Diplomarbeit

Interaction Management for Ubiquitous Augmented Reality User Interfaces

CAR - Car Augmented Reality

Aufgabensteller:	Prof. Gudrun Klinker Ph.D.
Betreuer:	Dipl.-Inform. Christian Sandor
Vortragender:	Otmar Hilliges

May 13, 2004

Summary

- Diploma thesis within the CAR project November ‘03 - May ‘04.
- Designed and implemented a method for interaction management for UAR systems.
  - Providing easy I/O device adaption
  - Introduced an abstraction layer for I/O devices.
  - A powerful formal model to design UI behavior.
- Designed and implemented a runtime development environment.
  - Significantly decreases implementation of UIs (runtime prototyping).
  - Allows the adaption and exchange of devices at runtime.
  - Tweaking and tuning UI behaviour to experiment with interaction techniques is possible.
- Implemented the UI behavior descriptions for CAR.

Outline

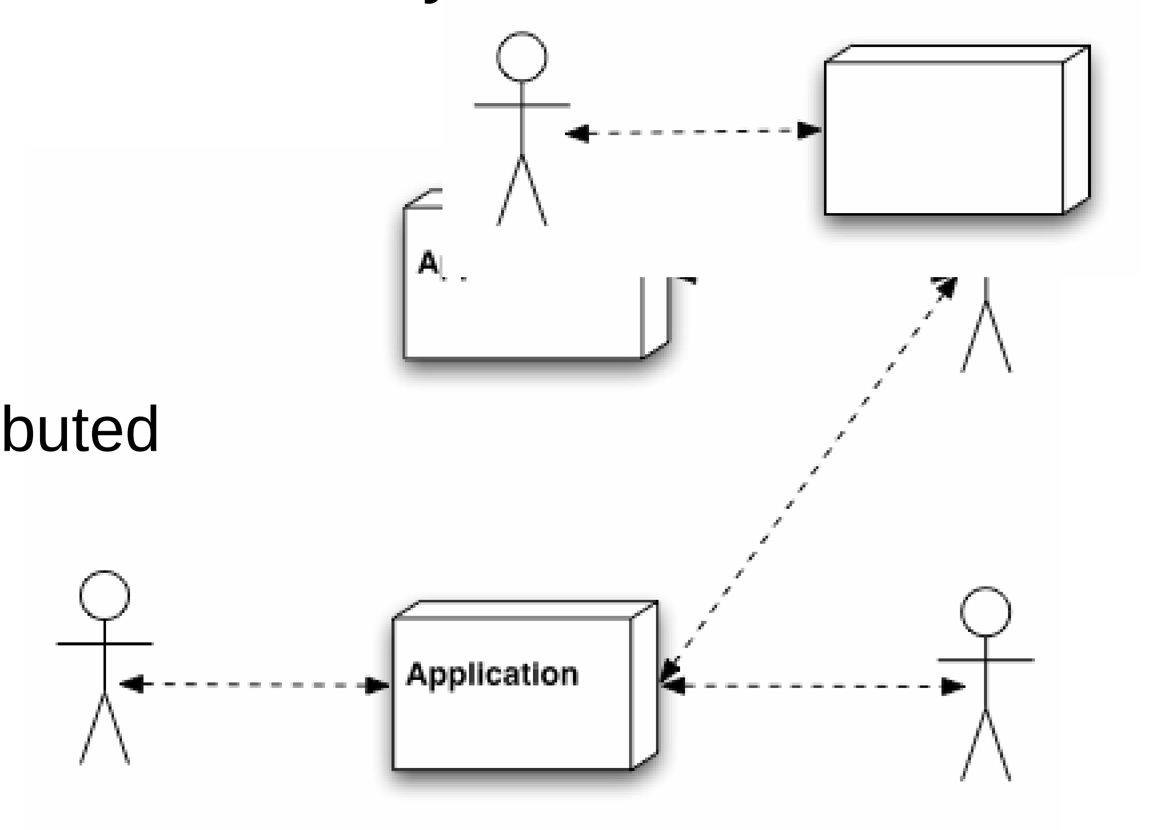
- Introduction
- Requirements Analysis
- Related Work
- Implementation
- Future Work

Introduction

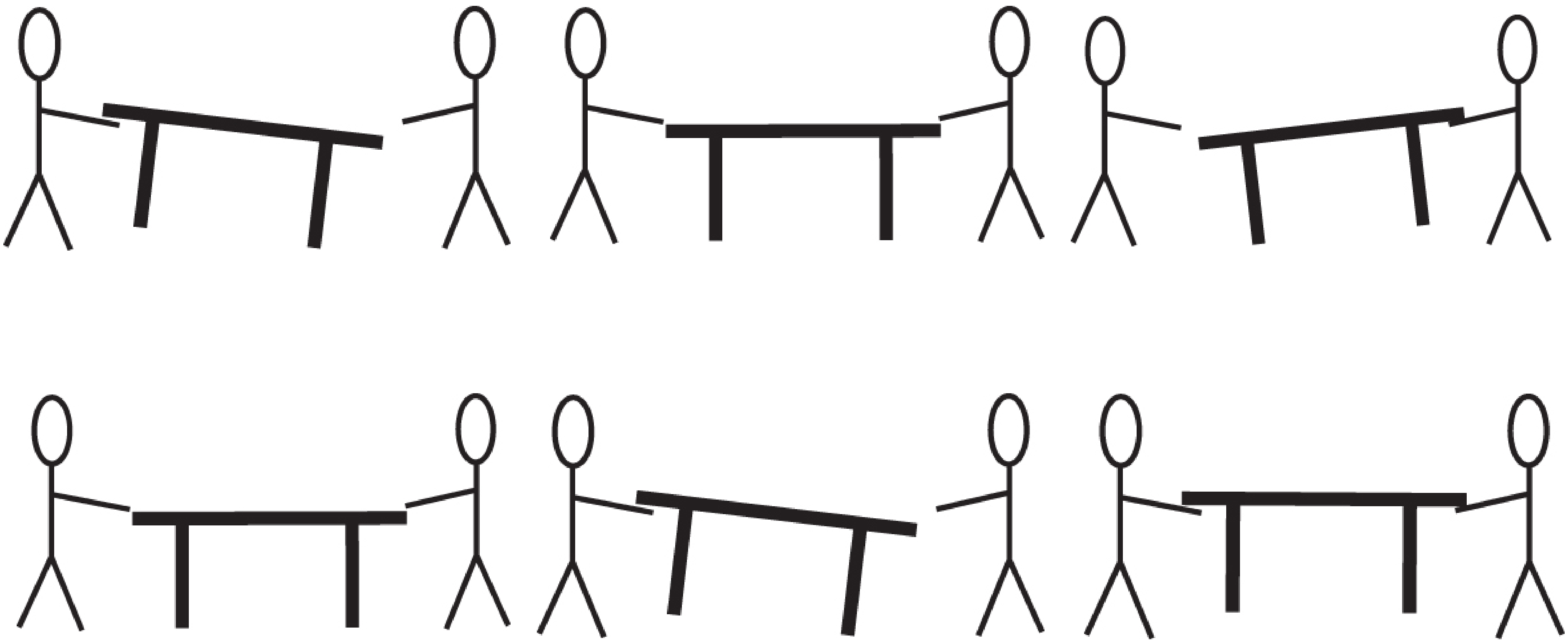
- What are UAR user interfaces?
- What is the problem space for such user interfaces?
- What design issues do those problems precipitate?

Introduction - Concepts

- Ubiquitous Augmented Reality user interfaces
  - Multi-user
  - Multi-device
  - Multi-modal
  - Mobile and distributed



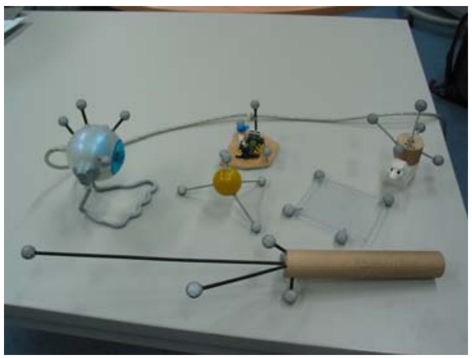
Introduction - Collaboration

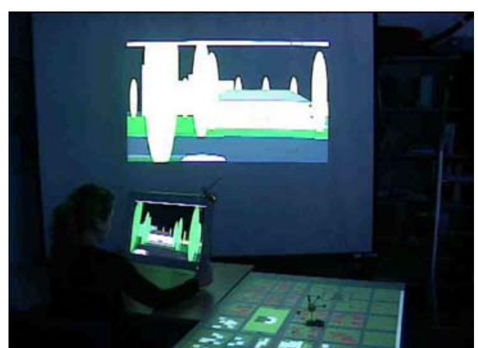


Co-allocated vs. Collaborative working

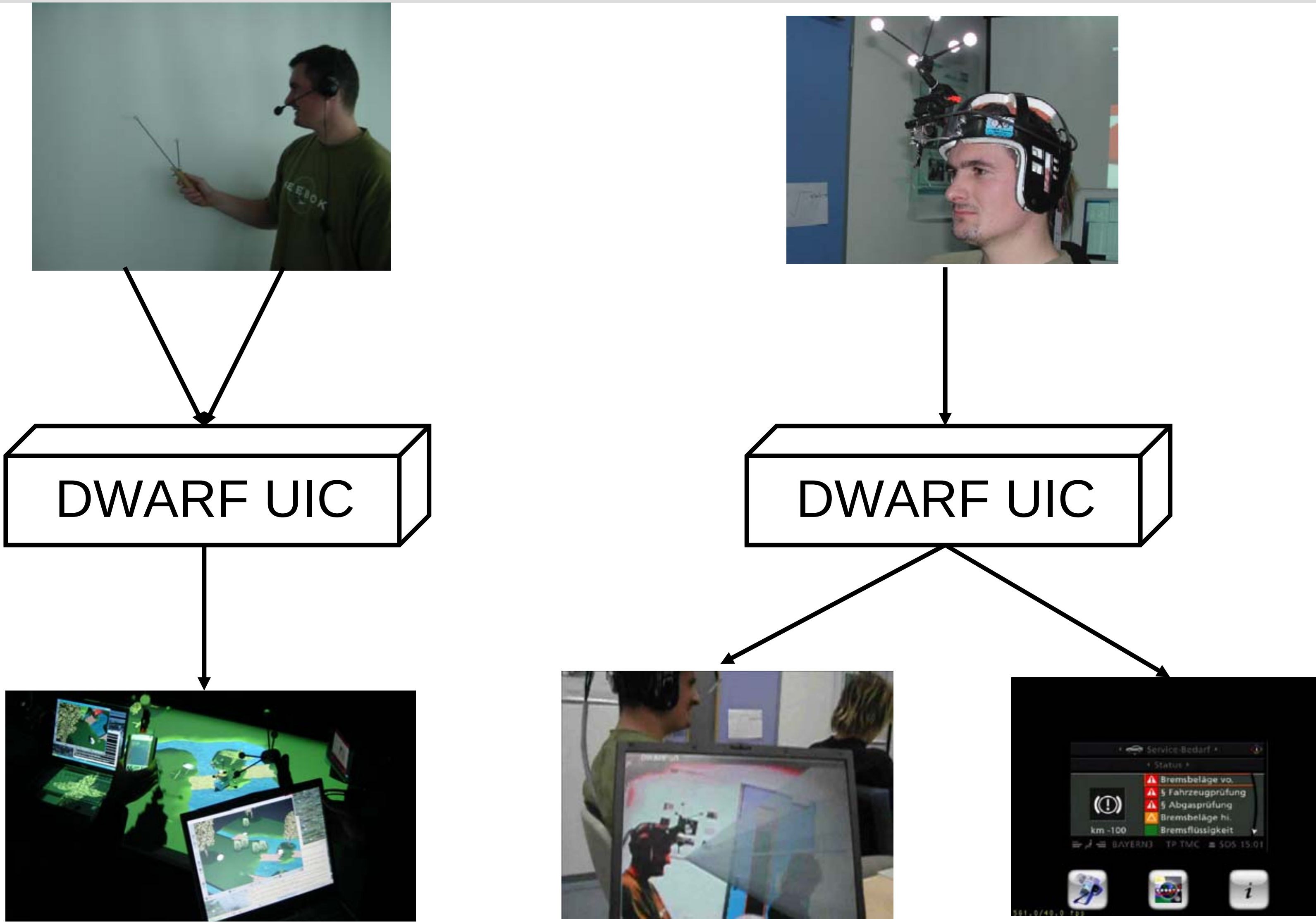
Introduction - I/O adaption

- UAR user interfaces incorporate new devices
  - Special purpose input devices.
  - Multimedia output.





Introduction - Multimodal Integration



Introduction - Runtime Prototyping

- Variety of I/O devices
- Dynamic system setups
- Non standardized interaction techniques

- Experiments with interaction techniques must be carried out
- Changing the connectivity structure at runtime

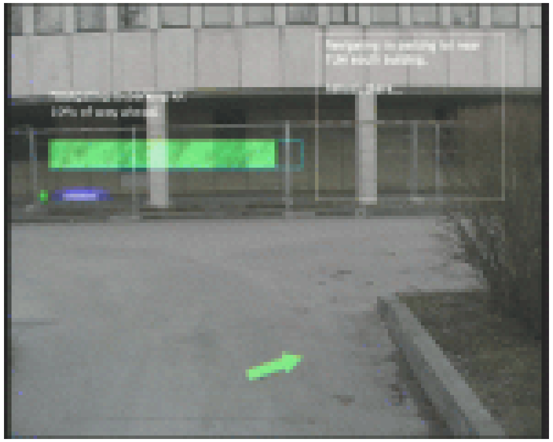
$\Longrightarrow$ Runtime Prototyping

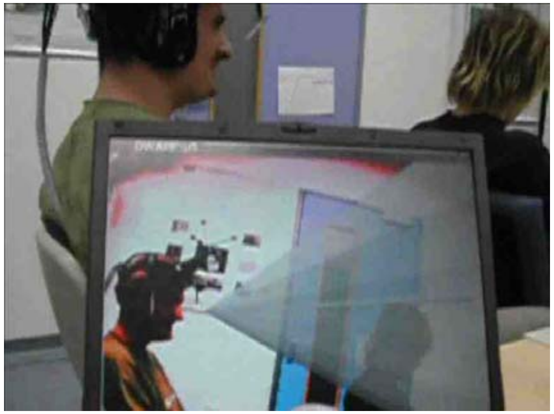
Outline

- Introduction
- Requirements Analysis
- Related Work
- Implementation
- Future Work

Requirements Analysis

- The requirements have been gathered throughout different projects:
  - TRAMP.
  - SHEEP.
  - ARCHIE
  - CAR.





Requirements Analysis

- Functional Requirements:
  - **Adapt I/O components**. The control component is the glue that holds together the complete UI.
  - **Input fusion**. To deal with different modalities the component must be able to **integrate multi-modal** input.
  - **Output fission**. Generate content for multiple output components.
  - **Input Recognition.** Disambiguate input from inter-social communication.
  - **Handle Privacy.** Differentiate between public and private information.
  - **Formal model** to describe UI behavior is needed that can be executed, modified and stored persistently.

Requirements Analysis

- Non - Functional Requirements:
  - **Availability**. If the UIC fails the whole system gets unusable.
  - **Robustness**. New users will make errors in the usage of the system.
  - **Reliability**. The same interactions must always produce the same results.
  - **Responsiveness**. For usability reasons the user must get immediate feedback whether an interaction succeeded or not.
  - **Scalability** due to steep increasing interpretation and management effort.
  - **Flexibility** to deal with inherently dynamic setups and changing I/O components.

Requirements Analysis

- Pseudo Requirements:
  - **DWARF** is the target environment and the developed component must be able to communicate with other services.

Outline

- Introduction
- Requirements Analysis
- Related Work
- Implementation
- Future Work

Related Work

- Interaction Management
  - Quickset
  - Unit
  - MetaDESK
  - Papier-Mâché
  - DART
- Petri Net vs. Finite automata
- Runtime Prototyping

Related Work: Quickset

- Quickset: Cohen et.al

Oregon Institue of Science and Technology

System for collaborative, multi-modal planning of tactical military simulations.

+ Powerful integration of speech, gesture and web-based input.

+ Very robust resolving disambiguites using AI techniques.

- Rigid architecture heavily application dependent.

- System can not be used in other setups.

QuickTime™ and a
Cinepak decompressor
are needed to see this picture.

Related Work: Unit

- Unit: Alex Olwal, Columbia University 2002

- Framework for the design of flexible interaction techniques.

- Abstraction layer between I/O devices and application.

- Units form a graph that allows the programmer to develop powerful interaction techniques.

+ Flexible data manipulation.

+ Units are reusable.

- No clear differentiation between discrete and continous data.

- Developers have to deal with I/O device‘s details.

QuickTime™ and a
Cinepak decompressor
are needed to see this picture.

Related Work: MetaDESK

- MetaDESK: Brygg Ulmer et.al., MIT 1997

Groundbreaking system in the field of TUIs. The DESK is a illuminated table enriched with special purpose tools (TUIsf) for urban planning.

+ Lots of creative tangible interaction and presentation techniques.
- Software architecture is application specific.

Related Work: Papier-Mâché

- Papier-Mâché:

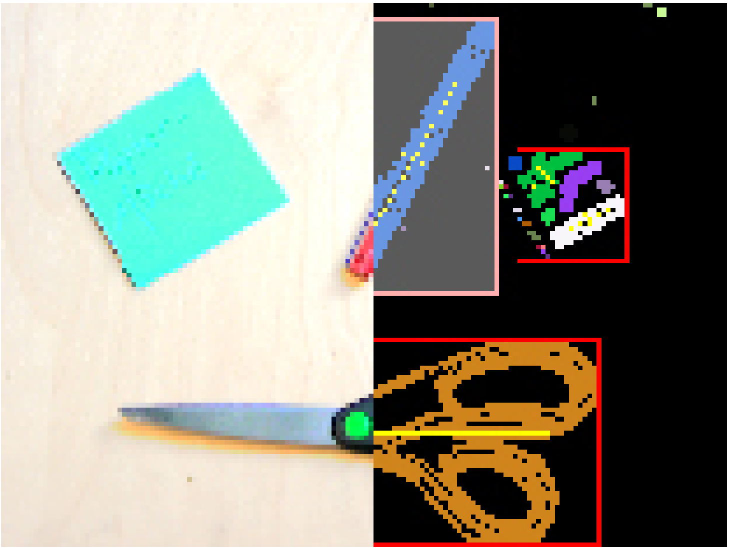
A Toolkit for developing TUIs. Using computer vision, electronic tags and barcodes.

+ Provides a API for TUI based systems.

+ Includes a variety of out of the box recognition algorithms.

- Code based approach.

- Only focuses on TUIs.



Related Work: DART

- DART:

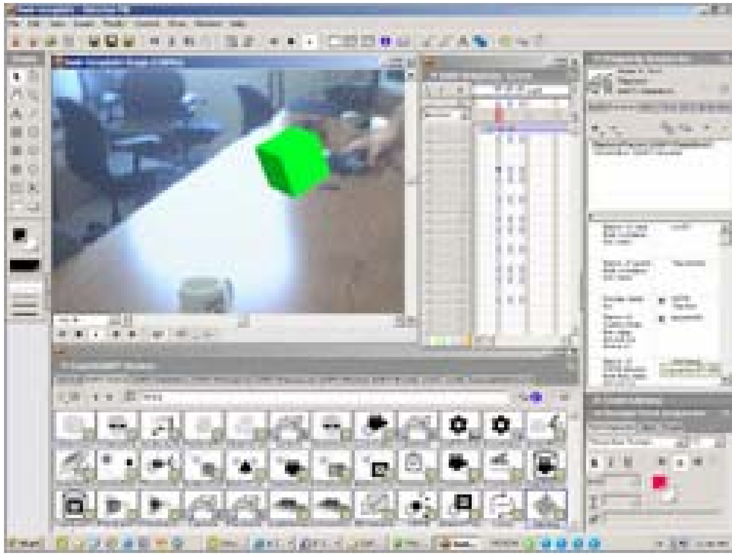
A toolkit for AR applications using a classic multimedia design tool (Macromedia Director).

+ Very easy to create content and application logic for non-programmers.

+ Director is already well-know and provides powerful means to design UIs.

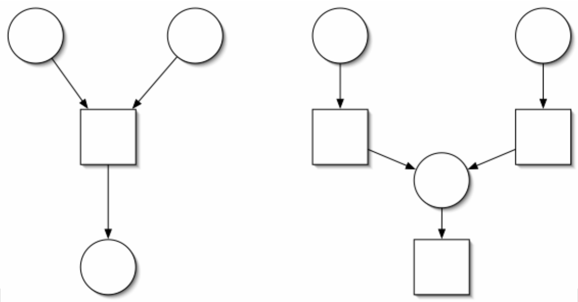
- Interactions are very limited.

- Not changeble at runtime.



RW: Petri Nets vs. Finite Automata

- FNA:
  - FNAs are used to model workflows (navigation, repair instructions).
  - One active state. Step by Step execution.
  - Very diffucult to model concurrent and multi-user situations.
  - Low learning threshold
- Petri Nets:
  - Introduced to model concurrent and distributed systems.
  - Powerful mathematical model
  - Meets requirements for distributed, multi-user and multi-modal systems.
  - High ceiling



Related Work: Runtime Development

- Squeak:
  - Multimedia design and development environment for educational purposes. Fully tweak-able.
  - Very easy to develop interactive graphical applications. Even kids can do it.
  - Limited to the classic WIMP-desktop.

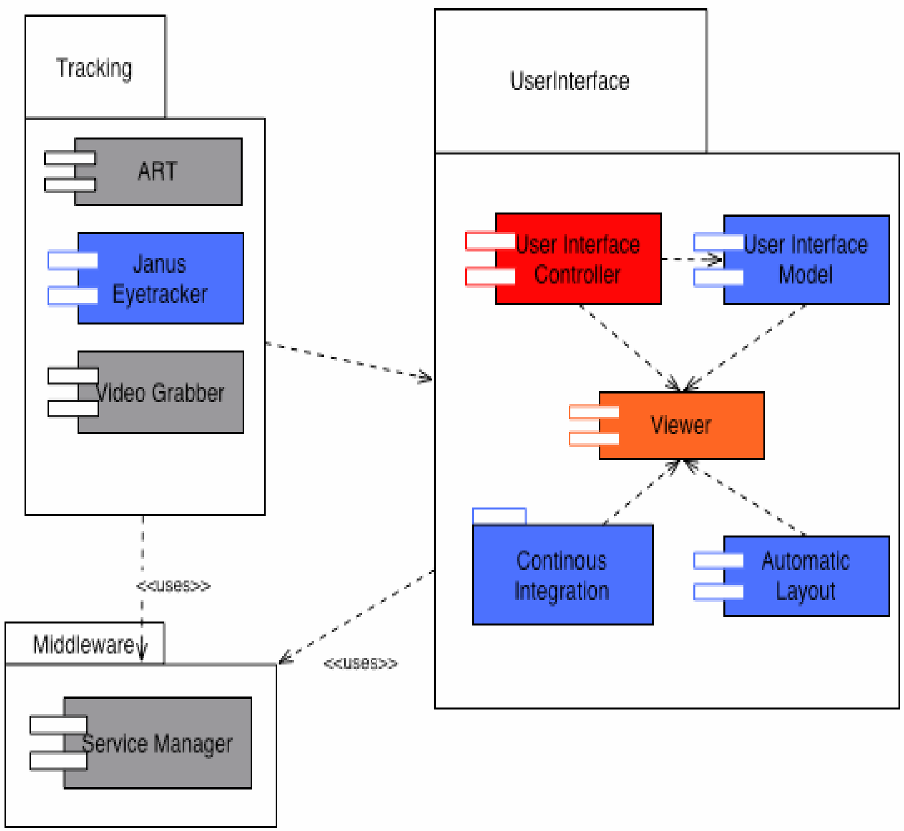
QuickTime™ and a
MPEG-4 Video decompressor
are needed to see this picture.

Outline

- Introduction
- Requirements Analysis
- Related Work
- Implementation
- Future Work

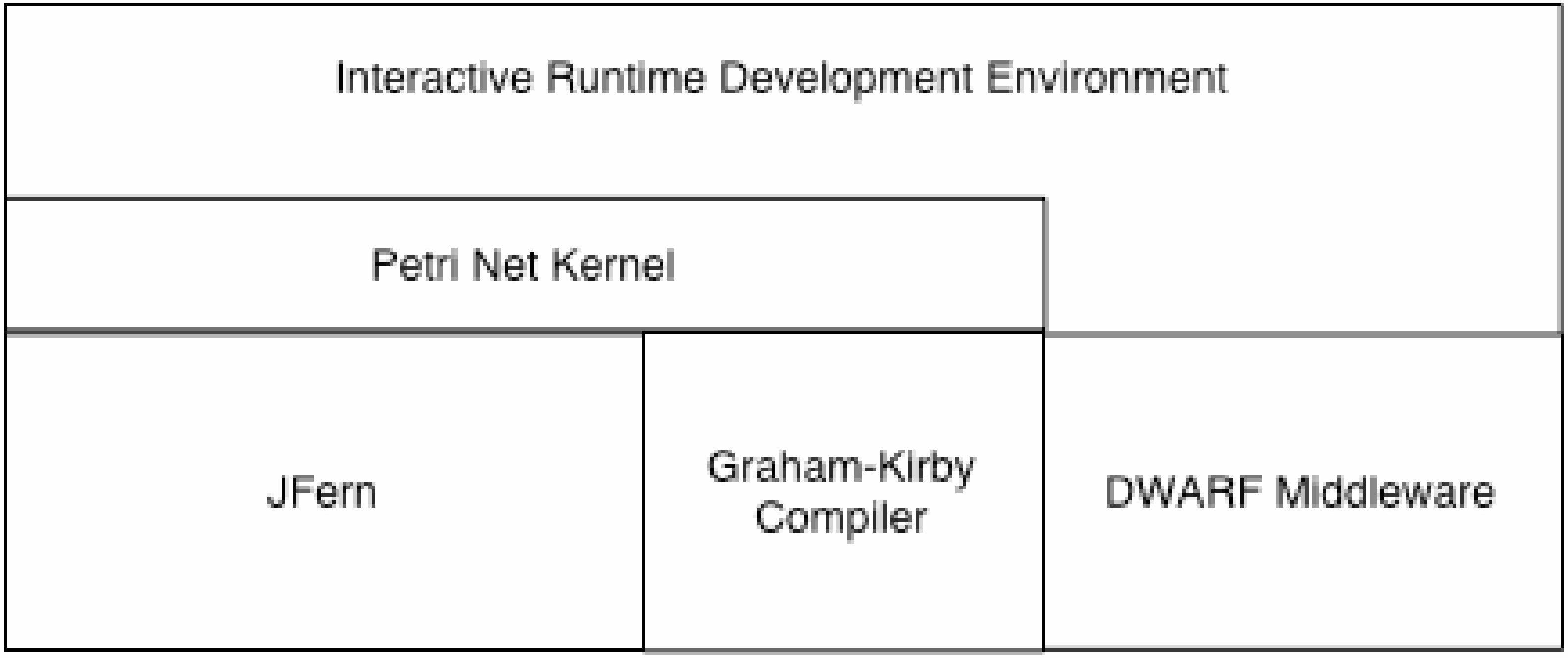
Implementation

- What I implemented in this thesis:
  – Interaction Management component based on DWARF and Petri Nets.
  – A runtime development environment for that component.

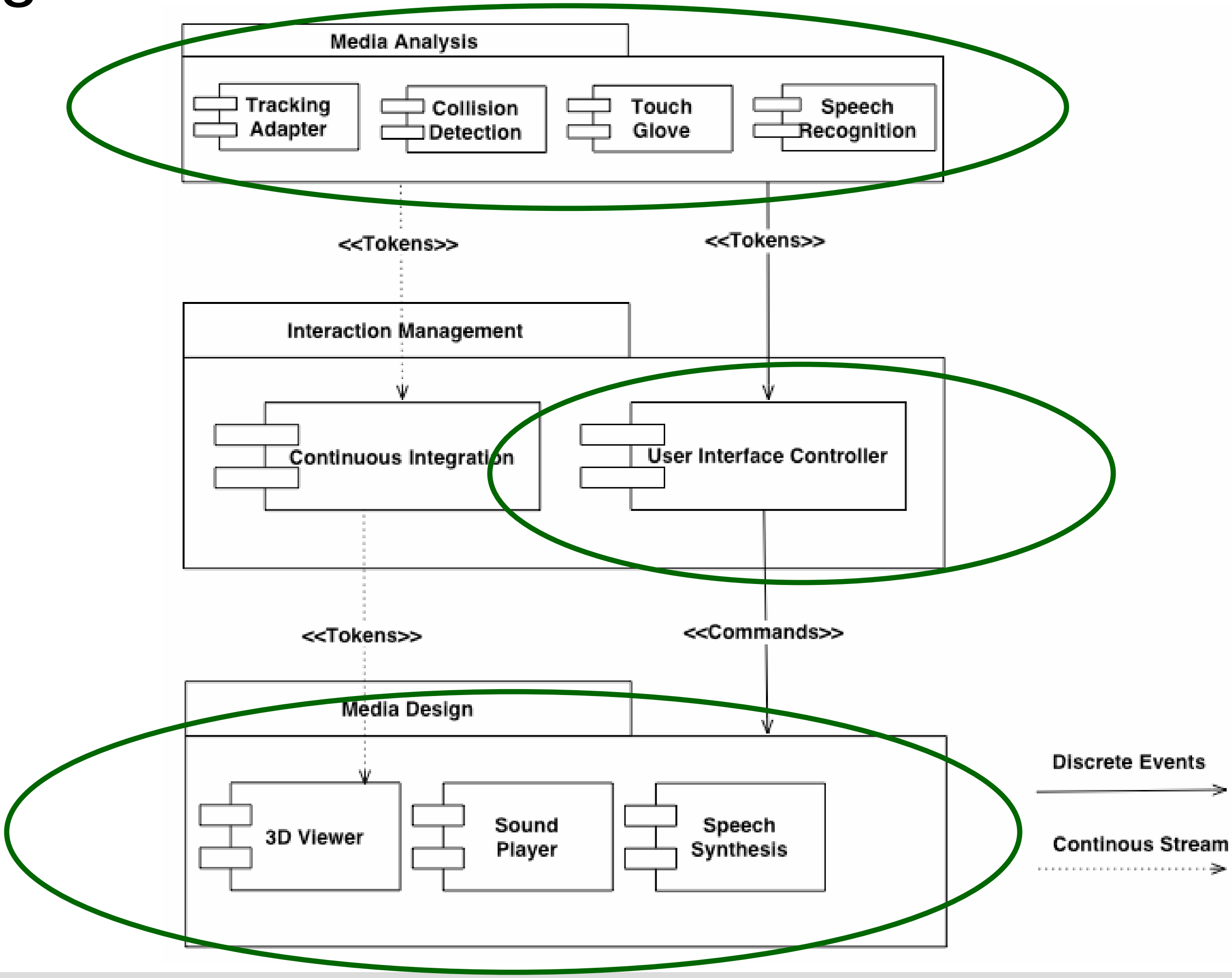


Implementation

- Layering and 3rd party software
  - DWARF, Jfern, Graham-Kirby Compiler

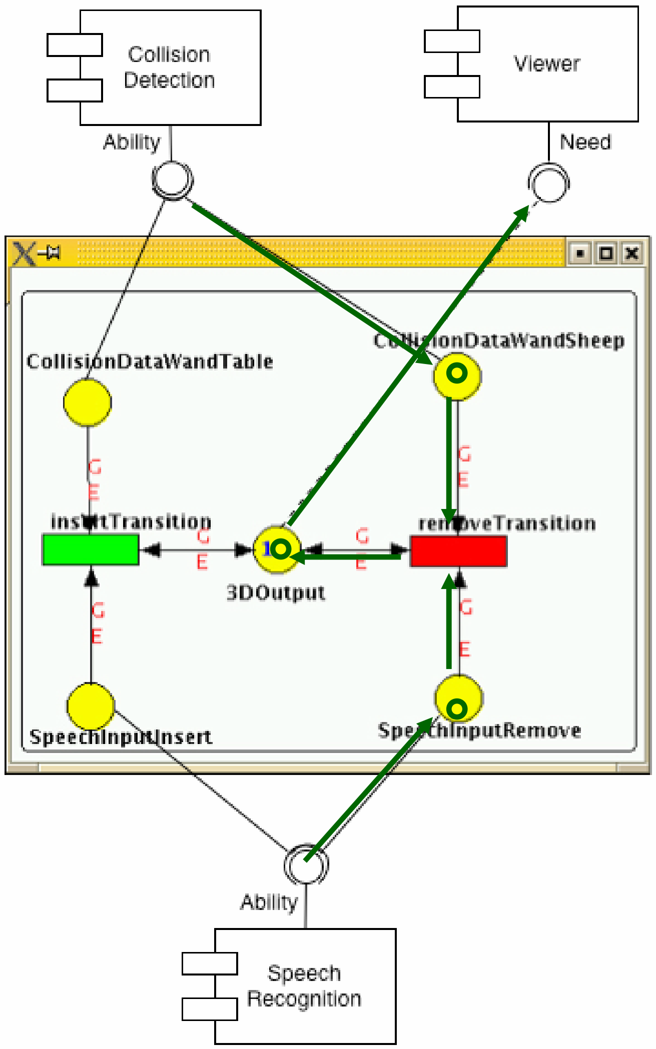


- Integration with DWARF UI architecture



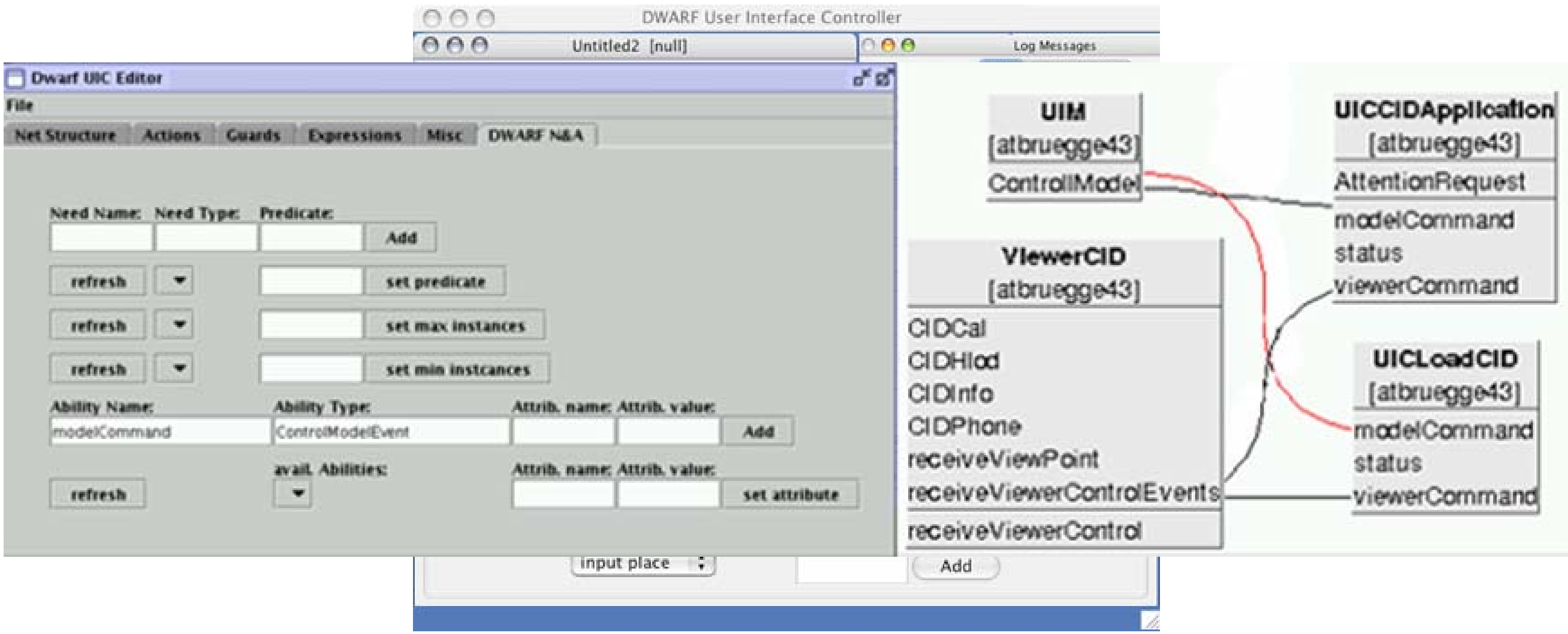
Implementation: Interaction Management

- Multi-modal integration
  - Input components emit tokens
  - Data is analyzed and modified inside Petri nets transitions
  - Commands are sent out to output components



Implementation: Runtime Prototyping

- Runtime development
  - Net structure modifications
  - Dynamic code modification
  - Connectivity management



Implementation: Runtime Prototyping

- Results: Mini-Sheep and CAR UI

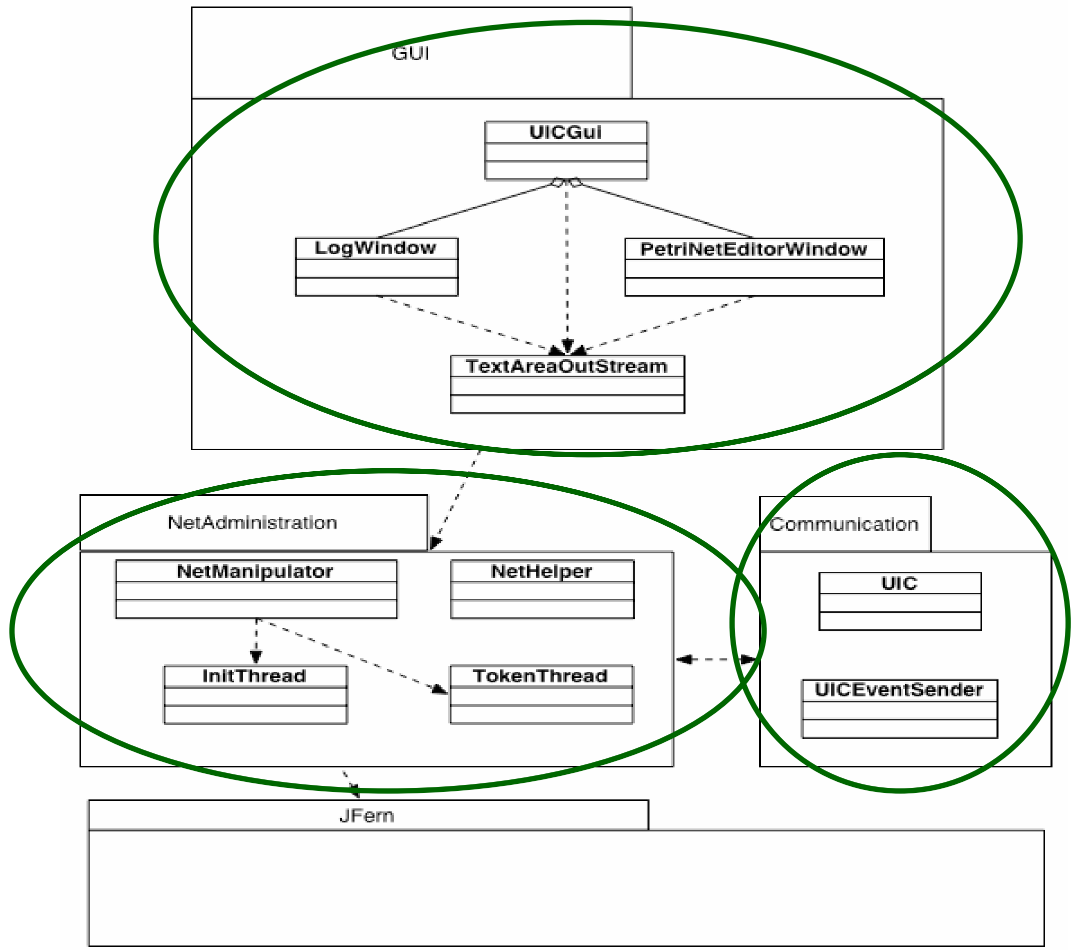
Implementation: Object Design

- UIC Implementation Details

- GUI
  - Visualize Petri Net
  - Add/Remove tokens
  - Controls for Editing PN and N&A
  - Logging and debugging Output.

- NetAdministration
  - Execute Petri Net execution
  - Modify net structure.
  - Compile guards and actions.
  - Modify Needs & Abilities

- Communication
  - Handle startup and communication
  - Receive & send structured events.



Outline

- Introduction
- Requirements Analysis
- Related Work
- Implementation
- Future Work

Future Work

- Improve UI of development environment
- Add convenience functionality
  - Palettes
  - Toolbars
  - Repository of interaction atoms.
- Programming by example
- Authoring within Augmented Reality.

Future Work II

- Extensions to the DWARF UI architecture:
  - User model.
  - Improved recognition techniques and multi-modal integration using Bayes nets and hidden Markov chains.
  - API for device integration.

Outline

- Introduction
- Requirements Analysis
- Related Work
- Implementation
- Future Work

Questions

Any Questions ?

Thank You!