Software Platform Concepts and Cognitive Robotics

David Vernon

(with contributions from Bernhard Sendhoff, CTO Honda Research Institute Europe GmbH)

iCub examples courtesy of the RobotCub Project
www.iCub.org
Software Platform Concepts

Technological Issues

• Cognitive systems are complex software systems

• Integration
  • Many researchers
  • Many years
  • Teamwork: both academic & industrial

• Must be
  • Modular
  • Interoperable
  • Industrial-grade software engineering
    (e.g. build, test, documentation, ..)
Software Platform Concepts

Technological Issues

Robotics middleware
- Hot topic
- Major players taking an interest

• Way forward
  - Open source or
  - “Hierarchically” proprietary software projects
Created to support Community research on embodied cognition

Goal: research platform of choice

- Exploit it quickly and easily
- Collaborate & Share results
- Benefit from the work of other users
iCub
iCub
Software Architecture

Multiple YARP processes
Running on multiple processors

Gbit Ethernet

Level 0 APIs: data acquisition & motor control

Level 1 APIs: perception/action behaviors
Innate perception/action primitives
loose federation of behaviors

Level 2 APIs: Prospective Action Behaviors
Coordinated operation: Ontogenic Development

Based on phylogenetic configuration

Cognitive Architecture

Software Architecture

iCub Embedded Systems

HUB pc104

DSP

Sensors & Actuators
What is YARP?

• An open-source software library for humanoid robotics; born on Kismet, grew on COG
  – University of Genoa / MIT collaboration

• Designed to support and encourage:
  – Collaboration (code-sharing across space)
  – Longevity (code-sharing across time)
Software Platform Concepts

Technological Issues

Modularity and interoperability ⇒ Incremental development & configuration

• switch between different hardware platforms
e.g. robot and automobile

• combine different functional modules
e.g. different sensors and actuators

• processing needs
e.g. speed can be adjusted for different hardware requirements

RE-USE!
Software Platform Concepts

Scientific Issues

• Research projects
  • Well-defined, functionally-oriented problems
    e.g. face recognition, dialogue system

• System-level properties receive much less attention
  • Why? Need an operational base system as a starting point

• BUT systems properties are extremely important
  • robustness, graceful degradation, e.g. safety critical applications (cars)
  • efficiency – increase operation time for mobile platforms
  • security & safety (e.g. a 60 kg humanoid robot being hacked)
  • learning – True learning is a system-wide property
    • control of learning & the learning path
      e.g. organization of short-term to long-term memory
Cognitive Robotics

*Developmental Embodiment*

- Embodiment & situatedness: well-established

- BUT, from an evolutionary point of view, morphology and processing structure are much stronger coupled.

- Genetic encoding of structure of brain and body: same mechanisms

- Embodiment and intelligence grow and develop *together*
Cognitive Robotics

*Developmental Embodiment*

- Robots will be one element of a personalized information infrastructure
- The future: NOT a ‘brain in a machine
- Machine will be part of a personalized information infrastructure
  - (ad hoc) information networks
  - Mobile devices
  - Ubiquitous sensors
  - …
Automotive Intelligence

- Robustness, robustness, robustness .. must be system inherent
- Learning must be restricted ("fail-safe")
- Single functionalities must be embedded
- The car will be part of an urban information infrastructure
  - intelligence will be distributed and available on demand
- The automobile must remain a personal space
  - that is part of its fascination
  - thus any cognitive aspect must be highly personalized

(cf. Christoph Eberst)
Cognitive Systems

The Roadmap

• Build simple systems
  • never(!) switched off
  • Incrementally change morphological and processing structure

• Build systems that can live – even a simple – life exploiting both a physical and a virtual presence

• Research system level properties (robustness/safety) in their own right
  • Not after the system functionality has been developed

• Regard intelligence and cognition as strategies to cope with limitations

• Build toys as simple cognitive systems