Component-based, Context-aware Software Systems

Workshop on Spontaneous Networking
Rutgers University

Michael Przybilski
Outline

• Motivation
• Research Problems
• Thesis
• Example
• Ongoing Work
Motivation

• Increase of available context information
  • Wireless information devices
  • Integrating increasing number of input and output devices

• Use for applications
  • Explicit, potential

• Use of communication possibilities
  • Remote devices
  • Number of devices
Context and Context Reasoning

- **Context** is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and application themselves.

  [A. Dey and G. Abowd, 1999]

- **Context-reasoning** can be defined as deducing new and relevant information from the various sources of context-data.
Challenges

• Acquisition, processing and provision of context data
• Rapidly changing and uncertain sources

• Reuse
  • Common steps
  • Context information

• Scalability
• Flexibility
Software Components

• A *component* is a coherent package of software artifacts that
  • can be independently developed and delivered as a unit, and
  • can be composed, unchanged, with other general components to build something larger

• An *interface* is a description of a set of operations related to the external specification of a component. An interface consists of:
  • a set of operations that a component needs to access in its surrounding environment (*required interface*),
  • a set of operations that the surrounding environment can access on the given component (*provided interface*).

• An *operation* is unit of functionality implemented by a component which may map to:
  • a method,
  • a function,
  • a procedure.
Distribution

- Distributed processing / information sharing
  - Nearby, resulting in a context-aware system
  - Remote

- Simple sensors

- Wireless information devices
  - Online processing
  - Groups of devices
  - Limited resources (processing, storage, networking, power, etc.)

- ...

- Servers
  - High level of resources
  - Specialized
Research Problems

• What are the minimum requirements for a distributed context management system, based on software components?

• How can context reasoning mechanisms be efficiently integrated?

• How can context-aware applications be built from software components?

• How can additional requirements be efficiently integrated?
Scope

• Ranging from embedded devices, to mainframe computers
  • Context management on user’s wireless information device
  • Using remote devices for acquisition of context information and processing tasks (context reasoning)
  • Multi-bearer ad-hoc / P2P networks

• Sensor-based context
  • Not specific to location context
  • Not specific to static contexts (device configuration)

• Context-aware computing view
  • Not specific to semantic web

• Flexible, component-based framework that can integrate additional QAs
Component Framework

• Functionality
  • Basic sensing
  • Learning + Inference
    • Provision of new context (prediction)
    • Abstraction of higher level of context (interpretation)
  • Store context data
  • Application logic

• Basic communication

• Compile-time / runtime extension with other QAs
  • Access list; restrict communication to specific other components
  • Privacy, security, (network) location
Components

- 1-way communication (Messaging; Publish / Subscribe – Publish; synchronous / asynchronous / isochronous)

- 2-way communication (RPC; synchronous / asynchronous)

- Optimization: combination of components, re-use of common functionalities at runtime
Simple Application

• The stolen / forgotten mobile phone
  • Sensor
    • Simple BT beacon
    • Attached to user
    • Transmit Freq. <= 1 Hz
  • Reasoning
    • Beacon is with user
    • Mobile phone is owned by user
    • Should never be out of reach
      => stolen / forgotten
  • Application logic
    • Sound / display alarm if stolen / forgotten

subscribe(select NULL where user = Michael, epoch = 1000)
Advanced “Application”

- Inferring User Activity from 3D Accelerometer
  - Steps
  - Classification
  - Feature Selection
  - Feature Extraction
  - Pre Processing
  - Data gathering
Inferring User Activity from 3D Accelerometer

- Components

- Sensor

- Preproc.

- FeatureExtr.

- FeatureSel.

- Classifier

get(select activity from activity where user = Michael)

subscribe(select x, y, z from accelerometer where user = Michael, epoch = 10)

subscribe(select variance, absoluteMagnitude from Feature-Selection where source = accelerometer and user = Michael, epoch = 10)
Inferring User Activity from 3D Accelerometer

- **Sensor**
  - 220,000 samples (100 Hz)
- **Preprocessor**
- **Feature Extraction**
- **Feature Selection**
  - Variance, absolute Magnitude of Acceleration
- **Classifier**
  - Simple linear classifier
  - 300 samples for learning
  - 300 samples for calibration

Nokia 6600

Node 1
- Sensor

Node 2
- Classifier
- Preproc.
- FeatureSel.
- FeatureExtr.
Inferring User Activity from 3D Accelerometer

Variance

Absolute Magnitude

*International Conference on Artificial Intelligence 2005
Other work

• Agent-paradigm
  • Distributed preprocessing
  • Integration of policy-based privacy and security
  • Autonomous Intelligent Systems: Agents and Data Mining, 2005

• Evaluation of reasoning mechanisms
  • Software framework for remote data gathering
  • WS / EJB components / WEKA
  • Pervasive Systems and Computing, 2005
Ongoing Work

• Further integration
  • iMotes (NesC)
  • Series 60 / 80 Mobilephones (Symbian C++)
  • JBoss Application Server

• Communication
  • Ad-hoc (BT) and P2P (IP; 3G/GPRS / LAN)

• Energy saving by remote processing

• Reasoning mechanisms
  • Simple Bayesian networks, SVM
Questions?