

Een robot als een gedistribueerde  
applicatie

# Wat is een robot?

- A *robot* is an autonomous system, which exists in the physical world, can sense its environment and can act on it to achieve some goals (Maja J. Matarić, *The Robotics Primer*)
- *Autonoom*
- *Sensoren*
- *Actuatoren*
- *Doelen bereiken*

# Benaderingen robot control

- “Deliberative”
  - Vergaar alle gegevens, maak een plan, voer het uit
- “Reactive”
  - Een samenstel van reflexen
  - “Subsumption architecture”(Brooks)
- “Hybrid”
  - Combineer de goed kanten van beide

# MARIE



# Mobiele robot architectuur

- Diverse hardware en software
  - Sensoren
  - Actuatoren
  - Gelaagde, modulaire software
- Verschillende tijdschalen
  - Reactief – dus snel. B.v. voorkom botsingen
  - Midden – b.v. volg een pad
  - Lang – voer een complexe taak uit
- Robuust
  - Het uitvallen van een module mag niet resulteren in het uitvallen van de besturing

# Hybride architectuur

- Onderste laag: reflexief, gedragingen
  - Pad rijden
    - In kleine stukjes
  - Omgeving waarnemen
    - B.v. obstakels
  - Botsingen vermijden
    - Om obstakel heen
    - Of remmen en stoppen

# Hybride architectuur

- Hogere lagen:
  - Schakelen tussen gedragingen
  - Integratie van verkregen informatie
- Planning
  - Welke reeks gedragingen leidt (alsnog) tot het gewenste doel?
  - Hoe kom ik op een bepaalde plaats?  
(padplanning)

# Experiment

- De software en de hardware zorgen er samen voor dat de robot een bepaalde (samengestelde) taak kan uitvoeren.
- Omgekeerd bepaalt de uit te voeren taak dus ook welke SW en HW componenten nodig zijn en hoe ze moeten samenwerken.
- De taak/het experiment zijn leidend voor het ontwerp!



# Experiment

- De robot moet in een vrij volle ruimte een pad langs een wand rijden.
- Botsingen vermijden.
- Zonodig een wand opzoeken.
- Uit doodlopende stukken ontsnappen.
- Bijhouden waar hij is en welk pad hij heeft gevolgd.
- De omgeving in kaart brengen
- ...

# Werkplan

- Bouw je software in testbare stappen op, component voor component.
- Specificeer steeds eerst een (deel-) experiment dat je met je nieuwe component moet kunnen doen.
- Soms moet je eerst nog verkennen:  
b.v. wat ziet mijn ultrasoon sensor eigenlijk als ik rondrijdt?

# Testen

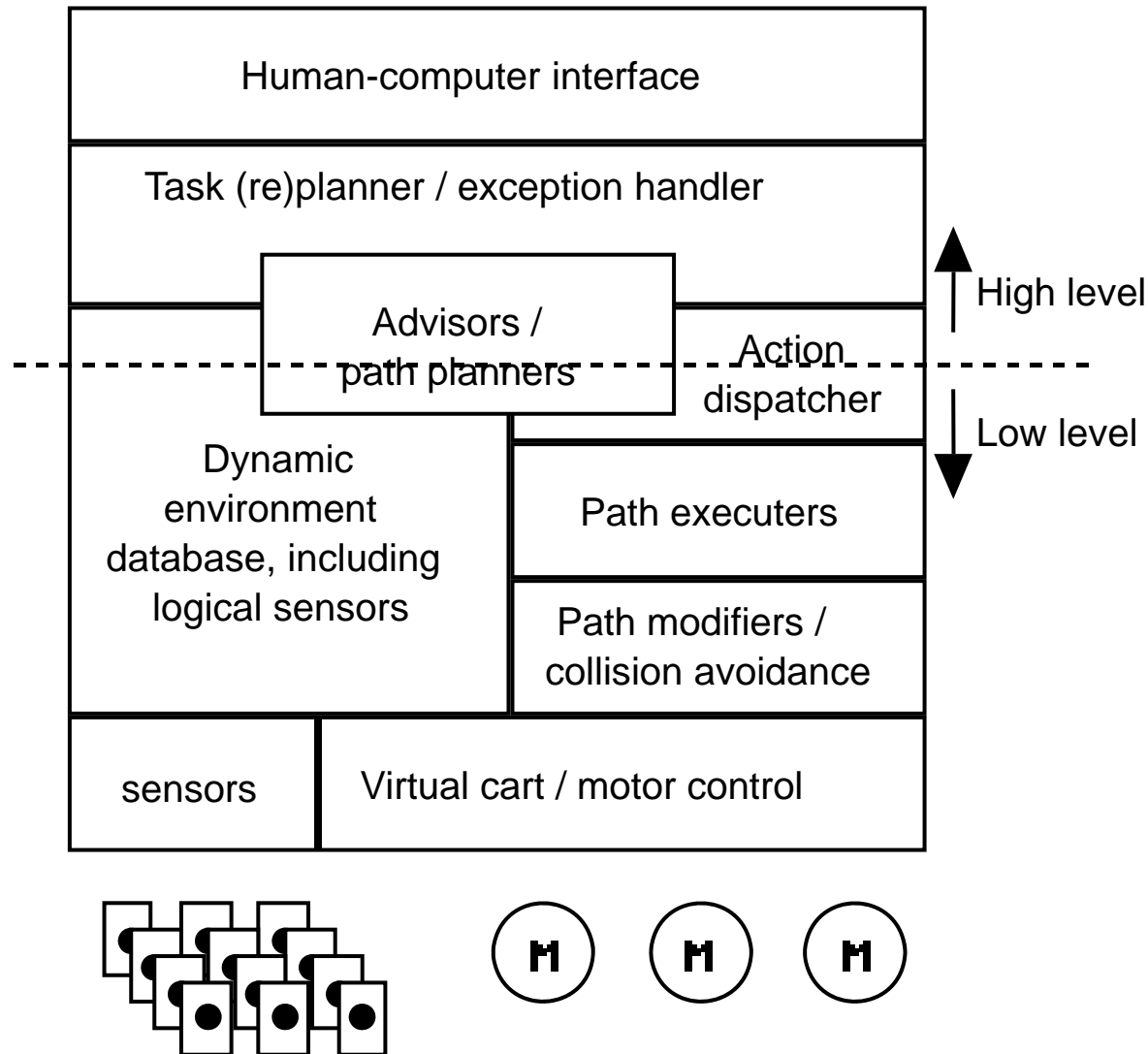
- Test niet alleen of een module correct werkt in een “correcte” omgeving.
- Test ook het faalgedrag:
  - Blijft het systeem veilig?
  - Wordt de juiste foutmelding doorgegeven?
- B.v. muurvolger zonder muur.

# Integratie

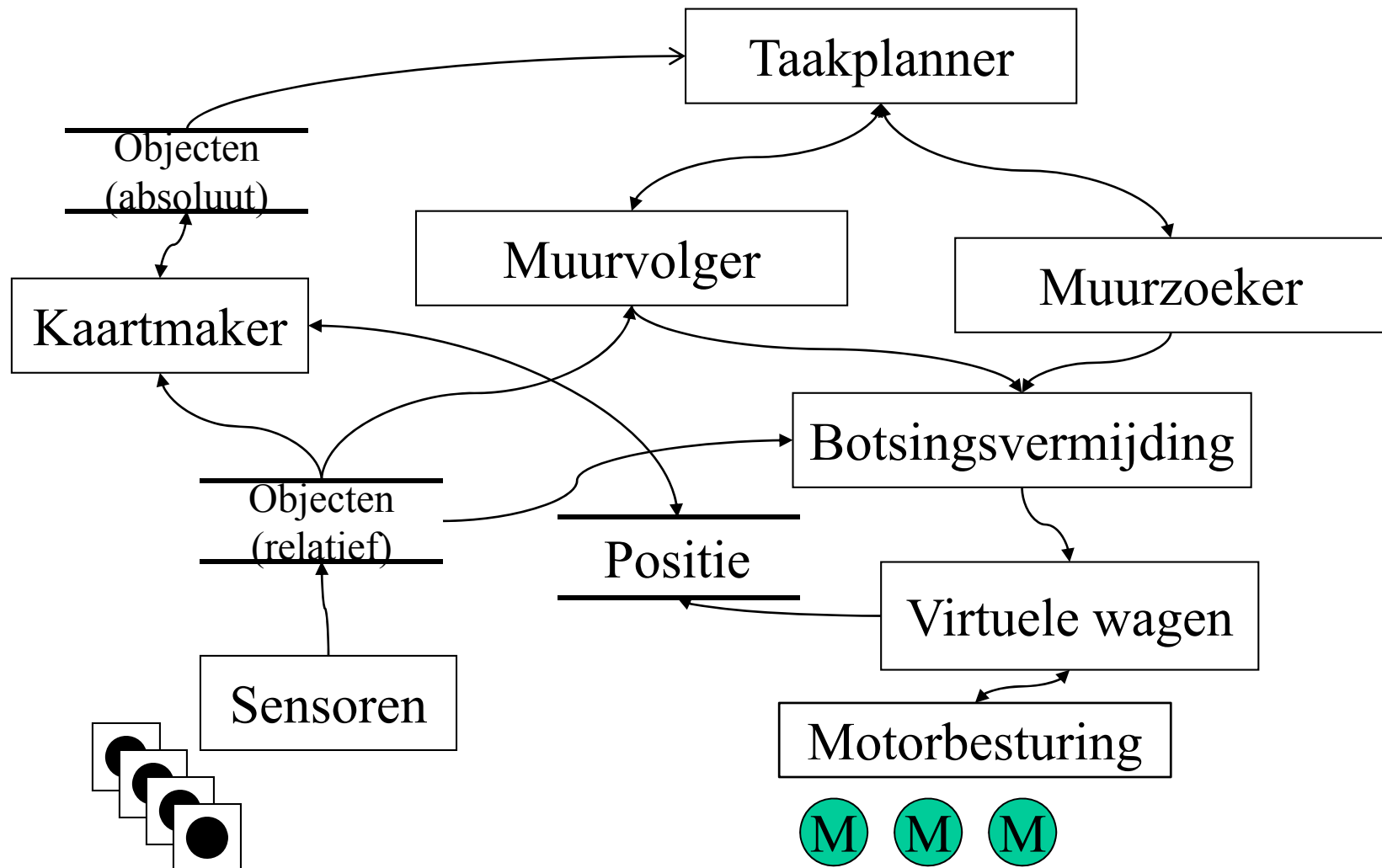
- De interactie tussen modules kan in onverwacht gedrag resulteren.
- B.v. muurvolger + botsingsvermijding
- Ook weer: vooraf experiment specificeren, uitvoeren en documenteren.



# MARIE Architectuur (modules)



# Mogelijke architectuur



# De virtuele wagen

- Abstraheert en bestuurt de echte wagen
- Berekent feitelijk gereden pad, positie, oriëntatie, snelheid, onzekerheid
- Vergelijkt met gewenste pad
- Garandeert fail-safe gedrag
- Kan op elk moment nieuwe instructies ontvangen
- Stuurt motoren aan



# Sensormodule

- Leest sensoren uit
- Meldt evt. falen van een sensor
- Vertaalt b.v. sensordata naar geschatte relatieve posities van objecten

# Botsingsvermijding

- Vergelijkt opgegeven pad met posities van objecten in de directe omgeving.
- Past snelheid, padkromming, padlengte zo nodig aan
- Vindt toch een botsing plaats, voorkom dan dat de robot doorzet.

# Ondersteunende modules

- Denk b.v. aan
  - Non-blocking communicatie
  - Real-time data opslag
  - Monitoring
  - Communicatie tussen modules en simulator

# Te gebruiken platform

- USARSIM
- Binnen de robotsimulatiewereld een standaard
- Gebaseerd op “Unreal Tournament” game engine
- Is o.a. geïnstalleerd in G0.10-G0.12 (windows)
- Handleiding staat on-line

# Kaart maken

- *Simultaneous Localization and Mapping* (SLAM)
- Odometrie heeft een beperkte nauwkeurigheid
  - Eigen positie wordt gaandeweg minder precies
- Waargenomen objecten kunnen worden gebruikt om relatieve beweging nauwkeuriger te bepalen.
  - Maak een kaart waarin je later objecten opnieuw kan terugvinden en zo je eigen positie kan herijken

# De te gebruiken robot

- P2DX
  - 2 aangedreven wielen en een zwenkwiel

