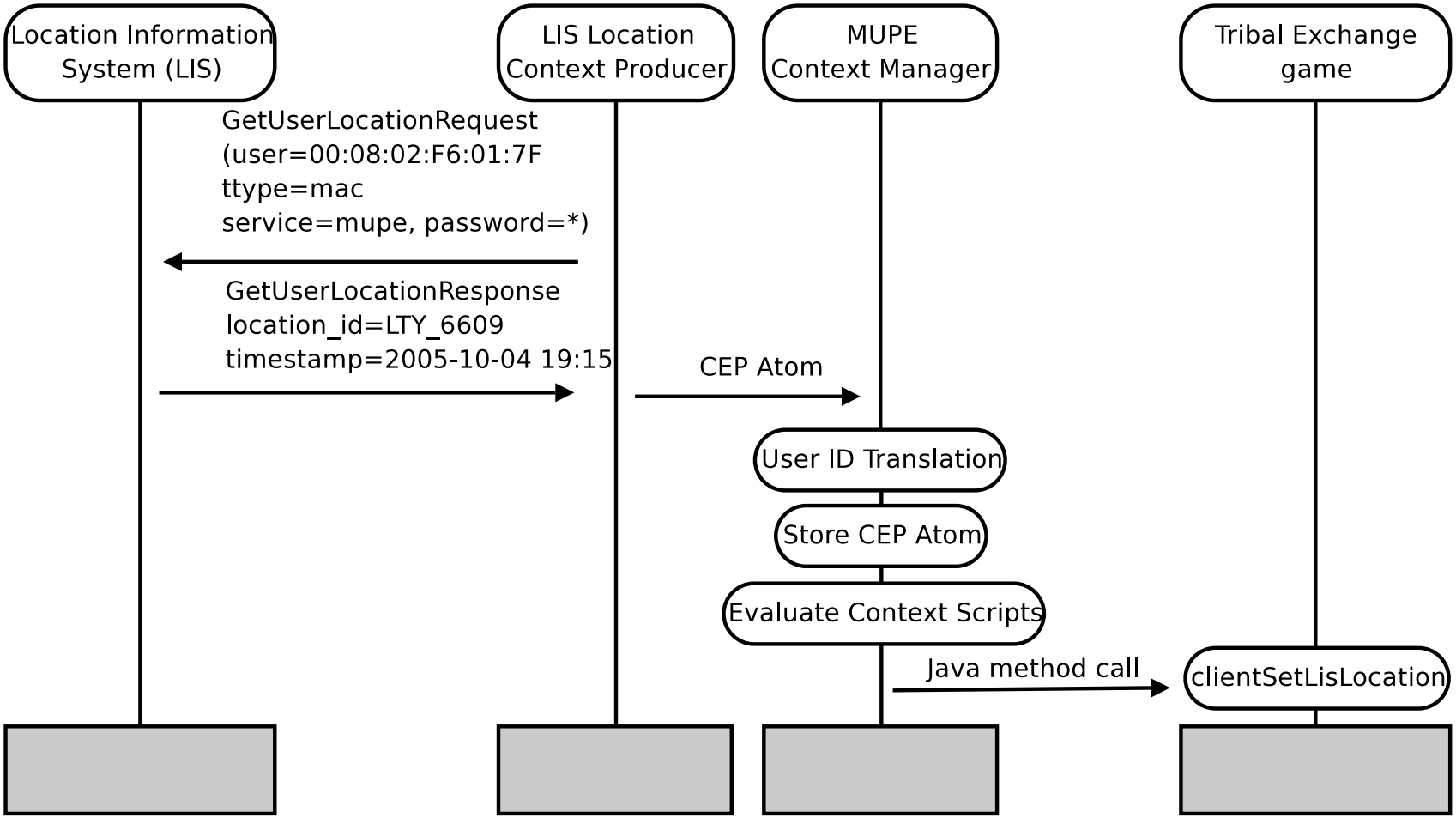
Weather Information

- There are many weather services in the Internet (e.g. AccuWeather)
- The Finnish Meteorological Institute web service was used for the weather context producer
- Provides measurements (temperature, wind speed and direction, cloudiness...) once per hour
- Simple implementation for the context producer
  - Periodically fetch the HTML page of the measurements, parse it and provide the data as CEP atoms

Physiological context information

- A tunturi recumbent fitness bicycle was used
- Provides heart-rate and pedalling speed (km/h) through a serial connection
- A dedicated C program was made to transfer the exercise information with TCP
- The context producer queried the exercise information with the TCP protocol and provided the exercide data as CEP atoms

Example context production with LIS



An example of location information mediation from LIS to MUPE application

Example CEP Atom message

```
<atom name='LISLocation'
      timestamp='2005-9-5 19:23:26,00 +1'
      source='http://gamesrv.wlpr.net:1234'
      userId='00:08:02:F6:01:7F'>
  <string name='timestamp'>2005-9-5 19:15:35...
  </string>
  <string name'location'>LTY_6609
  </string>
  <string name='description'>...building, 6th floor
  </string>
</atom>
```

Mapping context data id's to MUPE application id's

```
<addUserMap>
  <userMap>
    <contextProducerName>LISlocation
    </contextProducerName>
    <contextProducerId>00:08:02:F6:01:7F
    </contextProducerId>
    <serverId>Brian the great
    </serverId>
  </userMap>
</addUserMap>
```

Context Script example

```
<?xml version=''1.0'' encoding=''ISO-8859-1'' ?>
<script id='lis'
  xmlns='http://www.nokia.com/ns/cep/script/1.0/'
  xmlns:cep='http://www.nokia.com/ns/cep/1.0/'>
  <if>
    <atomChanged>
      <atomRef userId='*' name='LISLocation'>
      </atomRef>
    </atomChanged>
    <actions>
      <mupeCall>
        <![CDATA[2::clientSetLocation
                 {${userId}}
                 {${LISLocation::location}}]]>
      </mupeCall>
    </actions>
  </if>
</script>
```

Context Mediation by Example: Trix (Tribal Exhange)

- An example of a context-aware mobile game
- Combines a virtual game world with the physical world (location and weather)
- Game logic:

Goal:	Protect the houses of the tribe from natural disasters
Game world:	Map with forest, water, rock and animals
Objective:	Tribes build protecting walls around the houses and receive a new house if all existing houses are safe (behind an intact wall)
Interaction:	Players can trade wall building blocks and items with other (team) players

Contexts in Trix

- Location
  - Codes hidden in locations determined by LIS
  - Players can search for codes in real world
  - Codes give game items: wall building blocks or tools (e.g. dynamite to blow up rocks)
  - Inputting code into the game uses the code
  - Alternatively codes can be distributed in the real world with e.g. paper notes
- Weather
  - Temperature, wind speed and humidity decrease the strength of the walls
  - Ice wall melts, straw wall is blown away by wind, humidity makes wood wall soft

MUPE analysis[2]

- Context usage is application-oriented, dynamical changes in context data have to be implemented by the application developer
- Up to the developer to handle contextual state inferring. MUPE is targeted toward mobile games, may not be enough for "real applications"
- Making Java method calls from context scripts is direct and fast. But the division of context specific logic may not be clear.

[2]Partially based on Korpipää Ph.D Thesis: Blackboard-based software framework and tool for mobile device context awareness

MUPE analysis continued

- Slow response times as the appplications are client-server based (limited but a growing set of direct client-side functionality, such as persistent client side UI scripts, Bluetooth and GPS plugins)
- CEP protocol lacks security
- Context handling in compound structures may waste resources.
- Context recognition, customization and context-based application control not addressed
  - Up to the application, provides mainly a platform to develop applications in which context plays a major role

Conclusions

- Context mediation in MUPE allows wide usage of contxt information but does not exhaustively consume resources in the client devices
- Context producers provide a uniform (and reusable) access to context data
  - A blackboard -based context script engine, with reusable scripts
- Allows application developers to concentrate on context specific application logic
  - Rapid development of context-aware applications