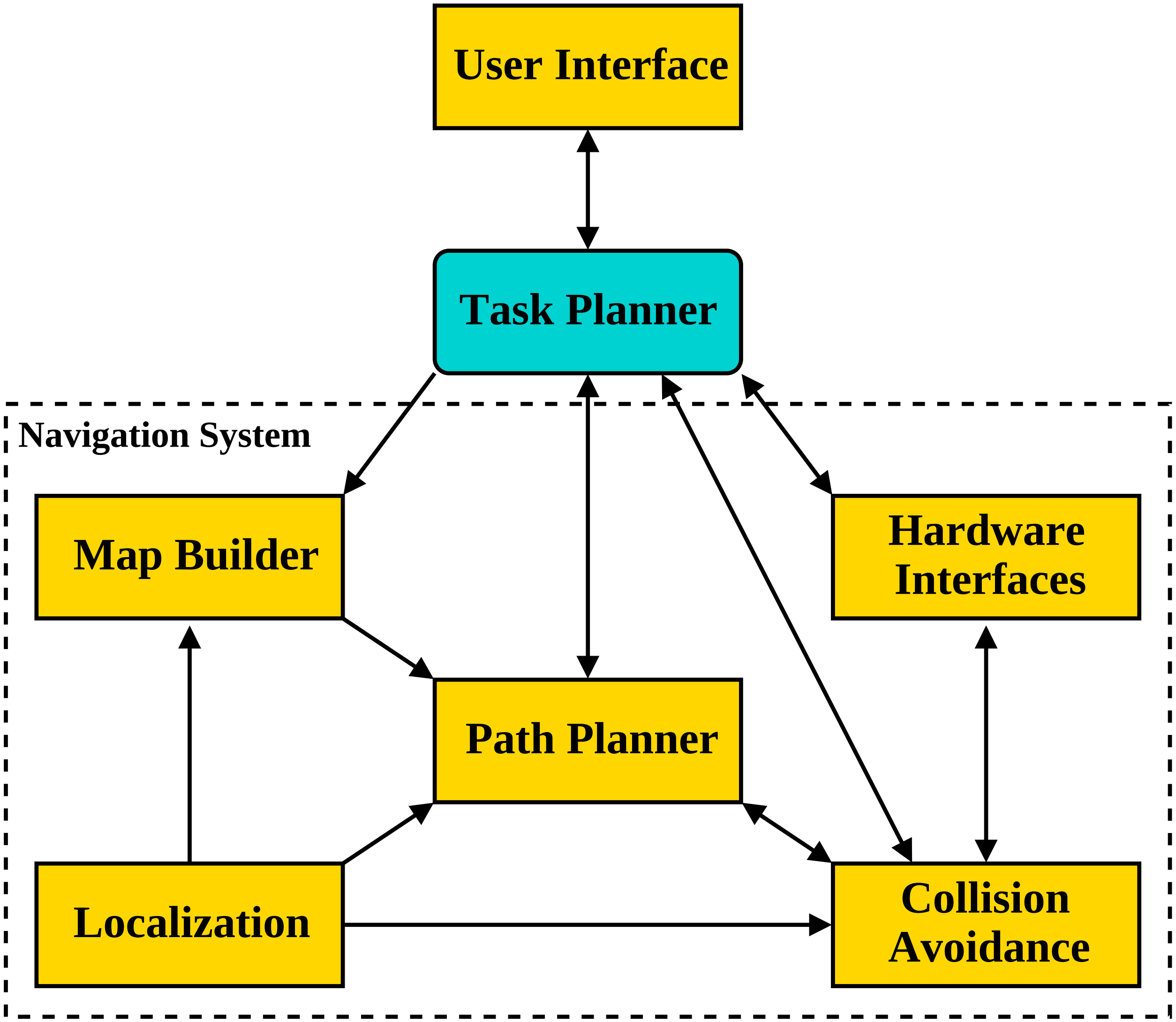
Issues in Multi-Robot Coordination

Gerhard Lakemeyer
Computer Science V
Aachen University of Technology
Germany

The Mobile Robot Carl

Carl's Control Software



Controlling Robots

There have been significant advances controlling single robots:

- CMU's XAVIER operating as a delivery robot
- The RHINO and Minerva Museum tour guides at the Deutsches Museum Bonn and the Smithsonian

Key features:

- Robust operation over several days
- Very good localization methods
- A lose integration of high-level (logic-based) and low-level control

Coffee Delivery in GOLOG

proc deliverCoffee

while $(\exists pers)$ **wantsCoffee(pers)** $\wedge$ $\neg$**hasCoffee(pers) do**

goto(coffeeM); pickupCoffee; goto(pers); giveCoffee(pers)

endWhile

endProc

proc goto(loc)

if robotLocation(rloc) $\wedge$ **rloc** $\neq$ **loc then drive(rloc, loc)**

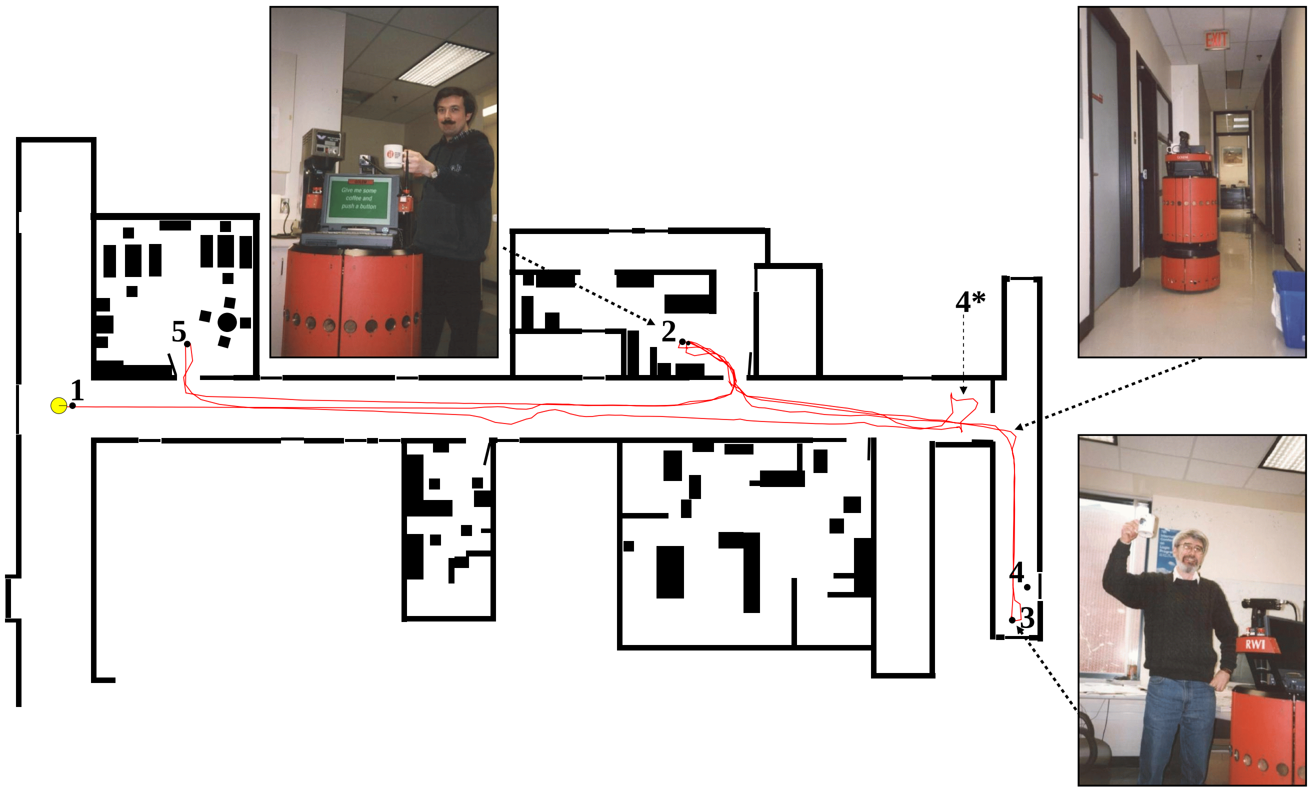
endProc

One of the possible solutions for **Do(deliverCoffee,**s_0**,**s'**)** is

s'=**do(. . . ,do(giveCoffee(ray),do(drive(coffeeM,ray),**

do(pickupCoffee,s0)))). . .)

Coffee Delivery in Action



The Multi-Robot Case

Things do not easily scale to many robots. **Why???**

Main Problem: The robo-centric view

- The robot assumes it is the only one acting deliberately.
- All objects outside the robot are considered obstacles, e.g., no difference between a chair and another robot.
- In general, neither high-level nor low-level control mechanisms take into account other agents.

A Scenario: Coordinated Delivery

There are n mobile robots in an office environment.

Incoming delivery tasks are to be serviced subject to conditions:

- Jobs can have different priorities and/or deadlines.
- New jobs can arrive at any time.
- People should be disturbed as little as possible.
- The environment is only partially known: doors/hallways may be blocked at times.
- Must deal with unforeseen circumstances in a flexible way:
  - a robot may be held up for too long;
  - complete failure of one or more robots.

Multi-Robot Delivery: Abstract Level

Not much different from typical multi-agent scenarios:

Make use of results in **distributed AI**, e.g.:

- Direct negotiation between robots
- Employ a central dispatcher
  - Use a (partial) global world model
  - Choose appropriate scheduling method

But there is more: **robots interact physically**.

Spatial Coordination

- **Conflict Prevention** (through **conventions**)
  - Drive on the right hand side of the hallway
  - Robots leaving a room have right of way over those entering
- **Conflict Detection**
  - Assess the situation by direct communication and/or querying a global world model
  - Active sensing
- **Conflict Resolution**
  - Relay intentions such as “I want to enter room A”
  - Choose strategy from a library
  - Planning

Language Req. for High-Level Control

The language in which to represent the application should include:

- concurrent actions (even for a single robot)
- actions with uncertain effects
- actions for sensing
- an agent concept
- a model of intention

CONGOLOG has been extended along some of these dimensions, yet not put to practice with those.

Many Things Can (and Will) Go Wrong

- Robots that want to meet are delayed.
- Robots break.
- Actions do not go as intended
  (even simple things like goto(x) fail).

A description of a plan can only serve as a **default**.

There is a need for **exception handling** during execution. One possibility is **plan transformation**,
e.g. RPL/XFRM (Bonn/Yale).
We plan to combine CONGOLOG with RPL/XFRM.

The Need for Integrated Solutions

High-Level Control

Actively monitor the environment

- identify agents in your neighborhood
- foresee problems (like expecting a robot in a room)
- adjust (low-level) navigation strategies to the current situation

Low-Level Control (e.g. navigation software)

- measure progress
  (e.g. heading towards destination)
- if there is a problem, get help from high-level control
  (e.g. if a robot gets lost, it might ask some other robot for directions)

Conclusions

- Controlling real robots is a rich playing field for agent technologies.
- It goes beyond what most work in distributed AI is concerned with.
- It needs an integrated and interdisciplinary approach to succeed.