Integrating Studierstube and DWARF

Technische Universität München

Martin Bauer, Otmar Hilliges, Asa MacWilliams, Christian Sandor,
Martin Wagner, Gudrun Klinker

Vienna University of Technology

Joe Newman, Gerhard Reitmayr, Tamer Fahmy,
Thomas Pintaric, Dieter Schmalstieg

Component-based approaches

- Studierstube
  - C++ classes on top of Open Inventor (OIV)
  - Object-oriented scene graph
    - Geometric information
    - Active interaction objects
  - Distribution of applications
    - Shared scene graph through DIV
- OpenTracker
  - Library operates on tracking data
  - Breaks up transformations defined by XML

Component-based approaches

DWARF

- Basic unit is distributed ***service***
  - needs, abilities
- Services bundled with hardware in units

Strong modular design

- Easily extended by adding new components

Adapters

- OpenTracker <-> DWARF
- Open Inventor <->DWARF

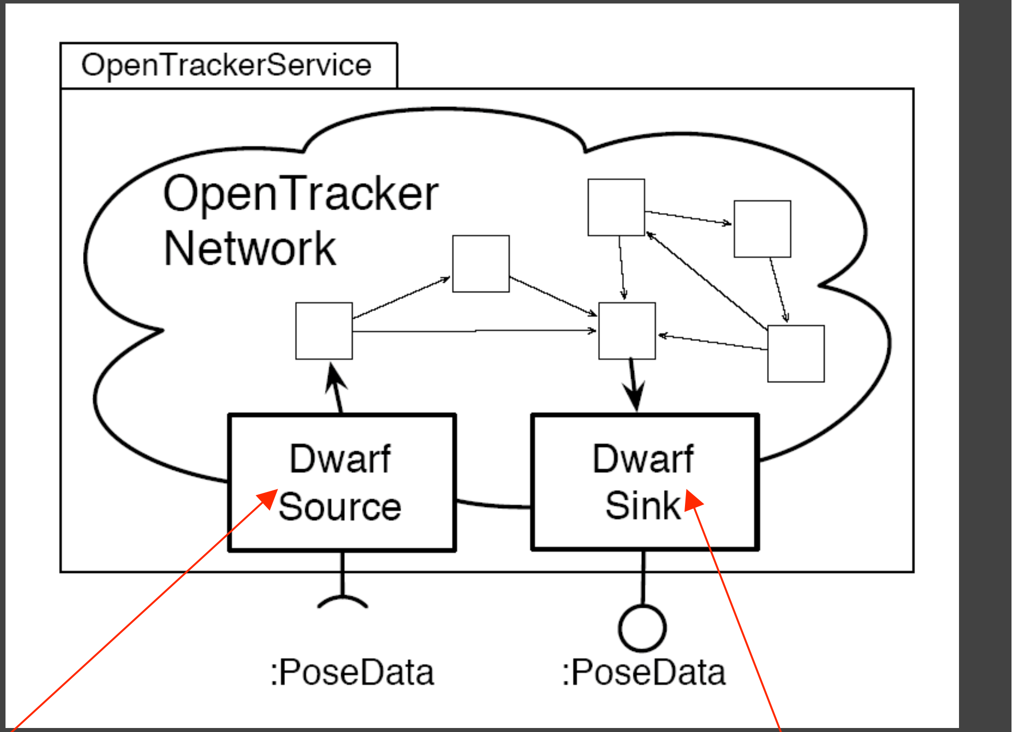
OpenTracker network as DWARF service

OpenTracker extensible by modules

- interface devices
- algorithms
- other frameworks

DWARF module

- Implements nodes
  - *DwarfSink*
  - *DwarfSource*
- Complete DWARF service

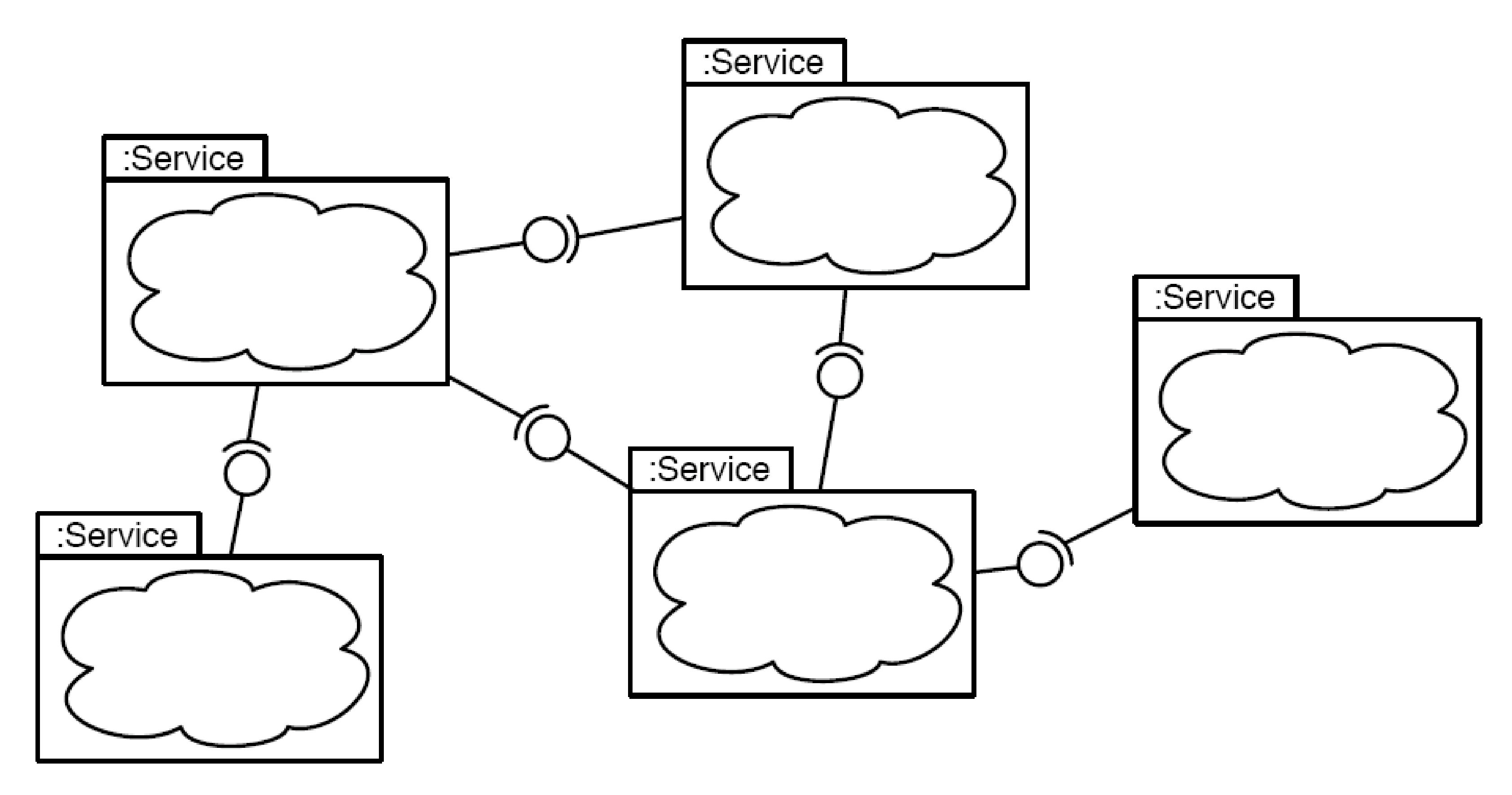


needs

abilities

DWARF connecting different OpenTracker networks

Smaller networks for dynamic scenarios



DWARF service embedded in an OIV scene graph

OIV supports nodes in a scene graph

- Contain *Fields* of predefined types

DwarfService node is single DWARF service

- *Fields* configure service parameters
- Contains lists of subnodes
  - needs, *DwarfNeed*
  - abilities, *DwarfAbility*

Studierstube can express DWARF service within scene graph

Conclusion

Wider choice of tools leads to more elegant solutions

Reduce overheads

- device drivers, filter objects realised only once

Existing local static setups

- Defined by OpenTracker
- Dynamically combined using DWARF
  - Large-scale Ubiquitous Computing Environments

Encourage interoperability with other AR frameworks

Thankyou

?