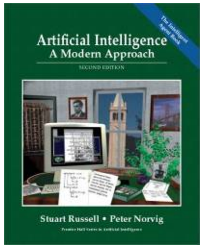
Introduction to Multi-Agent Programming

1. Introduction

Organizational, MAS and Applications, RoboCup

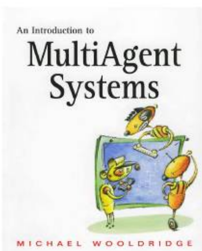
Alexander Kleiner, Bernhard Nebel

Lecture Material



Artificial Intelligence – A Modern Approach, 2nd Edition

by Stuart Russell - Peter Norvig



An Introduction to MultiAgent Systems

by Michael Wooldridge

Copies of the lecture slides as well as further information can be found on the Web at:

`http://informatik.uni-freiburg.de/~ki/teaching/ws0809/map`

Many illustrations have been taken from the above book.

Organizational

Lectures:

- **Time:**

  Tu 14:15 – 16:00

- **Lecturer:**

  Dr. Alexander Kleiner (kleiner@informatik.uni-freiburg.de)

Invited Talk:

- **Time:**

  Tu 13.1.2009 14:15

- **Lecturer:** Dr. Klaus Dorer (Whitestein Technologies, Donaueschingen)

- **Title:**
  “Applications of Multi-Agent Systems in Logistics”

Exercises:

- **Time:**

  Th 14:15-16:00

- **Organizers:**

  Christian Dornhege, Dapeng Zhang

- Exercises handed out on Thursdays, to be submitted the following week
- Programming tasks may be solved in groups of three students

Credit Requirements:

- Written Exam (max. 100pts)

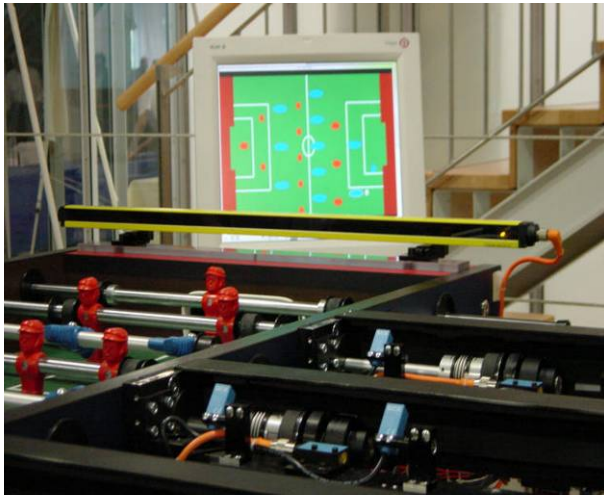
- Bonus marks for reasonably solved exercises and programming tasks (max. 30 pts)

Course Content

I. Introduction to Multi-Agent systems (today)
II. Societies of Agents
III. Fundamental Agent Architectures
IV. Search algorithms and Path-finding
V. Game Theory and MAS
VI. Agent Communication
VII. Common Sensing and World-Modeling
VIII. Multi-Robot Exploration
IX. Auctions and Cooperation
X. Case-studies RoboCup
XI. Learning in MAS
XII. Swarm Intelligence
XIII. GameAI: Solutions found in computer games

Foundations of Artificial Intelligence

- Action Planning: Theory and Practice
  - Fast planning systems (proven at int. competition)
  - Applications at airports and for lift systems
  - Theoretical results (see new Russell/Norvig)
- Qualitative Temporal-Spatial Reasoning
  - Theory and reasoning algorithms
  - Application in qualitative layout description
  - SFB
- Autonomous table soccer
  - Further developed to a market-ready product (Gauselmann Group)



Multi-Robot and Multi-Agent Activities

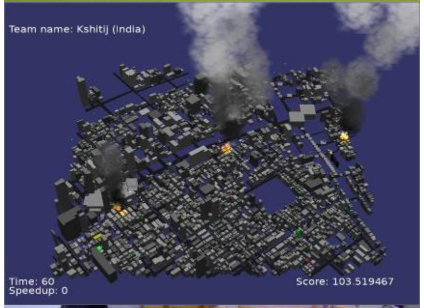
- RoboCup Soccer (CS-Freiburg)
  - Mid-sized robot team
  - World champion 1998/2000/2001

Subject of the exercises

- RoboCup Rescue Agent (ResQ Freiburg)
  - Large Multi-Agent-System for disaster relief
  - World champion 2004

- RoboCup Rescue Robot (Rescue Freiburg)
  - Heterogeneous team of rescue robots
  - Best Autonomy 2005/2006

- RoboCup Rescue Virtual Robots (Rescue Freiburg)
  - Virtual larger robot team
  - World champion 2006



What are Multi-Agent Systems (MAS)?

An MAS can be defined as a loosely coupled network of problem solvers that interact to solve problems that are beyond the individual capabilities or knowledge of each problem solver (Durfee and Lesser 1989).

These problem solvers, often called agents, are autonomous and can be heterogeneous in nature.

What are Multi-Agent Systems (MAS)?

Most importantly, the vision that *intelligence* emerges from complex interactions of multiple simple units ...

... individual heterogeneous capabilities for an efficient team

Characteristics of MAS

1. Each agent has incomplete information or capabilities for solving the problem and, thus, has a limited viewpoint
2. There is no system global control
3. Data is decentralized
4. Computation is asynchronous

(K. P. Sycara 1998)

What MAS are expected to do better?

- To solve Problems that are too large for a centralized agent with limited resources
  - distributed computing
- To reduce the risk of failure of a centralized system
  - Disaster mitigation / Urban Search And Rescue
- To keep legacy systems inter-connectable and inter-operational
  - Migration of outdated software
- To solve problems that can naturally be regarded as societies of autonomous components
  - Air-traffic control, Meeting scheduling

OOP (Object Oriented Prog.) vs. MAS

OOP	MAS
Objects are passive, i.e. an object has no control over method invocation	Agents are autonomous, i.e. pro-active
Objects are designed for a common goal	Agents can have diverging goals, e.g. coming from different organizations
Typically integrated into a single thread	Agents have own thread of control

Objects do it for free; agents do it for money. (Jennings et al. 1998)

Applications of MAS I
Computer Games



Real Time Strategy (e.g. Starcraft, Age of Empires)

→ group task assignment, and multi-agent path planning


First Person Shooter (e.g. Half Life 2, Splinter Cell)

→ character interactions, team formation, path planning, etc…

Simulations (e.g. The Sims)

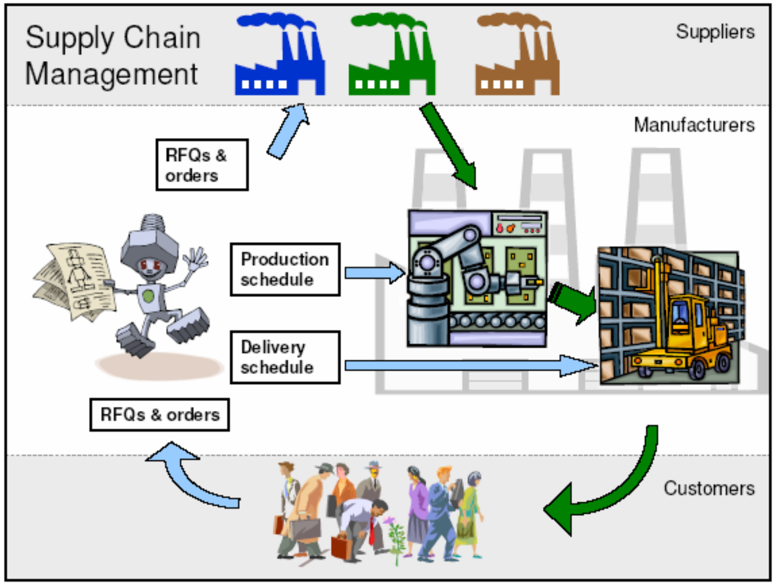
→ character interaction, utility maximization

Applications of MAS II

Supply Chain Management, B2B, Aircraft control

- Supply chain management
- B2B, Logistics

→ coalition formation problem, standardized communications, auctions



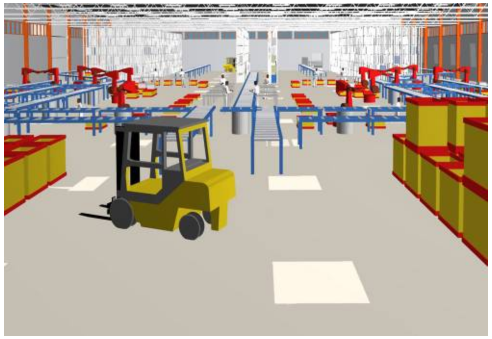
- Air traffic control

→ distributed sensing, auctions, ...

Applications of MAS III
Industry and Rescue

- Industry
  - car assembly, factory management
  - container terminal management

  → Task assignment, coalition formation, path planning



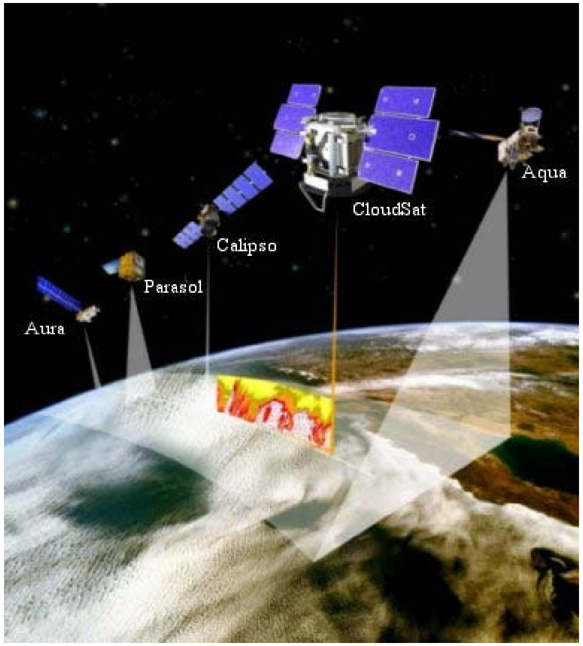
- Urban Search And Rescue (USAR)
  - distributed sensors
  - unmanned vehicles
  - First responder management

  → Decentralized sensing, task assignment, coalition formation, path planning

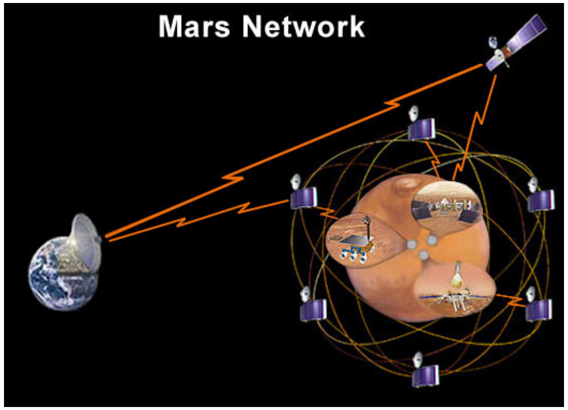
Applications of MAS IV
Space

Space Missions with multiple rovers

Space ship repair



Earth orbiters



Mars network

→ Decentralized sensing, task assignment, coalition formation, 3D path planning, and many more challenges ….

The RoboCup Project
Soccer and Rescue

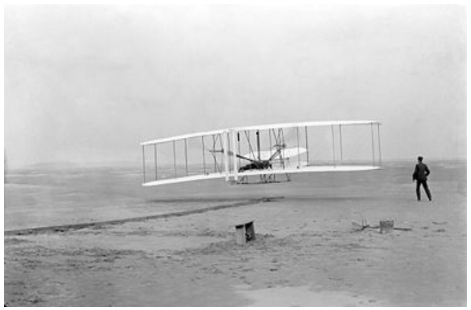
http://www.robocup.org/

The vision: By 2050, build a team of fully autonomous humanoid which win against human world champion under the official regulation of FIFA.

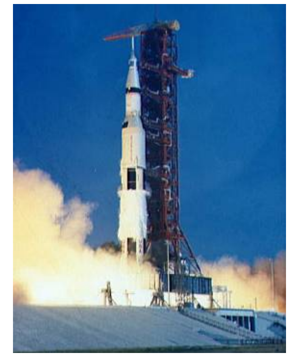
- Since 1997 annual competitions and workshops, since 2001 RoboCup Rescue
- A platform for project-oriented education in science and technology
- A standard problem for AI and robotics
- Technology transfer
- A landmark Project: challenging goal and spill-over of technologies

CS Freiburg vs. CMU, Seattle 2001

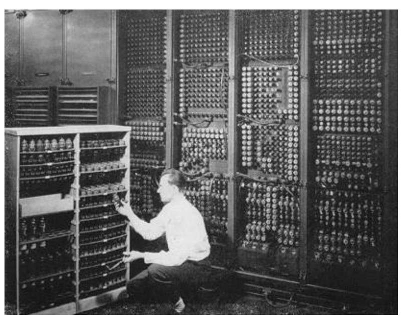
Some famous landmark projects: the Apollo program, computer chess



Wright Flyer 1903



NASA 1969



Eniac 1946

Deep Blue 1997

RoboCup Soccer

Example of successful team coordination

CS Freiburg vs. Osaka, Final, Seattle 2001

The RoboCup Project

Computer Chess vs. RoboCup

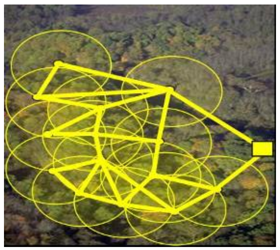
Feature	Chess	RoboCup
Environment	Static	Dynamic
World accessibility	Complete information	Incomplete information
Percepts	Symbolic	Non-symbolic
Execution	Turn-based	Real-time
Action effects	Deterministic	Stochastic
Agents	Central	Distributed

Why RoboCup Rescue?

- After a disaster many places are unreachable for humans
  - Robots can access places humans can't (e.g. small openings and confined spaces under the floor)
  - Robots can detect hazardous places and warn first responders
- Destroyed infrastructure: Problem of self-localization
  - Quality of disaster response strongly depends on information, such as maps with victim locations
  - Tom Haus (firemen at 9/11): "We need a tracking system that tells us where we are, where we have been, and where we have to go to"
- Efficient coordination of victim search, e.g. mixed initiative teams of humans and robots

The landmark of RoboCup Rescue:

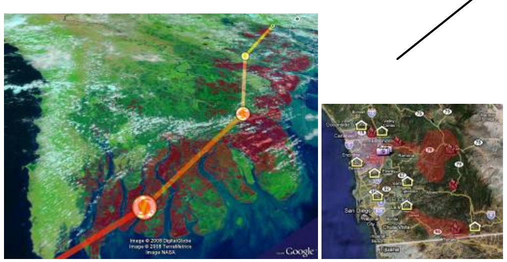
By the year 2050, enable large-scale MAS support for disaster mitigation



Sensor Networks
Integration of Sensors distributed in the city

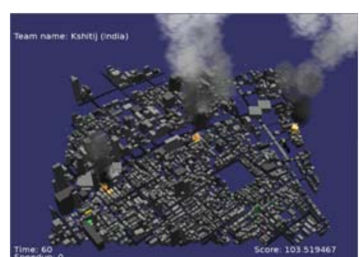
Emergency Response Center:
Efficient MAS decision making

Human rescue personnel
Digitally Empowered by wearable computers



Shared GIS Knowledgebase
e.g. GoogleMaps for sharing mission critical data

Simulator network
e.g. Fire Grid, RRSim



Robot Teams
Reconnaissance
Exploration of inaccessible places

The RoboCup Rescue Project

Rescue vs. Soccer

Feature	RoboCup Rescue	RoboCup Soccer
# Agents	1,000 or more (today just hundreds)	11 per team
Agent types	Heterogeneous	Homogenous
Environment	Unknown	Constructed
Real-time	Second/Minute	Millisecond
Hostility	Environment	Opponent
Decision effects	Long-term	Short-term

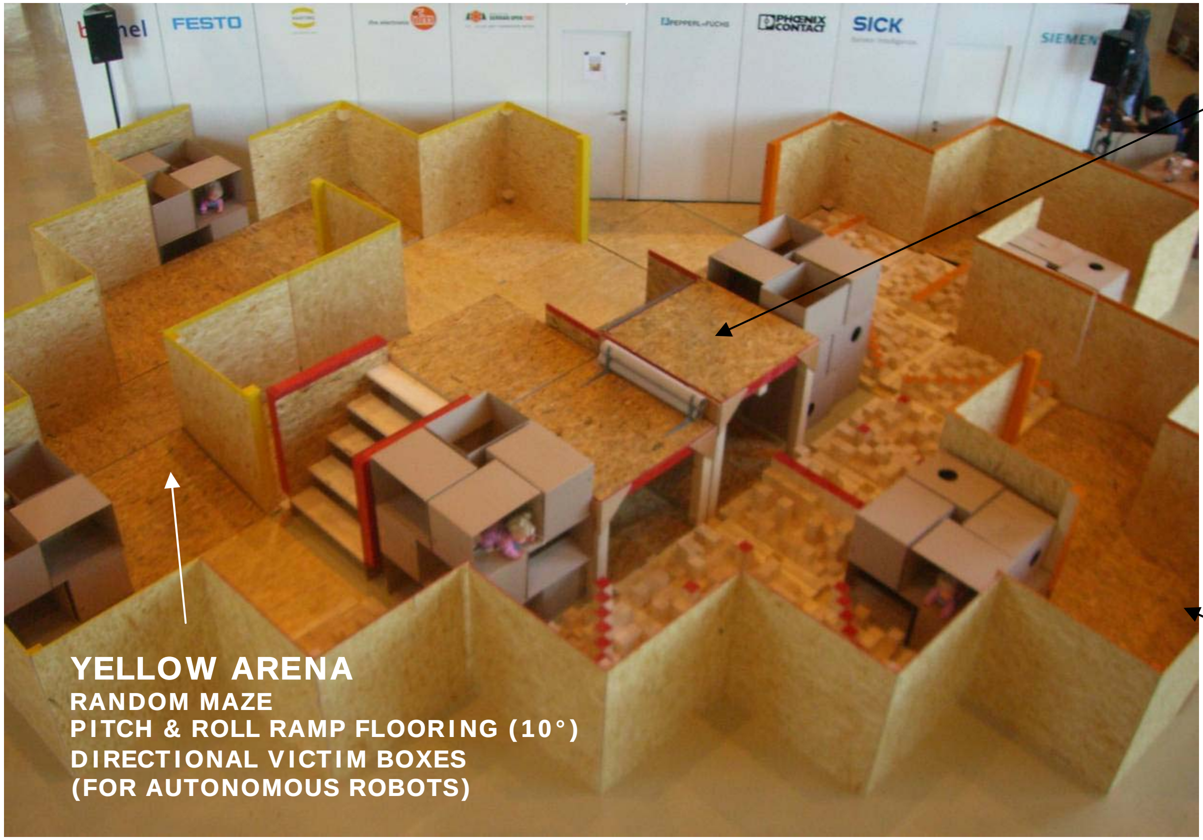
Rescue *Robot* Competition
Introduction



- Step-wise increase of difficulty (e.g. like golf courses)
- Building of standards for mapping and data exchange between heterogeneous units
- Towards “mixed-initiative” solutions, i.e. humans and robots build one team for efficient disaster response
- Cooperative development with simulation league

Rescue Robot Competition

Three types of arenas

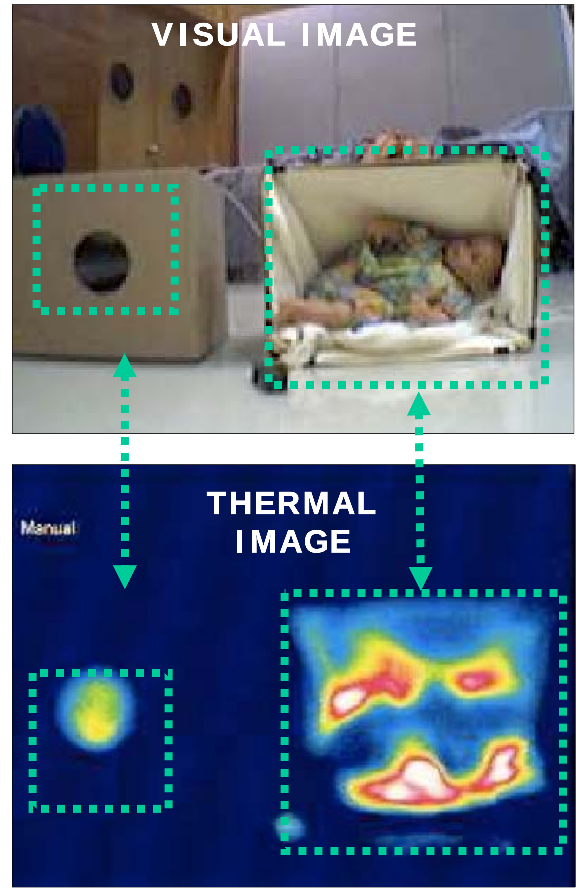


RED ARENA
FULL CUBIC STEPFIELDS
STAIRS (40°, 20CM RISERS)
RAMP (45° WITH CARPET)
PIPE STEPS (20CM)
DIRECTIONAL VICTIM BOXES

ORANGE ARENA
PITCH & ROLL RAMP
FLOORING (10°, 15°)
HALF CUBIC STEPFIELDS
CONFINED SPACES (UNDER
ELEVATED FLOORS)
VICTIM BOXES WITH HOLES

Rescue Robot Competition

Simulated victims

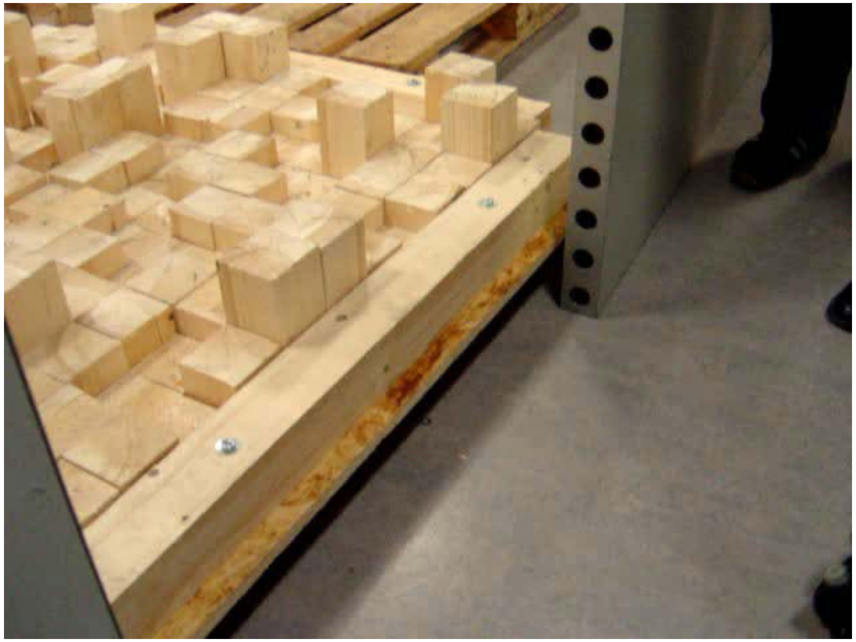


Signs of life: form, motion, heat, sound, CO_2

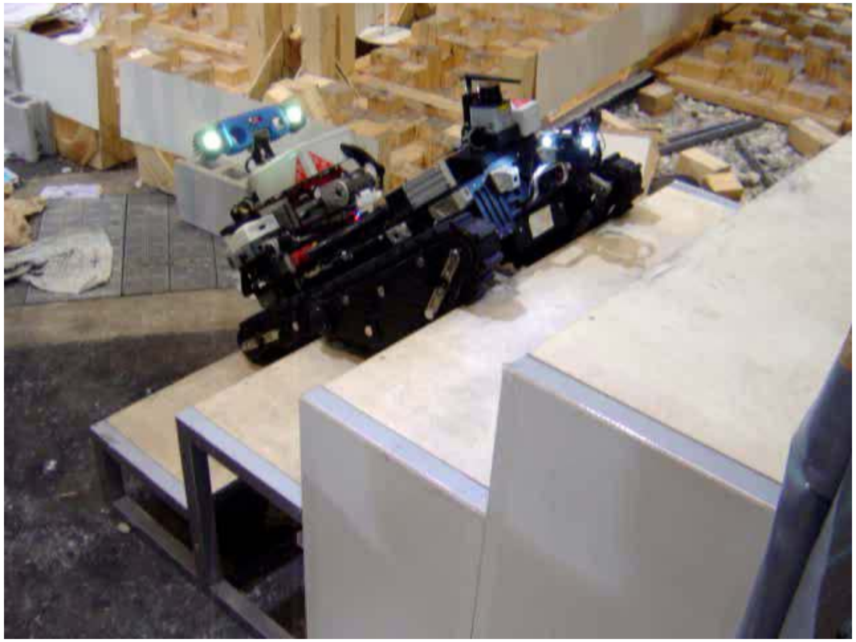
Heterogeneous teams at RoboCup Rescue

Rescue Robot League
Sometimes a hard job!

Your robot might be too small …

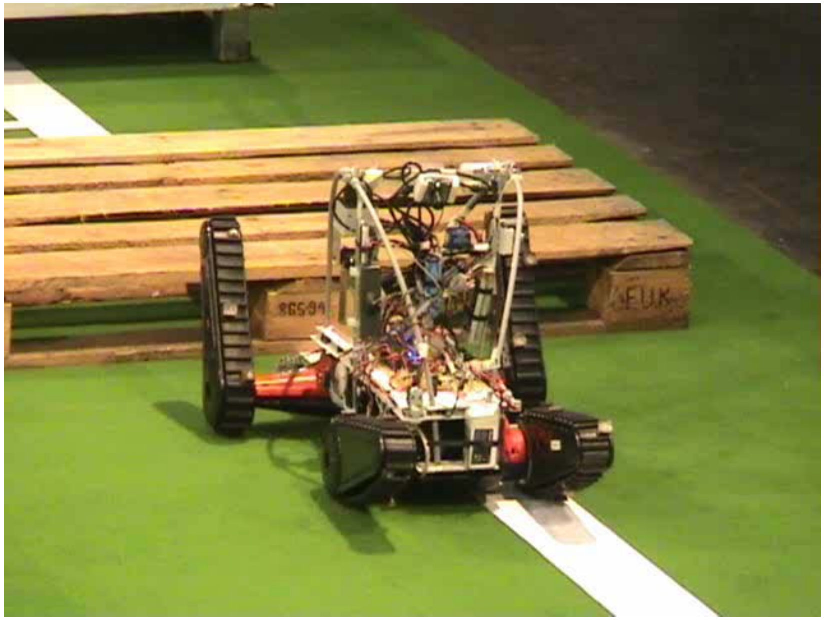


… or you robot might be too big



Results from RoboCup'06

Center Court Demo (Joint Work with AIICS Sweden)



Lurker robot overcomes autonomously 3D obstacles

Team cooperation between a Zerg robot and an UAV from Linköping University (Sweden)

Robot receives thermo images from UAV.

Results from RoboCup'06
Rescue Autonomy Competition

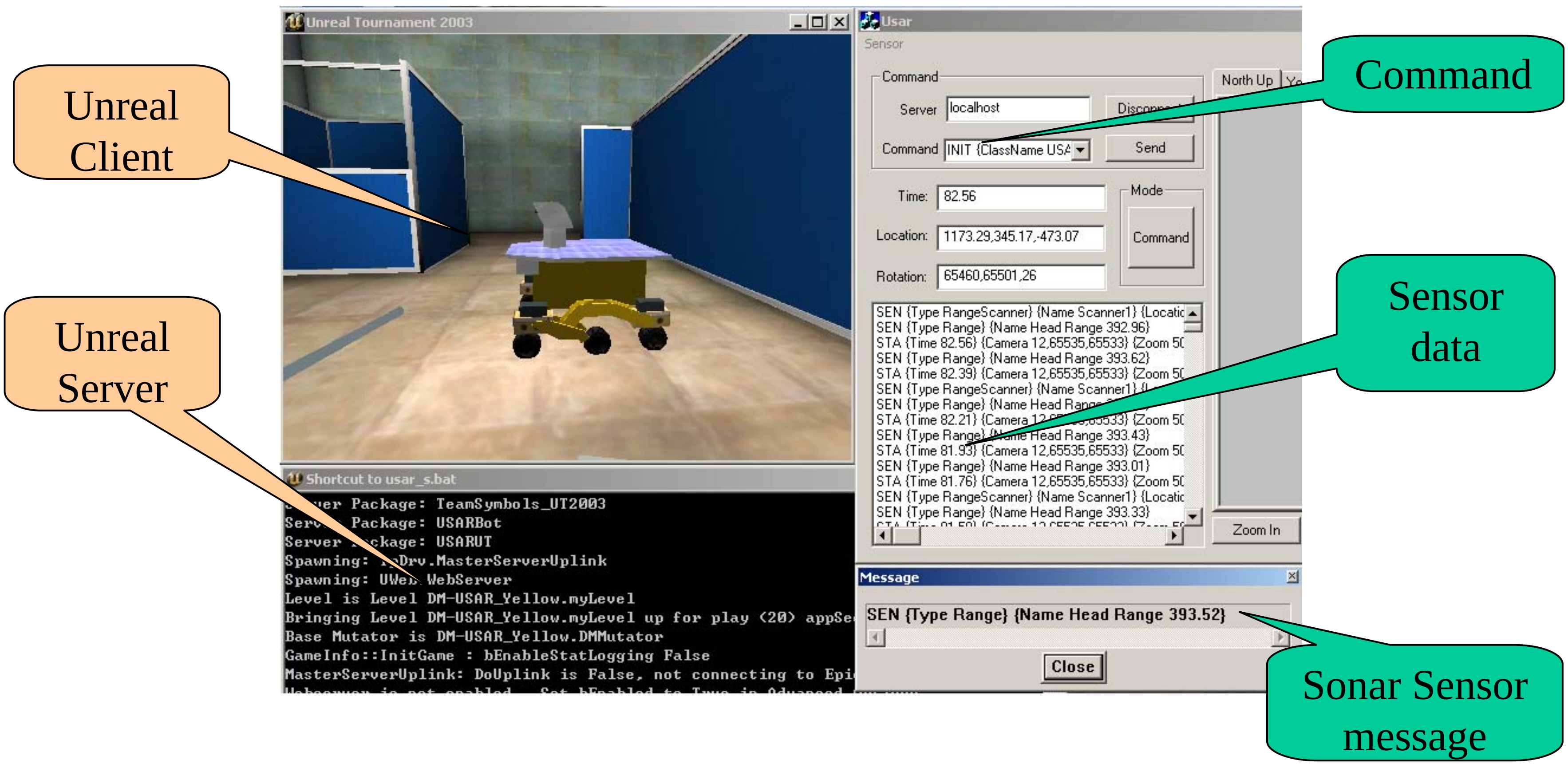
Autonomy final

Rescue *Virtual* Competition
USAR simulation based on game engine

- Based on the Unreal game engine (UT2004, Epic Games)
- Realistic models of USAR environments, robots (Pioneer2 DX, Sony AIBO), and sensors (Laser Range Finder, Color Camera, IMU, Wheel Odometry)
- Multiple heterogeneous agents can be placed in the simulation environment
- High fidelity simulation of up to 12 robots
- Agents connect via a TCP/IP interface
- NEW: Wireless-Communicatation simulation

Rescue Virtual Competition

Agent Interface



Rescue Virtual Competition
Physics and Mapping

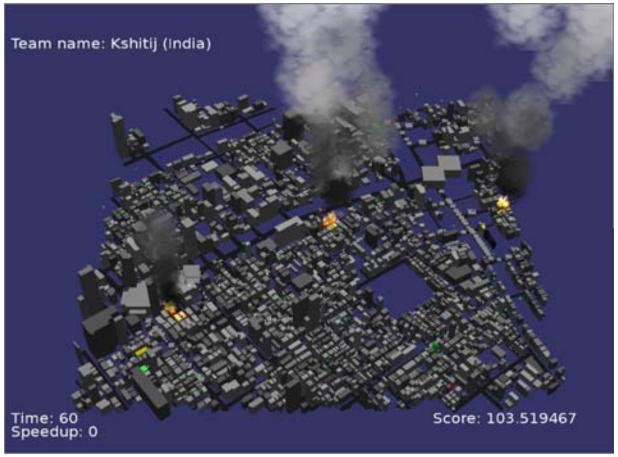
- Improved robot models for realistic mobility
- Robots can be customized

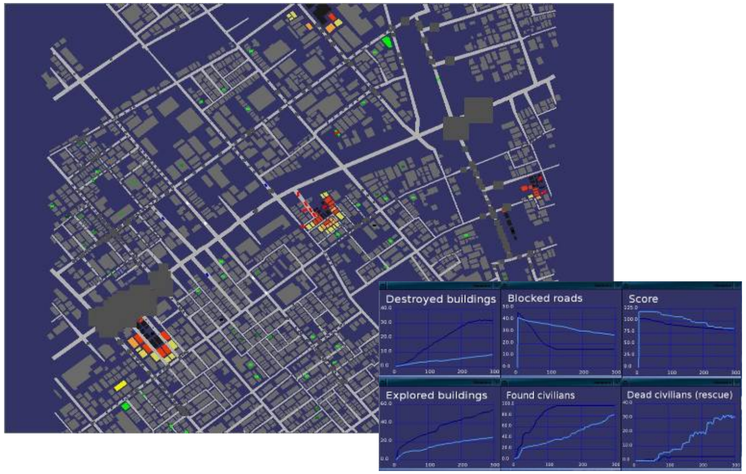
- Robots generate maps that have to be returned in GeoTIFF format
  - Maps will be overlaid on and compared to ground truth
  - Areas that have been “cleared” by the agents must be annotated (green color)

Rescue *Agent* Competition
Introduction

- Large scale disaster simulation
  - Simulators for earthquake, fire, civilians, and traffic
  - The task is to develop software agents with different roles, that
    - make roads passable (police)
    - extinguish the fires (fire brigades)
    - rescue all civilians (ambulances)
  - Difference to Soccer Simulation:

    A challenging MAS Problem since Agents ***must*** cooperate
  - Simulator components are developed within the "Infrastructure Competition"



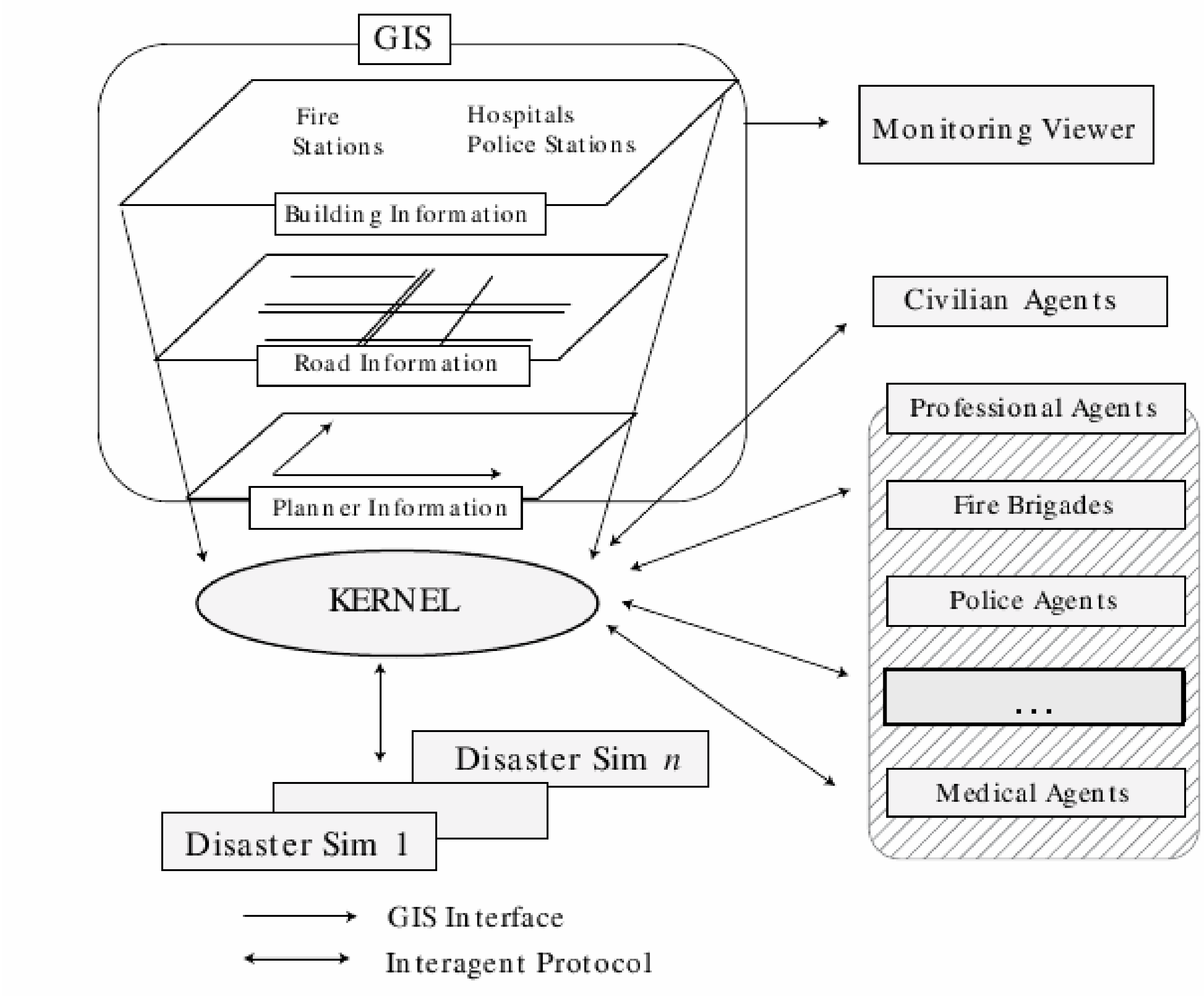


Rescue Agent Competition
Problem Classification

- The domain models a large, cooperative multi-agent problem (#Agents > 50)
- The environment is partially observable, agents have to act rationally given the history of their local percepts
- The domain is stochastic, effects of fire fighting and rescue might vary
- The environment is sequential, i.e. continuously progressing
- The domain is dynamic, e.g. fires and collapsing buildings
- The world is a simulation, therefore discrete
- Agents are heterogeneous since they have different capabilities
- The domain is decentralized due to a limited communication bandwidth

Rescue Agent Competition

Structure of Simulator and Agents



AI problems to solve by Rescue Agents

- *All Agents:*
  - Cooperative sensing and world modeling
  - Efficient victim search in the disaster area (team exploration)
  - Path planning with incomplete information (Canadian traveler problem)
- *Police Agents:*
  - Coordinated removal of road blockades (multi-agent path planning)
- *Fire Fighting Agents:*
  - Coordinated fire fighting and fire prevention (data clustering / coalition formation)
- *Ambulance Agents:*
  - Victim rescue (scheduling / sequence opt. problem)

Conclusion

- To learn about Multi-Agent systems from books is difficult
  - There exists no dominating strategy or algorithm (maybe in the future)
  - However, challenges within different domains are very similar
  - For learning about MAS you have to touch them!
- RoboCup Rescue offers a rich set of problems to MAS-AI
  - Lets solve them!

- Links:
  - Rescue Simulation League:
    - Homepage: http://www.robocuprescue.org
    - USARSim (code base): http://sourceforge.net/projects/usarsim
    - Rescue Agent (code base): http://sourceforge.net/projects/roborescue