ADE - An Architecture Development Environment for Virtual and Robotic Agents

Virgil Andronache and Matthias Scheutz

Artificial Intelligence and Robotics Laboratory

Department of Computer Science and Engineering

University of Notre Dame

Notre Dame, IN 46556, USA

http://www.nd.edu/~airolab

Overview

- Why another agent architecture development environment?
- Overview of ADE
- Example of using ADE with a robot
- Status quo and future work
- References to our work on agent architectures

A Case for Distributed Agent Architectures

- Suppose you have to design an architecture for a robot that needs to operate in real-time with the following restrictions (at any given time):
  - The robot need to process several different, computationally intensive input channels from onboard sensors in parallel and be react to them quickly (e.g., if a case of emergency is detected)
  - Multiple onboard computers are involved in the processing, but computational resources of each are very limited
  - Additional offboard processing needs to be done (e.g., using wireless ethernet connections)

SAS or MAS, that's the question

- What system would be appropriate for this kind of setup - *SAS* (single agent systems) or *MAS* (multi agent systems)?
- *Claim:* neither one alone is sufficient, but rather a combination of both is required
- A brief note on terminology:
  - by “SAS-agent” we mean an agent in the sense of an autonomous robot
  - by “MAS-agent” we mean an agent in the sense of a computational component of a distributed system

Multi-Agent Systems (MAS)

- Multiagent systems provide the infrastructure for the distributed agent-based systems:
  - registry, white page, yellow pages, communication protocols, load balancing, movability of agents, etc.
- Some examples:
  - Jade (Bellifemine et al. 2000)
  - Retsina (Sycara et al. 2003)
  - ZEUS (Nwana et al. 1997)
  - AgentBase (http://www.sics.se/~market/toolkit/)

Single-Agent Systems (SAS)

- Single-agent systems (or toolkits) provide components of agent architectures, for example:
  - *Cognitive architectures:* knowledge representation, reasoning, learning (e.g., chunking)
  - *Behavior-based architectures:* behavior representation, sensory processing, learning (e.g., reinforcement)
- Some examples:
  - SimAgent (Sloman and Logan 1999)
  - Saphira (Konolige 2002)

Integrate SAS and MAS Functionality

- Want to implement computationally demanding realtime architectures for complex robots
  –> combine tools from SAS and MAS systems!
- Specifically, want to distribute components of "SAS-agents" using "MAS-agents" (i.e., the components of MAS systems)
- Need:
  - framework in which agent architectures can be expressed to allow implementation of different architecture paradigms (e.g., want to be able to implement rule-based systems as well as behavior-based systems)
  - multi-agent system that supports real-time interactions

university of notre dame
computer science and engineering

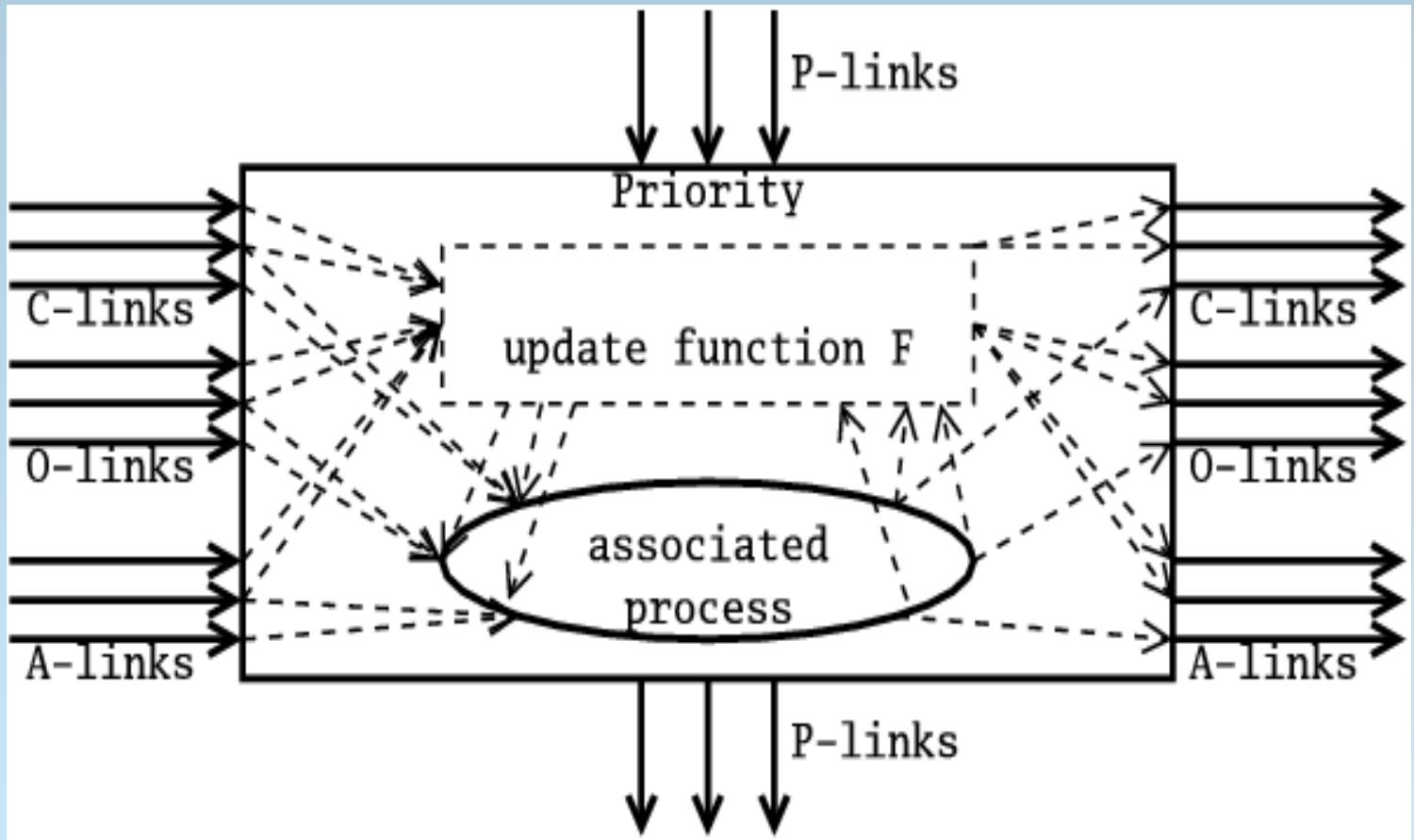
Our Proposed Solution: ADE

- ADE - "APOC Development Environment", which builds on and combines:
  - APOC – a generic agent architecture framework, in which other architectures can be expressed in a unified way
  - AGES – a distributed MAS system with emphasis on real-time robotic applications
- JAVA-based for platform-independence, threading, and RMI
- Multi-user distributed GUI for collaborative design, testing, and run-time supervision

SAS Aspects: the Agent Architecture Framework APOC

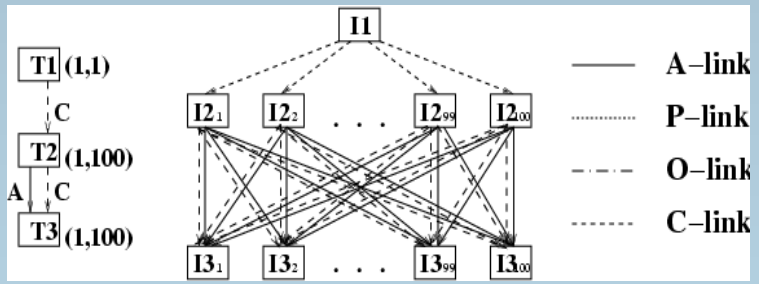
- APOC ("Activating", "Processing", "Observing", "Components")

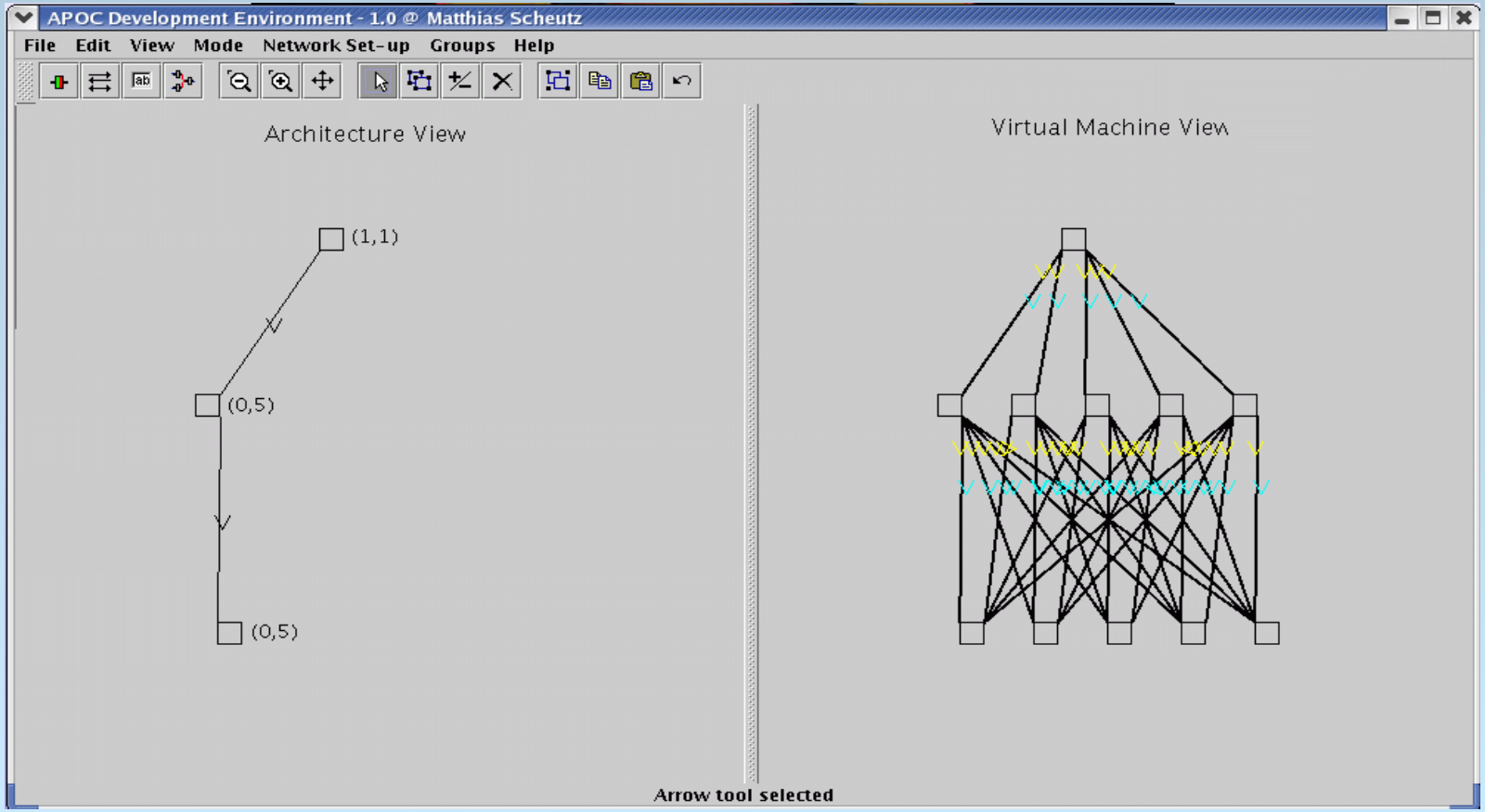
- One basic component type
- Four basic link types:
  - A-link
  - P-link
  - O-link
  - C-link



SAS Aspects: Expressing other Architectures in APOC

A neural network architecture type in APOC and one particular instance of it





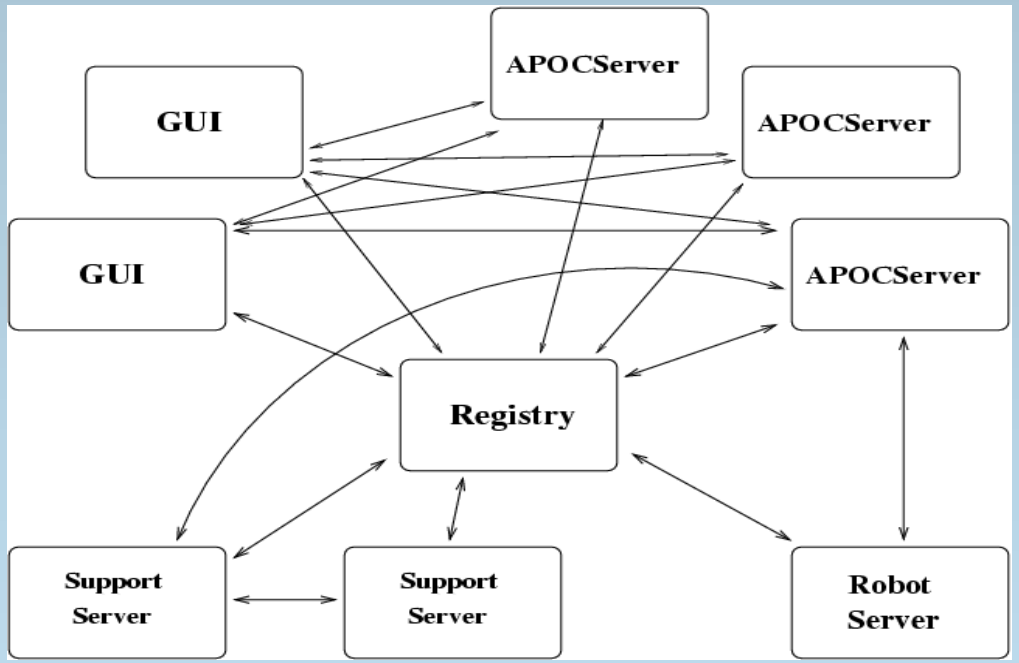
The ADE GUI

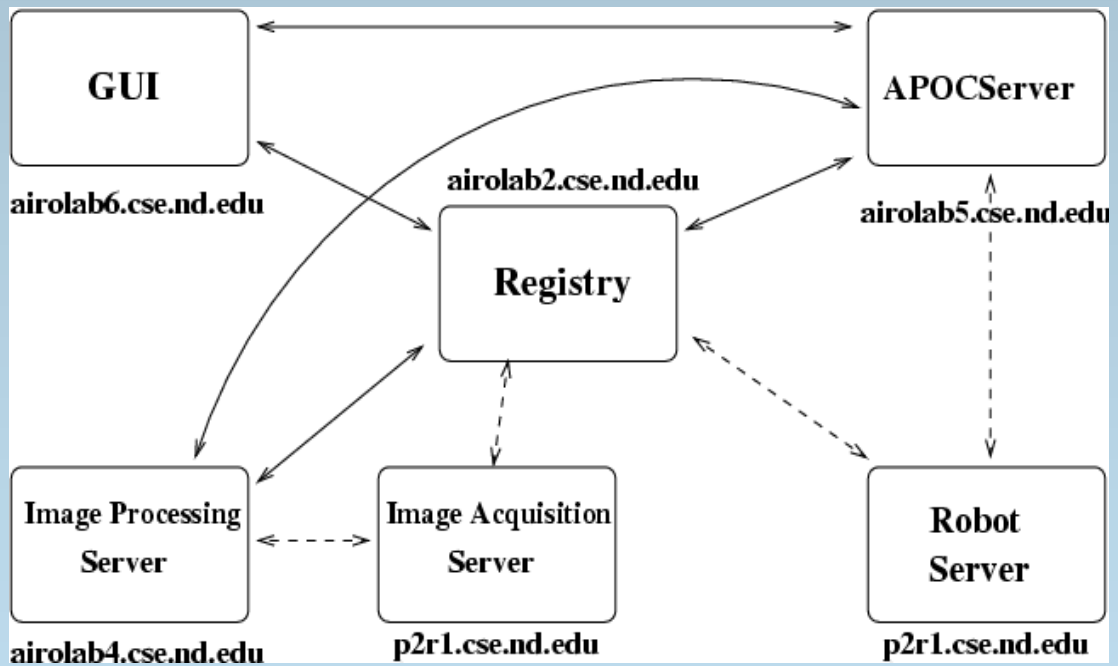
- Left: *the architecture view*
- Right: *the virtual machine view*

MAS Aspects: Servers in ADE

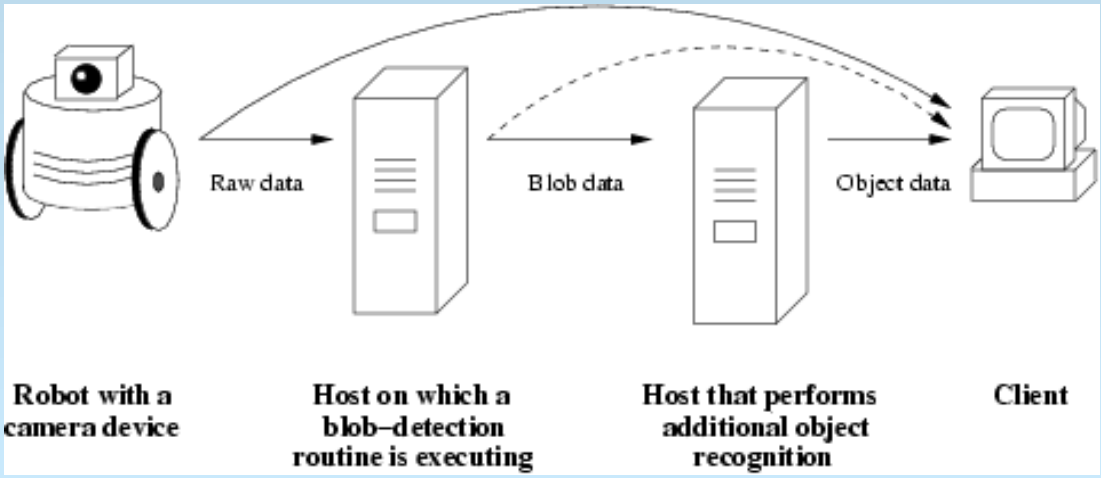
- A *server* in ADE is a computational unit that represents a computationally independent resource of the ADE system:
  - *APOC server:* the basic APOC virtual machine with capabilities for instantiating and deleting new components and links
  - *GUI server:* a visual resource which allows the user to view the architecture and its distributed instantiation in the virtual machine
  - *Agent server:* a representation of a virtual or robotic agent within ADE with access to its sensors and effectors
  - *Utility server:* provide a system-wide service which may be needed by one or more APOC components

MAS Aspects: Single-Robot Setup

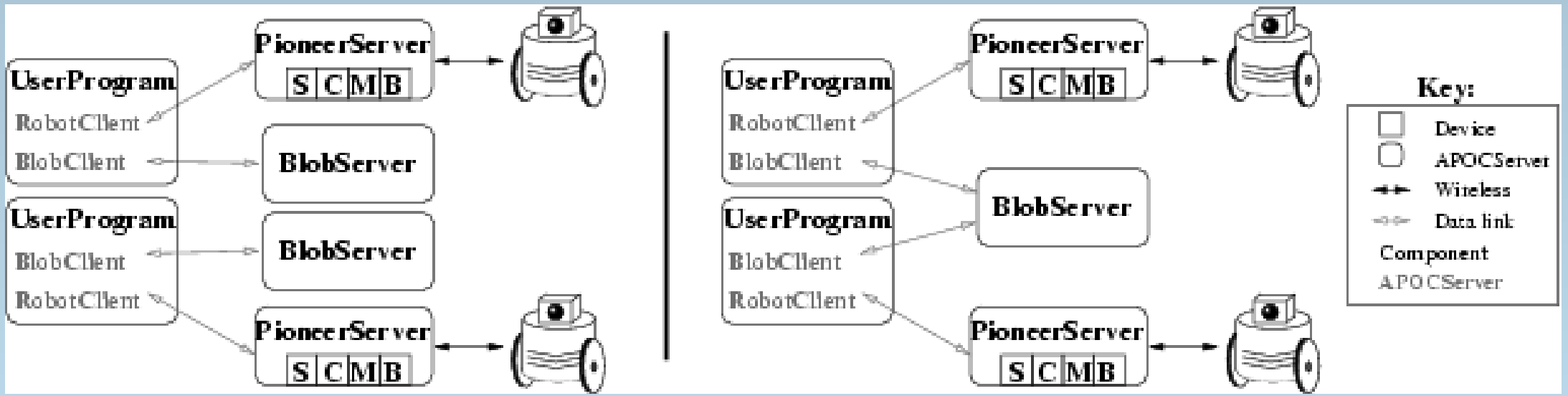




Generic and specific ADE setup for a distributed robotic architecture using one registry and various servers



MAS Aspects: Multi-Robotic Setup

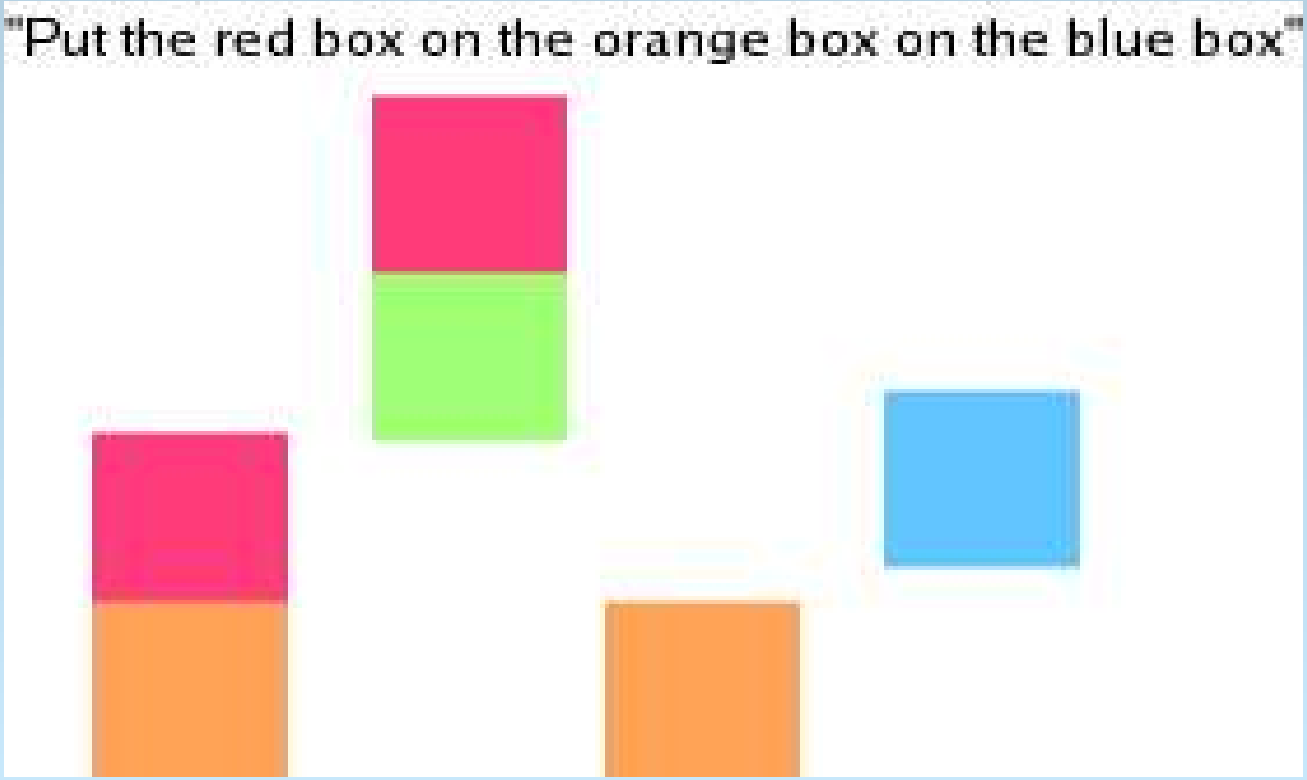


ADE setup for a distributed robotic architecture for tracking using two instances of the *BlobServer* for color blob detection

ADE setup for a distributed robotic architecture for tracking using one shared instance of the *BlobServer* for color blob detection

Example: Human Reference Resolution

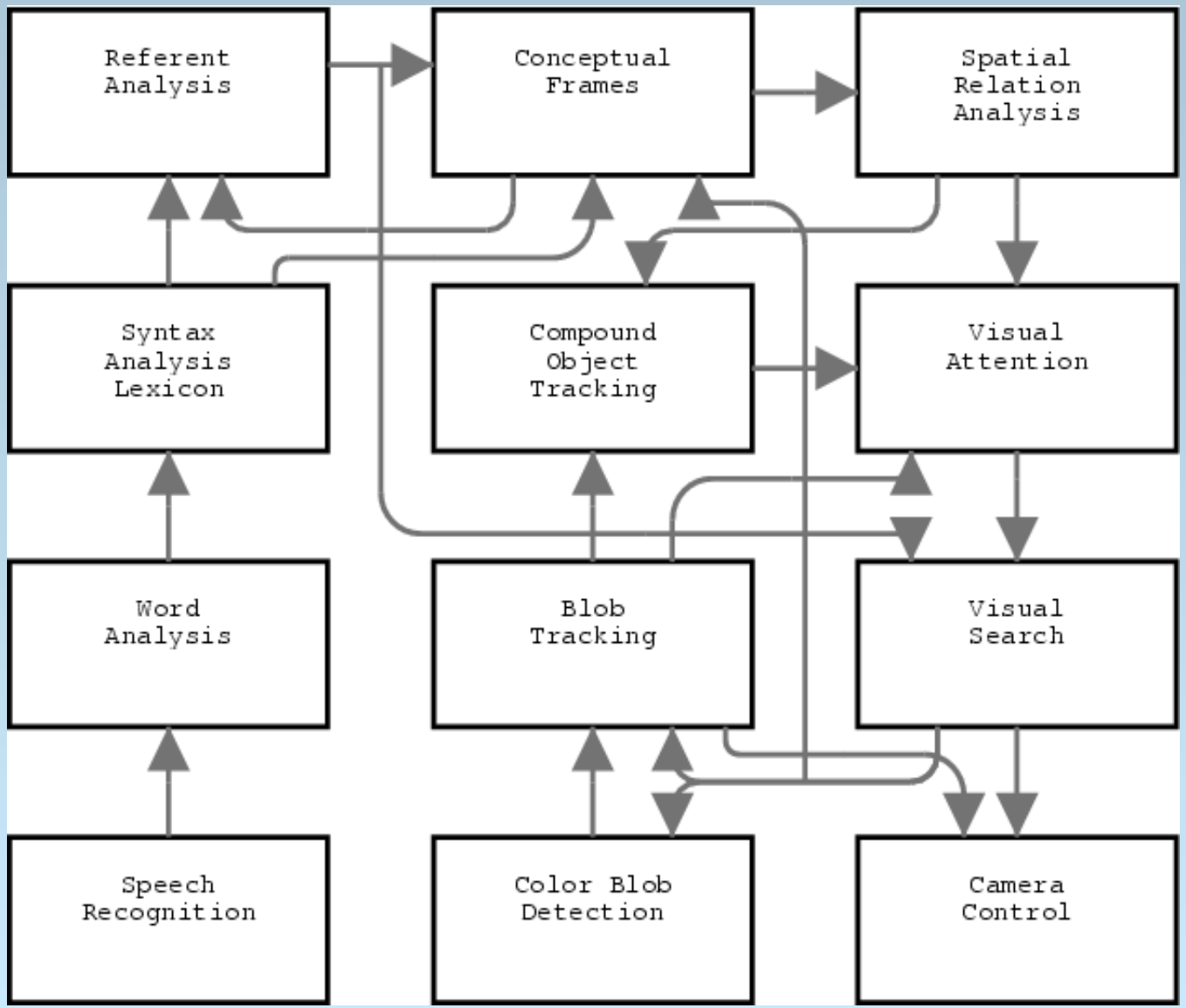
- Model in real-time the processes in humans when they try to make sense of ambiguous spoken sentences like this using visual cues:



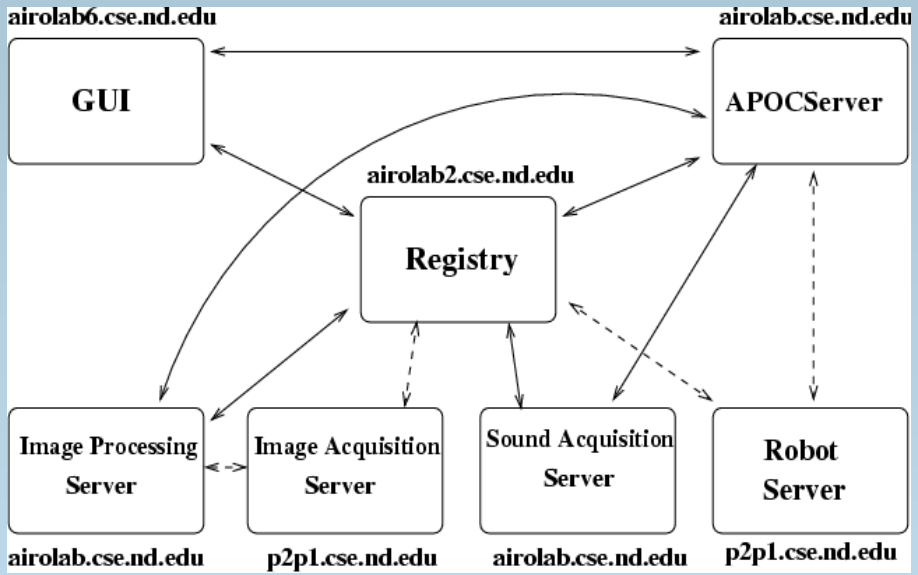
Human Reference Resolution: Processing Architecture

High-level view of the processing architecture:

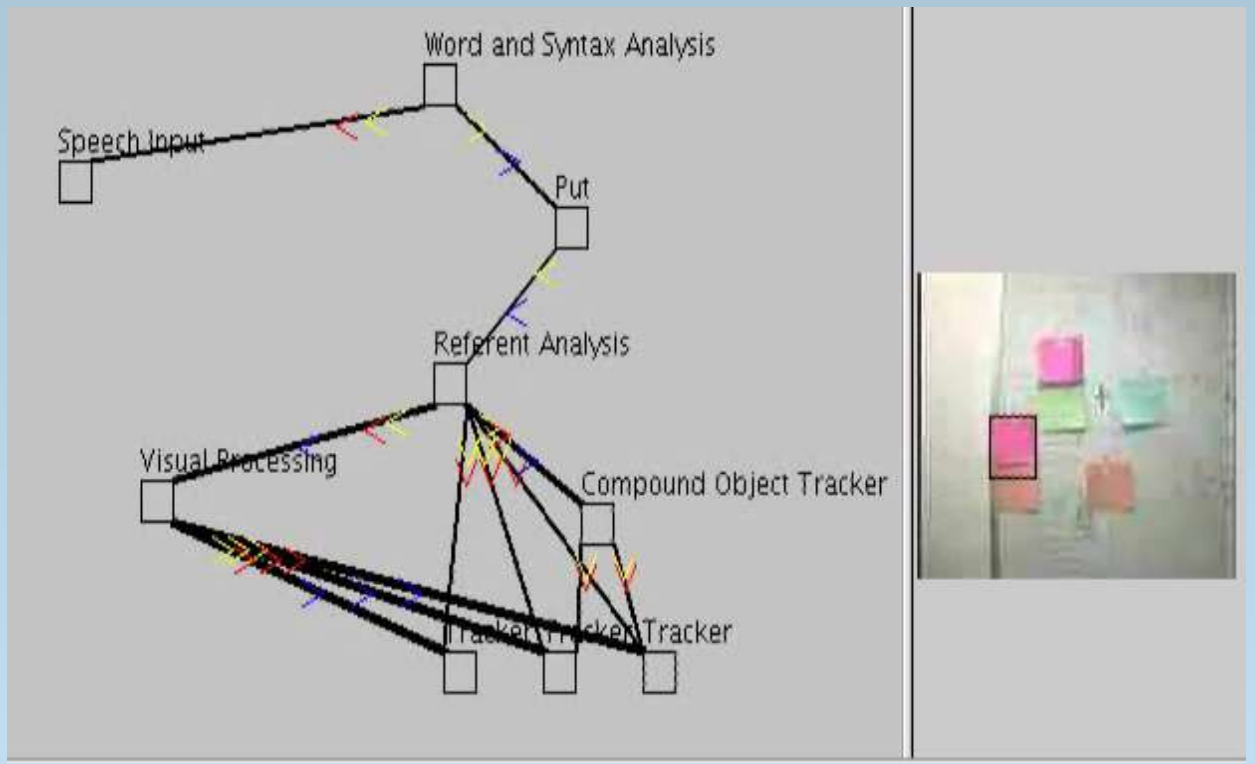
- boxes depict components that are active in parallel
- arrows represent flow of information through the architecture)



In ADE ...



- 4 hosts (including robot)
- dashed lines are wireless
- solid lines are ethernet
- mixed Solaris/Linux



- C-links (yellow)
- A-links (blue)
- O-links (red)

Camera View

Attention Box

nd lab
university of notre dame
computer science and engineering

Status Quo and Future Work

- First prototype ready, so far used
  - for several research projects ("ND Hexapod", "People Tracking", "Multi-robot ball following", "Hybrid agent architectures", etc.)
  - for teaching (CSE 498F "Behavior-based Robotics" in Spring 04)
- More debugging, monitoring, and testing tools (+ GUI) in the works (+ better support for real-time control)
- More architectures to be translated in APOC, which will then be available in ADE (e.g., SOAR, ICARUS)
- Implementation of ADE on special, highly parallel PIM hardware (currently in prototype phase at ND)

References to Our Recent Work on Agent Architectures

- Andronache, V. and Scheutz, M. (2004) "Integrating Theory and Practice: The Agent Architecture Framework APOC and its Development Environment ADE" In *Proceedings of Autonomous Agents and Multiagent Systems 2004*.
- Scheutz, Matthias (2004) "The APOC Framework for the Comparison and Evaluation of Agent Architectures". *Proceedings of AAAI 2004 Workshop on Intelligent Agent Architectures.*
- Scheutz, Matthias (2004) "APOC - An Architecture Framework for Complex Agents". In Darryl Davis (ed.)*Visions of Mind*. Idea Group Inc.
- Andronache, Virgil and Scheutz, Matthias (2003) "Growing Agents - An Investigation of Architectural Mechanisms for the Specification of 'Developing' Agent Architectures". In *Proceedings of FLAIRS2003*, AAAI Press, 22-26.
- Andronache, Virgil and Scheutz, Matthias (2003) "APOC - a Framework for Complex Agents". In *Proceedings of AAAI Spring Symposium 2003*, Standford, AAAI Press, 18-25.
- Sloman, Aaron and Scheutz, Matthias (2002) "A Framework for Comparing Agent Architectures". In Proceedings of UKCI'02, 169-176.
- Scheutz, Matthias and Andronache, Virgil (under review) "Architectural Mechanisms for Dynamically Modifiable Behavior Selection Strategies in Behavior-Based Systems".
- ADE can be downloaded from `http://www.nd.edu/~airolab/software`