Towards a flexible control center for cyber-physical systems

Martin Franke, Diana Brozio, Thomas Schlegel
Agenda

- Ubiquitous and Cyber-Physical Systems vs.
- VICCI Project
  - Motivation
  - Goals
  - Global Scenario
- Towards a flexible control center for cyber-physical systems
  - State-Of-The-Art
  - Concept
  - Scenarios
- Conclusion and further work
Ubiquitous and Cyber Physical Systems

- 1991: “In the 21st century the technology revolution will move into the everyday, the small and the invisible.” [12]


- 2008: “Cyber Physical Systems are integrations of computation and physical processes” [Edward A. Lee]
Cyber-Physical Systems (based on ACATECH2012)

Towards a flexible control center for cyber-physical systems

Internet Of Things, Data and Services (Knowledge)

Cyber Physical Systems (Information)

Networked Embedded Systems (Data)

Embedded Systems

...
Classification based on Sou [9]

- Individual Smart Space (ISS): e.g. Static SmartHome environments
- Open Smart Spaces (OSS): e.g. Open SmartHome with seamless integration of existing devices; use of Internet Services is also possible
- Smart Communitys (SC): e.g. Social Interconnection of OSS
VICCI – Visual and Interactive Cyber-Physical Systems Control and Integration
Area of application for cyber-physical systems shows an increasing interest, like AAL with robots, intelligent work offices, smart traffic infrastructures or smart grids.

Composition, development and optimizing of this networked ubiquitous systems, consisting of PCs, sensors and actuators, is a complex process [8].
Towards a flexible control center for cyber-physical systems
VICCI Project – Main Goals

- Dynamic assisting of the user in cyber-physical systems
- Help to visualize and control of as much as possible user groups
- Knowledge-assisted interconnection for seamless integration of heterogeneous CPS-elements, like sensors, actuators and existing devices
- Methods for quality assurance, like reliability, real time behavior and efficiency
Towards a flexible control center for cyber-physical systems
Flexible control center for cyber-physical systems

- Control center is an adaptive, ubiquitous dashboard to offer personalized and intuitive visualizations/interaction abilities [2]
- Controlled by different devices, like smartphones, tablets and PCs, and their corresponding interaction concepts

- Concept for adaptive controlling with devices (Apps)
- Concept for a middleware combining all together

Application Layer

Semantic Middleware

Sensor-Actuator Layer
State-Of-The-Art – Requirements

- Requirements for middleware-based context-aware applications [5]
  - Coordination of all resources
  - Interoperability with all participating devices
  - Mobility of the user
  - Autonomous behavior of the system
  - Auto-Discovery of services and devices
PERSONA [1]

- Middleware-based context-aware system
- Self-organizing infrastructure with point-to-point connection (after registration)
- Components register on communication bus, middleware provides channels
NAM – Networked Autonomic Machines [7]
- Middleware-based context-aware system
- Designed for autonomic wearables
- Fully decentralized P2P Network
Nodes are described by
- Set of physical resources
- Set of functional modules
  - Provided Services
  - Consumed Services
  - Consumed Context Events
  - Provided Context Events

Services are described according to the IOPE pattern from OWL-S [6]
State-Of-The-Art – Drawbacks

- Main drawback is the lack of flexibility with later added devices or distributed instances of applications
- PERSONA defines communication channels in time of registration
- NAM can handle late-added devices, but cannot update provided channels

<table>
<thead>
<tr>
<th></th>
<th>PERSONA</th>
<th>NAM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coordination</td>
<td>(X)</td>
<td>(X)</td>
</tr>
<tr>
<td>Interoperability</td>
<td>(?)</td>
<td>X (NAM4J – Java)</td>
</tr>
<tr>
<td>Mobility of the user</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Autonomous behaviour of</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>the system</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Auto-Discovery of services and devices</td>
<td>X</td>
<td>(X)</td>
</tr>
</tbody>
</table>

Towards a flexible control center for cyber-physical systems
- Seamless integration backend
  - UI is done over Apps
  - All participated elements (applications, sensors and actuators) register them at the Semantic Middleware (SeMiWa)
Execution Layer, respectively Runtime Layer
Distributed over multiple devices like smartphones, tablets or PCs
Abstract Layer, cause the heterogeneous devices

Apps have to provide
- Interface for data transmission (bidirectional)
- Dynamic description of Input / Output (IO) data in semantic way, e.g. “get temperature X from Y” or “open windows Z in W”
  - Parameterized during runtime, e.g. X={all, < 18°C, …}; Y={all, living room,… }
- Precondition and Effect (PE), e.g. “windows Z in W are opened”, deemed to used for error checking
Concept – SeMiWa

- Semantic Middleware
  - Knowledge Base with all lifecycle information about applications, sensors and actuators (Registrar)
  - Model repository for unifying messages
Concept – SeMiWa

- Registration of a sensor/actuator
  - Acquisition: Provide a network interface between the layers to transmit data to SeMiWa
  - Interpretation: Unify plain data against sensor model, annotate the provided channel
  - Routing: Provide a network interface interface between the layers to transmit data from SeMiWa
Routing of data to or from applications

- **Acquisition**: Provide a network interface between the layers to transmit data to SeMiWa
- **Aggregation**: Decompose semantic annotated instructions, aggregation of sensors/actuators (situation)
- **Routing**: Provide a network interface between the layers to transmit data from SeMiWa, routes data according to decomposed IOPE
- Distributed over multiple devices like microcontroller, robots or PCs
- Abstract Layer, cause the heterogeneous devices

- Elements has to provide the “Semantic Driver”
  - Network interface for data transmission
  - Semantic description of the sent data (sensor)
  - Semantic description of possible control instructions (actuator)
  - *Reconfiguration during runtime*
Scenarios

- Data flow from sensors to applications

```
Scenarios

Data flow from sensors to applications

Tablet1
  App1
  App2

Smartphone1
  App3
  App4

Acquisition → Interpretation / Aggregation → Routing

SD1
  S1
  S2

SD2
  S3

SD3
  S4

Sensor Layer

```
### Scenarios

Control flow from applications to actuators

- 1) Open all windows in living room
- 2) Heatings in living room are turned on
- 3) Turn of heatings in living room
- 4) *Real instruction* (Turn off...)
- 5) Open all windows in living room
- 6) *Real instruction* (Open windows...)
Conclusion and further work
Conclusion

- Need of adaptive user interface spread over devices, based on model description

- SeMiWa
  - Flexible data routing according to semantic model descriptions
  - Seamless integration of new components
  - Error checking with pre- and postcondition
  - Potential knowledge-tracking of made decisions
Further Work

- Creating Models
  - Sensor models and aggregation rules
  - Application models (IOPE)

- Dashboard concept for real-time and user-centered data visualization

- Reuse of technologies
  - Soprano for real-time processing and deduction on semantic models
  - UPnP connector for in-house communication and XMPP for WAN [10, 3]


