Brooks' Subsumption Architecture

Jacky Baltes

University of Auckland

Email: j.baltes@auckland.ac.nz

Web: www.tcs.auckland.ac.nz/~jacky

March 7, 2003

Brooks' Subsumption Architecture

- One of many possible frameworks for the design of the control of an intelligent agent.
-

Requirements

- Act appropriately in a real time, dynamic environment
- Robustness with respect to failures
  (Hardware/Software). Craceful degradation.
  - Sensors
  - Actuators
  - Control systems
- Extendability: add new sensors, actuators, tasks.
  Incremental approach.
- Generation of goals/subgoals
- Arbitrate/combine multiple goals
- Balance between strategic and reactive systems

Abstraction: The Crux of AI

- AI researchers: Don't get credit for their work
- Ever heard of an AI failure?
- Using abstraction, a problem is composed into different levels.
  - The ones that are handled by the AI system (AI component)
  - Non-AI Component. Factor out sensors and actuators. Simulation.
- Blocks world: Hundreds of papers and work. Good success, but doesn't scale up to the real world.
- Representation: Propositional, first order logic.How do you represent a car?
- Wishful semantics: (on blockA table) $=$ (on xyz yyx)
- Claim 1: Must build complete systems.Incremental approach
- Claim 2: Must act in the real world

Subsumption Architecture

- Time scale of evolution: 3.5 billion cells, 2.5 million first humans, 5000 years ago writing.
- **Functional** decomposition:
  - Planning, execution, navigation, reasoning, vision
  - Interfaces between the different layers?
- Independent **Activities** (Complete execution from the sensors to the actuators)
- Activities are goal oriented
- Different layers of behaviors
- Higher layers can overide lower layers (Side-taps of connections in the lower levels)
  - Suppression: Side tap at the input. Route message to higher layers and suppress sensor for a certain time period. Turn off sensors
  - Inhibition: Side tap at the output side. Prevent messages from being sent on this wire for a certain time.

- Activity at a higher layer is a subset of a lower layer
- Each activity consists of finite state machines, registers and computational units. Slow asynchronous communication channel.

Mobile robot example

- Layer 0: Avoid obstacles
  - Sonar: generate soner scan
  - Collide: send HALT message to `forward` if about to run into an object
  - Feel force: Compute overall repulsive force. This vector is passed to `run-away`, which sends it to the `turn` cell.
- Layer 1: Wander behavior
  - `wander` generates a random heading
  - `avoid` reads repulsive force and new heading and generates new heading and feeds it to `turn` and `forward`.
- Layer 2: Exploration behavior
  - `whenlook` notices idel time and looks for an interesting place. Inhibits wandering
  - `path-plan` sends new direction to `avoid`.
  - `integrate` monitors path and sneds them to the path plan.

Disadvantages of the Subsumption Architecture

- No internal representation. Reasoning, Inference
- No strategic planning
- Learning
- Switching between behaviors (Hormon Theory)
- Too reactive
- How to create **new** goals?
- Open Questions:
  - Maximum complexity of layers before the interaction is too complex.
  - Maximum number of layers
  - Maximum reasoning capability of the whole system