Ubiquitous Tracking
Advanced Topics in Augmented Reality

Martin Wagner

martin.wagner@in.tum.de

Lehrstuhl für Angewandte Softwaretechnik

Augmented Reality Group

Institut für Informatik, Technische Universität München

Boltzmannstr. 3, 85748 Garching b. München, Germany

Summary

- No tracking technology is perfect.
- Combining different trackers seems reasonable...
- ... but is very hard to do.
- Ubiquitous Tracking project digs into various aspects:
  - Formal model to describe tracker networks
  - Implementation concept
  - Software architecture for actual implementation
  - Application areas

Overview

- Motivation
- Formal Model: Spatial Relationship Graphs
- Implementation Concepts
- Distributed Software Architecture
- Applications
- Discussion

Motivation

- Every tracking technology has its drawbacks. Can you provide any examples?

Motivation

- Every tracking technology has its drawbacks. Can you provide any examples?
- Combination of many trackers could help. How?

Motivation

- Every tracking technology has its drawbacks. Can you provide any examples?
- Combination of many trackers could help. How?
- Imagine you are a developer creating applications that need tracking information. What API do you envision?

Motivation

- Every tracking technology has its drawbacks. Can you provide any examples?
- Combination of many trackers could help. How?
- Imagine you are a developer creating applications that need tracking information. What API do you envision?
- What problems will arise?

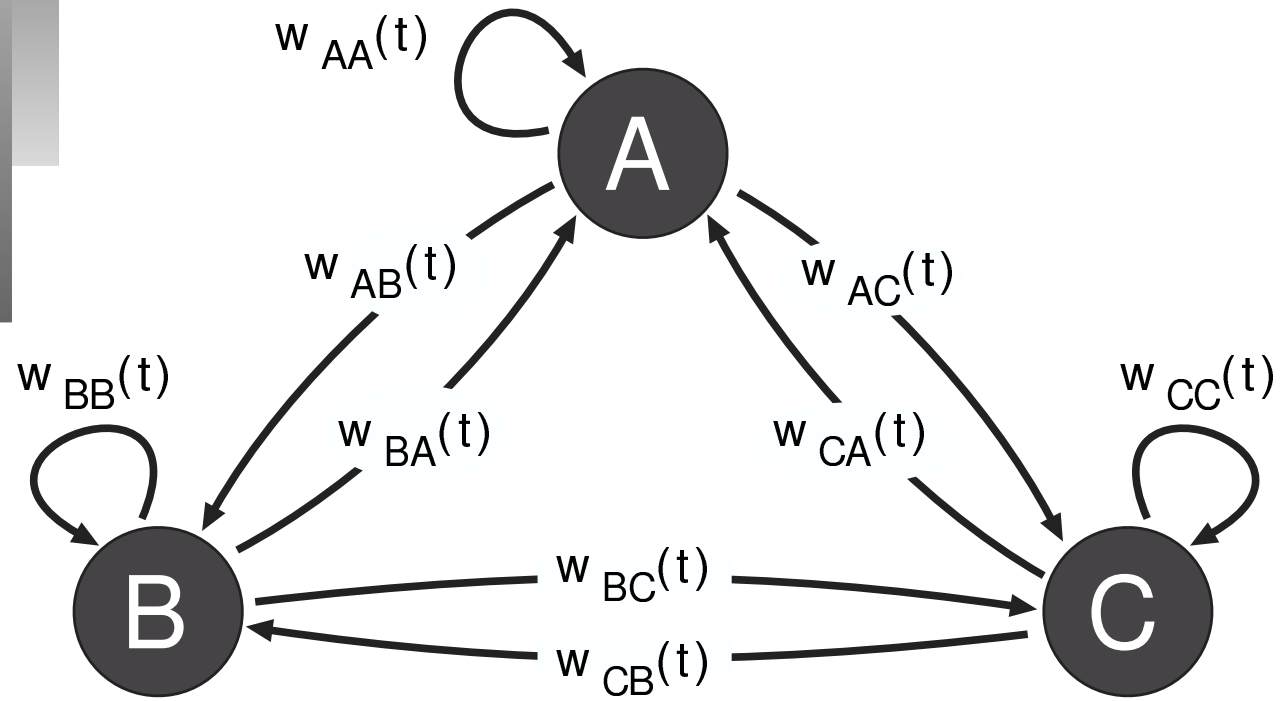
Overview

- Motivation
- Formal Model: Spatial Relationship Graphs
- Implementation Concepts
- Distributed Software Architecture
- Applications
- Discussion

Formal Model: Overview

- Interest: spatial relationships between objects
- Use a graph to model these: nodes are objects, edges are relationships.
- We distinguish between *real*, *measured* and *inferred* relationships.
- Our goal is to infer relationships as similar to the real world relationships as needed by a particular application.
- Often, we have multiple possibilities to do this. To select the best, we use *attributes* to describe measurements and inferences.

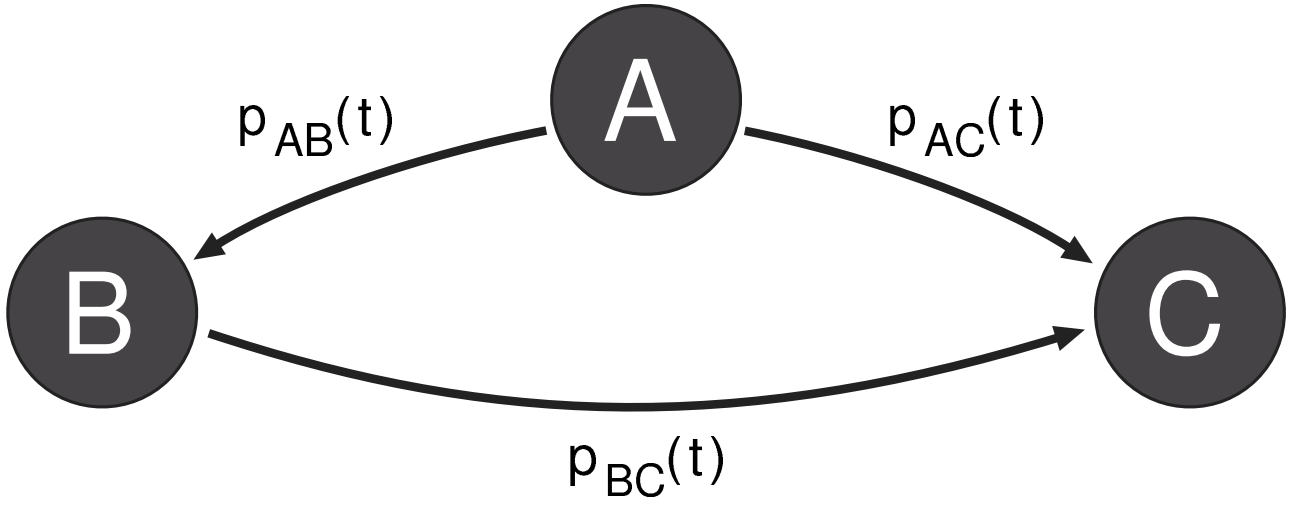
Real Relationships



$$\mathbf{W} : (\Omega = N \times N) \rightarrow w, \text{ where } w : D_t \rightarrow \mathbb{R}^{4\times4}$$

"The world as god sees it."

Measured Relationships



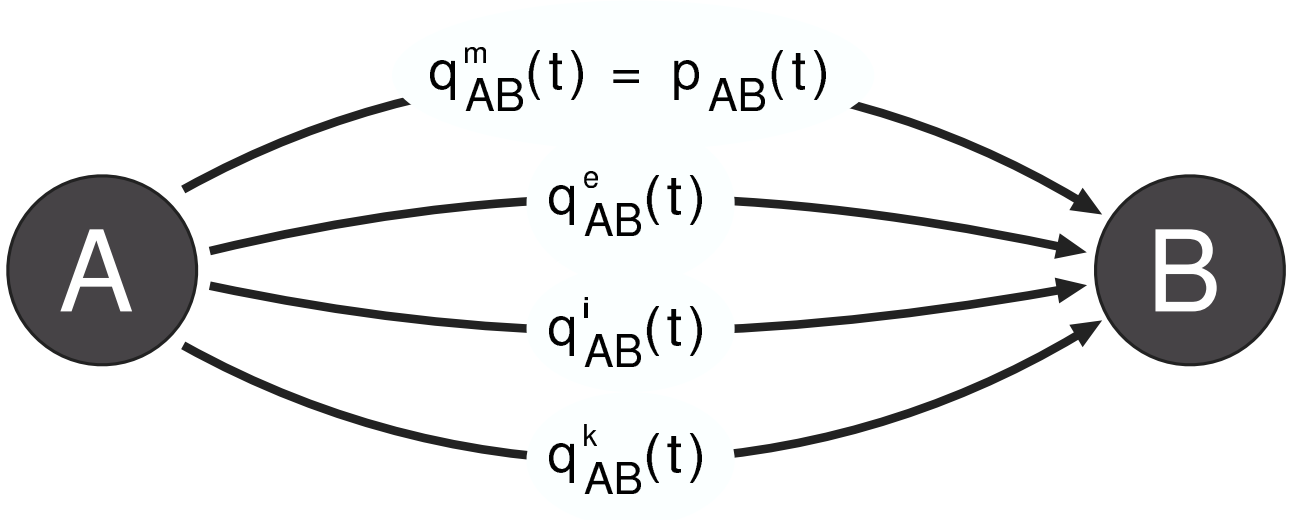
$$\mathbf{P} : (\Phi \subseteq N \times N) \rightarrow p, \text{ where } p : D_t \rightarrow \mathbb{R}^{4\times4} \times \mathcal{A}$$

Attribute set $\mathcal{A}$ describes how imperfect our measurements are.

Measured Relationships Example

$$
\begin{array}{rcrcl}
p_{AB} & : & \{t_1, t_2\} & \rightarrow & \mathbb{R}^{4\times4} \times \mathcal{A} \\
 & & t_1 & \mapsto & (H_1, \mathcal{A}_1) \\
 & & t_2 & \mapsto & (H_2, \mathcal{A}_2) \\
p_{AC} & : & \{t_3\} & \rightarrow & \mathbb{R}^{4\times4} \times \mathcal{A} \\
 & & t_3 & \mapsto & (H_3, \mathcal{A}_3) \\
p_{BC} & : & \{t_4\} & \rightarrow & \mathbb{R}^{4\times4} \times \mathcal{A} \\
 & & t_4 & \mapsto & (H_4, \mathcal{A}_4)
\end{array}
$$

Inferred Relationships



$\mathbf{Q} : (\Psi \subseteq N \times N) \rightarrow q,$ where $q : D_t \rightarrow \mathbb{R}^{4\times4} \times \mathcal{A}$

Multigraph with an edge for every measurement and inference.

Adding New Inferrences

2 Nodes example functions

$$q^m_{AB}(t) = p_{AB}(t) = \begin{cases} (H_1, \mathcal{A}_1) & \text{if } t = t_1 \\ (H_2, \mathcal{A}_2) & \text{if } t = t_2 \\ \text{undefined} & \text{otherwise} \end{cases}$$

$$q^e_{AB}(t) = \begin{cases} (H_1, \mathcal{A}_1) & \text{if } |t - t_1| < |t - t_2| \\ (H_2, \mathcal{A}_2) & \text{otherwise} \end{cases}$$

$$q^f_{AB}(t) = f((H_1, \mathcal{A}_1), (H_2, \mathcal{A}_2), \ldots, t)$$

Attributes & Evaluation Function

Examples of Attributes:

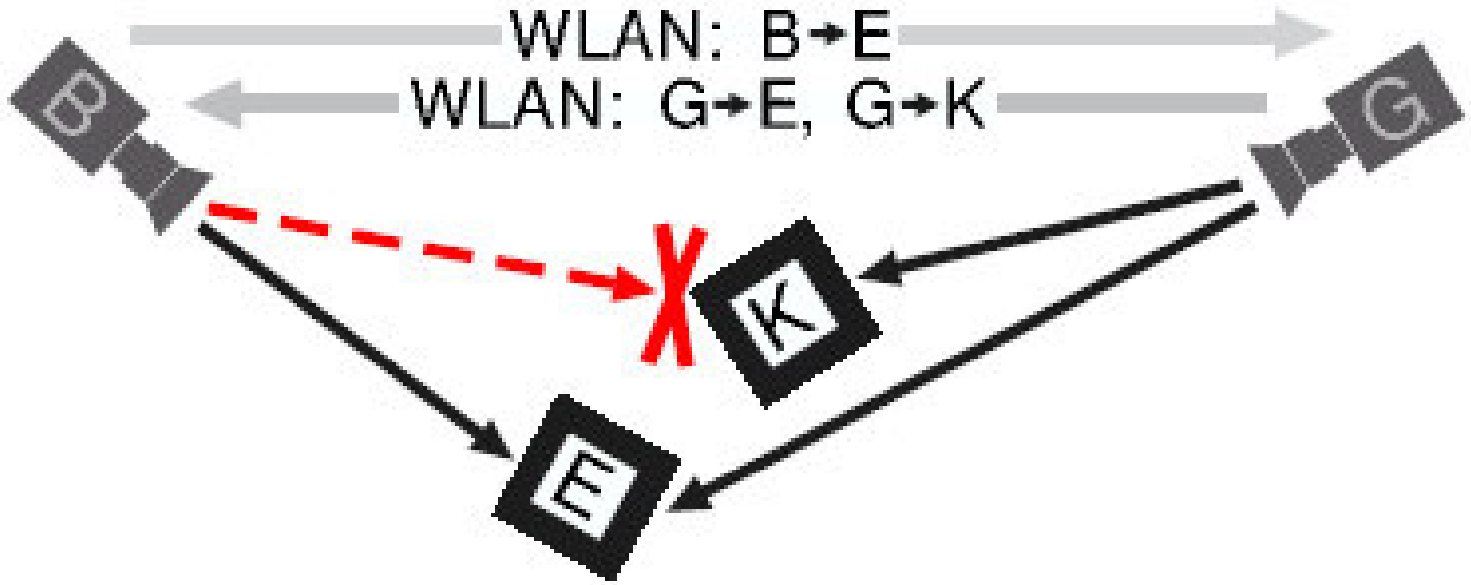
- Latency
- Accuracy
- Monetary Cost
- Update Frequency

Evaluation function defined over attributes:

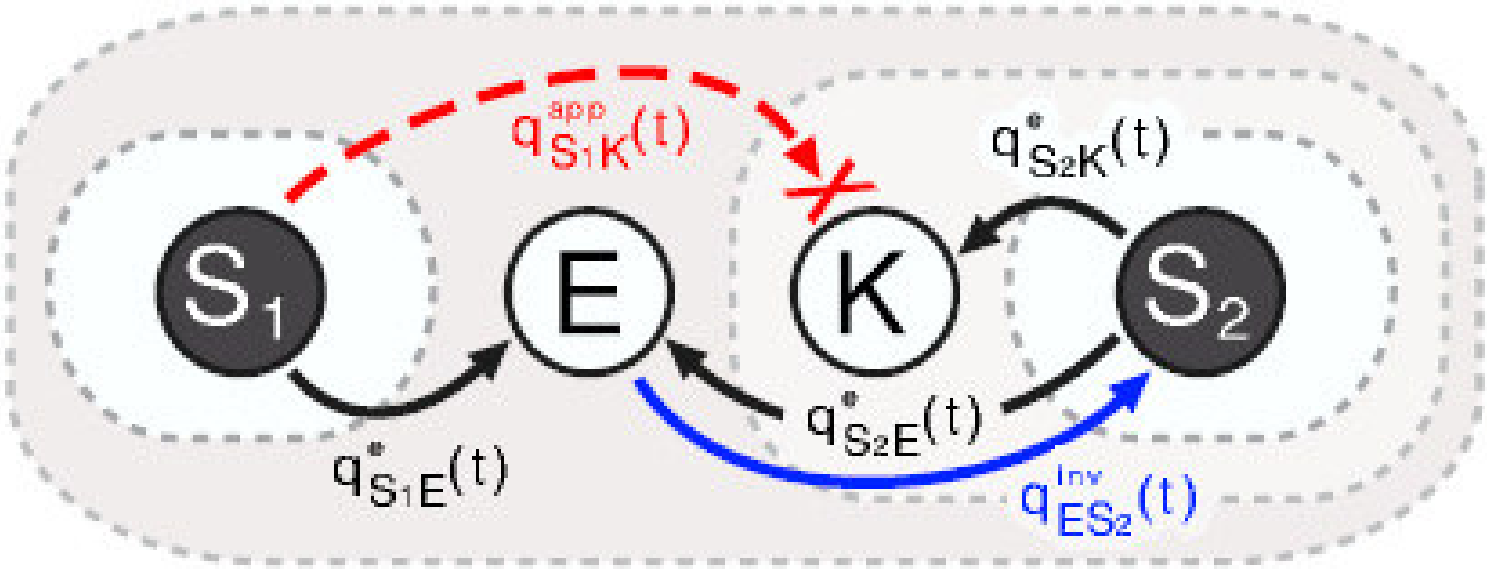
$$\begin{aligned} e \quad : \quad & \mathcal{A}^* \rightarrow \mathbb{R} \\ & \mathcal{A}_i^* \mapsto e(\mathcal{A}_i^*) \end{aligned}$$

Goal: Assign weights to edges to enable common shortest path algorithms.

Example: Shared Tracking Setup



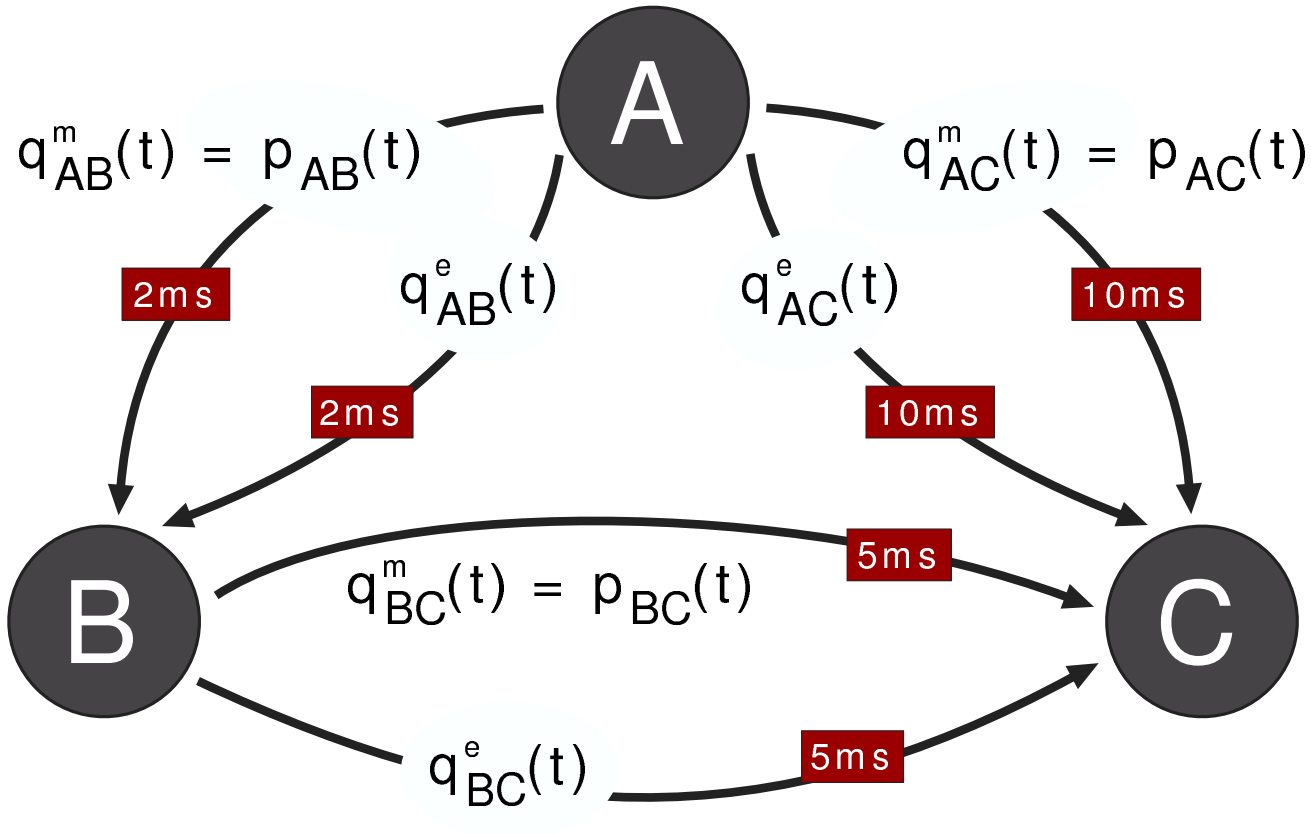
Example: Shared Tracking Graph



Overview

- Motivation
- Formal Model: Spatial Relationship Graphs
- Implementation Concepts
- Distributed Software Architecture
- Applications
- Discussion

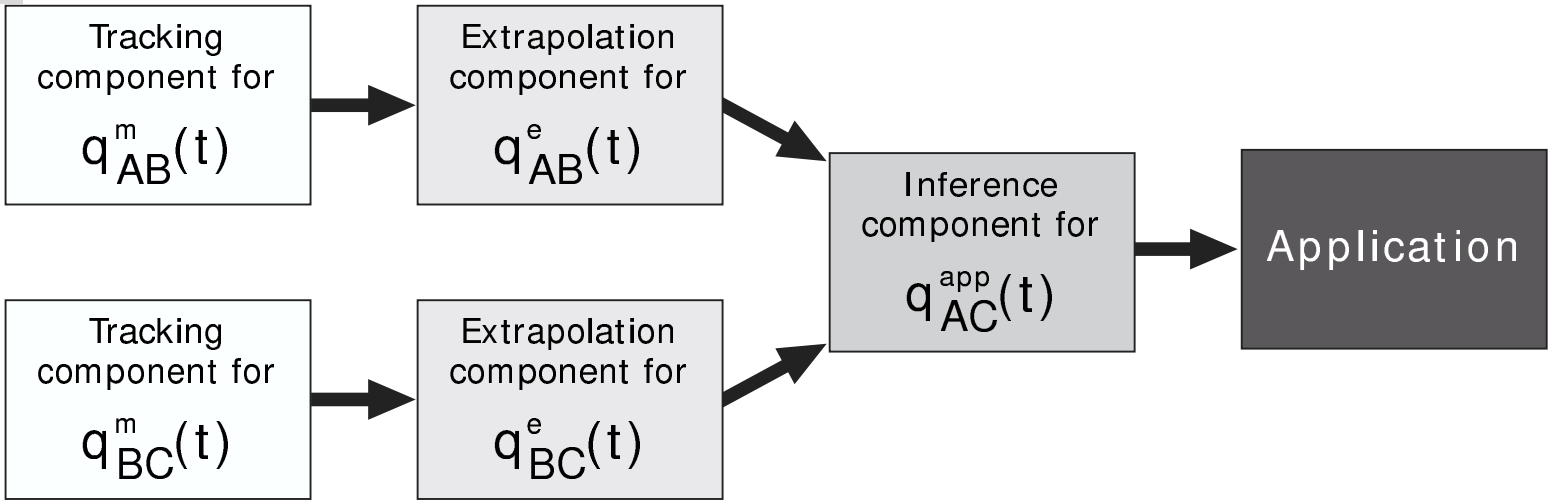
Implementation: Finding Optimal Paths



Implementation: Concepts

- Spatial relationship (SR) graph is just an abstract model
- Key assumption: SR graph's topology and attributes of edges change less frequently than spatial relations between objects.
- At runtime, a *Data Flow Graph* is constructed out of components that deliver tracking data and special *inference components*.

Precomputing Data Flow Graphs



Overview

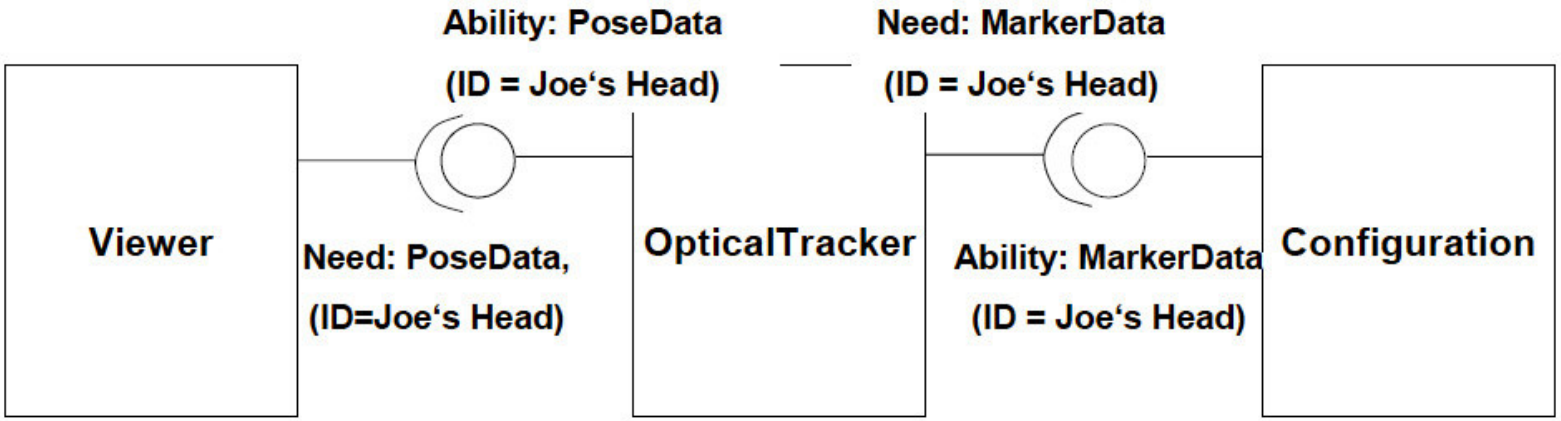
- Motivation
- Formal Model: Spatial Relationship Graphs
- Implementation Concepts
- Distributed Software Architecture
- Applications
- Discussion

Middleware: Distributed Wearable Augmented Reality Framework

- CORBA-based middleware dynamically connects Services (DWARF components) based on description of their Needs and Abilities
- No central component, Service Managers running on each network node handle connection of services
- Ability descriptions may be enhanced using Attributes describing contextual information
- Need description may give Predicates for narrowing the search space of matching services
- Abilities may change at runtime depending on how needs are satisfi ed

Middleware: DWARF ctd.

- Once a need and an ability match, DWARF service manager sets up a connector that both services use to communicate
- Example: Optical tracker



Mapping of Implementation Concept on DWARF

- Trackers modeled as edges in SR graph
- Descriptions of real objects (i.e. tracking hardware, locatable properties) modeled as nodes in SR graph
- Distributed *Ubitrack Middleware Agent (UMA)* cooperates with DWARF service manager to search the SR graph for optimal paths.
- Search done by a distributed asynchronous Bellman-Ford shortest path algorithm.
- Once an optimal path is found, UMA sets up data flow graph using DWARF service managers.

Overview

- Motivation
- Formal Model: Spatial Relationship Graphs
- Implementation Concepts
- Distributed Software Architecture
- Applications
- Discussion

Applications: Smart Building

- Building detects position of each person inside it.
- Continuous computation of most efficient escape routes.
- Routes take into account disabilities etc.
- Information can be provided to fire-officers and other emergency workers.

Applications: Museum Tour Guide

- Every museum visitor gets some display device that uses Ubitrack information.
- Some exhibits only need rough positional information to play some audio about the artwork the user is standing in front of.
- Other exhibits need very precise tracking info to provide visual HMD-style overlays.
- New museum experience by interactive gameplay, taking into account what the visitor has seen already etc.

Applications: Social Issues

- Do we want to provide our own position all the time?
- Is it socially acceptable to track someone down using Ubitrack information?
- How do we make sure the position information can not be abused?

Thank you

. . . any Questions? . . .