Away Day
June 2014
Robot Control, Intelligent Pervasive Environments and Cognitive Architectures

Robot Group:
  Ping Jiang
  Chandrasekhar Kambhampati
  Darryl Davis
  Yongqiang Cheng
  Peter Robinson
  Rob Miles
DND: Self Organising Cognitive Architectures

- Cognitive Architectures for Motivation, Affect and Learning
  - Large Single Agent Architecture (1998 onwards)
  - Mapping CAMAL $\Rightarrow$ SCARABs

- Situated Cognitive Architecture for Reasoning About Behaviour
  - Reduced Task-focused Architecture on board Robots (pi+)

- SOCA = CAMAL + SCARAB (x6) + ACA
  - ACA (Abstract Cognitive Architecture) on Laptop/P3DX
  - CAMAL sits on Pioneer P3DX + Adlink Embedded
  - SCARAB sits on AmigoBot+Pi+Camera or Nexus+Pi+Cam
SOCA: Why?

- Research area and application issues
  - How to do distributed cognition
    - Pervasive Intelligence and Adaptive Control
    - Adaptive Distributed (mobile) Processing
  - How to build domain and task models
    - Perception, Tasks and Spatial Reasoning
    - Big issue in industrial cognitive robots ⇒ Opportunity
  - How to generate BDI models
    - Instantiating a tested and reliable reasoning model
  - Architectures for distributed cognition
    - Ongoing (20 years) research into sophisticated AI architectures
      - Also feeds into other Intelligent Systems areas
    - Steady stream of journal outputs
Some simple experiments with simple robots
- I need to learn robotics! – mostly simulation previously
- Enabled first implementation of ACA

Swarm Intelligence - Dancing Robots!
- Flocking is easy to do in software simulations
- Challenge in doing it with simple robots
- Nine ActivityBots (I have one to repair)
  - All with variety of single or multiple Sonar and/or IR Sensors
  - 3 with Raspberry Pi B plus NOIR camera
  - Each has same set of multiple Behaviours (written in C)
  - Local Control Architecture to switch Behaviours
  - Laptop High-Level Control of swarm behaviour configuration
    - Simple Task Reasoning Model (1st version of ACA in Prolog)
    - Peer-to-Peer Wifi or closely coupled interfaces to Robot
CAMAL Architecture (single agent)

D.N. Davis, Cognitive Architectures for Affect and Motivation
Cognitive Computation 2(3), September 2010.
Goals and Behaviours

- Goals – ongoing, interrupt-able, periodic, etc
  - Maintenance Goals
    - State to be maintained (Positive or Negative $G_x$ or $\neg G_y$)
  - Achievement Goals
    - State to be achieved (Positive or Negative $G_r$ or $\neg G_s$)
  - Qualitative as described in most AI literature
    - Typically achieved through plans or associations or intentions
  - Quantitative as described in mathematical control theory
    - Typically achieved through lower-level (reactive) behaviours
  - Overall tasks achieved by sequences of sub-goals
    - Idea of Multiple Levels of Reasoning
      - Use of Goal Specific Plan Sets (as BDI)
    - Idea: SOCA as a set of distributed hierarchical planners??
Belief Desire Intention Models

- Consider “Sense – Think – Act” Cycle
  - Set of Propositions
  - P : Set of Perceptions
  - B : Set of (Current) Beliefs
  - D : Set of Desires (or Goals)
  - I : Set of Intentions (defining Behaviours or Actions)

- BDI Updating rules
  - Use Domain Model for Task under consideration
  - Affordances associated with Beliefs, Desires, Intentions
  - Extend with probabilities derived from experience

- Formal Specification and Proof
  - To Complete and Publish
Consider “Sense – Think – Act” Cycle

- $P := \{ r, s, q, p \}$
- $B := \{ \neg p \}$
- $P \otimes B \rightarrow B'$ (Belief Revision - Various models)
  - Preference := $\{ p \rightarrow \neg p, s \rightarrow r \rightarrow \neg p, q \rightarrow \neg p \rightarrow \neg s \rightarrow \neg r \}$
- $B' := \{ p, s, r, q, \neg p \}$ //beliefs etc given Affordance $\mathbb{R}[0,1]$
- $D := \{ p \& s \rightarrow gz, r \& \neg p \rightarrow g \neg z, s \& q \rightarrow gx \}$
- $B \otimes D \rightarrow D'$ (Desire Update)
- $D' := \{ p \& s \rightarrow gz, s \& q \rightarrow gx, r \& \neg p \rightarrow g \neg z \}$
- $I := \{ r \& gz \rightarrow b1, gz \rightarrow b2, p \& gx \rightarrow b3, ..., g \neg z \rightarrow b5 \}$
- $D' \otimes I \rightarrow I'$ (Intention Mapping)
- $I' := \{ p \& gx \rightarrow b3, r \& gz \rightarrow b1, s \& gz \rightarrow b2, \neg p \& g \neg z \rightarrow b5 \}$

Davis, D.N., Linking perception and action through motivation and affect
Distributed Cognitive Robots (SOCA)

- Intended Computational Nodes (plus laptop)
  - Robot WiFi Network Server PC
    - Peer to Peer and Peer to Network (or Subgroup)
    - Also Framegrabber for Environmental Camera
  - Pioneer P3DX plus Camera ⊕ Adlink Matrix ⊕ Kinect
  - AmigoBot1 (x4) (Red, Black, Green, RedII)
    - AmigoBot ⊕ Raspberry Pi B+ ⊕ Wifi ⊕ USB Webcam
  - AmigoBot2 (Red)
    - AmigoBot ⊕ Raspberry Pi B+ ⊕ WiFi ⊕ USB OmniDirectional
  - Nexus 2WD (Yellow)
    - Nexus ⊕ Arduino ⊕ Raspberry Pi B+ ⊕ Wifi ⊕ USB Webcam
  - Robot Environment (Ambient or Pervasive Intelligence)
    - Independent Intelligent Sensor
      - RFID Sensors, Environmental Cameras etc.
Initial Development of the Robots

• **Short Video of first run**
Multi Layer BDI with Localised Reactive Physical Environment

TCP Wifi Or Wired

Action and Behaviour Feedback

ACA or CAMAL

SCARAB\textsubscript{i}

Physical Environment

Sensor Mapping

Rights Reserved
Similar to HTN Task Decomposition

Used as recursive model for decomposition of problem solving behaviours

Increasing priority of preconditions
Domain and Task Models

- Distributed Cognitive Model
- Host Laptop running abstract architecture
  - Reasoning about Overall Task
  - Requests Specific Architectures to Perform tasks
    - Dynamic Reconfiguration across nodes via negotiation
  - Feedback in form of Perceptions and Task Responses
- Robot running Situated Architecture
  - Communicates with Host Architecture
  - Reasoning about Specific Tasks and Behaviours
  - Configures Robot microcontroller – Behaviour Selection
  - Manages Sensory Responses
    - Sonar maps and Blob Detection locally
    - Farms out intensive Vision processing to more capable nodes
Motivators - An instantiation of BDI combination
- Intensity = Degree of Belief, Goal Importance, Association Insistence (investigation of metric combinations)

Means of Organising Plans and Goals
- by designer
- by runtime agent for plan & goal selection
- by runtime agent for learning
- Typically a cyclic graph
  - Behaviours fail and things change due to other agents!

Nodes: Set of Possible World and Agent States
- Initial, Intermediary and Goal States

Arcs: Actions, Behaviours and Plans
- Link nodes separated by single action
Recurrent BDI in State Space

Space of Initial States

State\textsubscript{Initial}

BDI\textsubscript{1} \quad S1''

BDI\textsubscript{6} \quad S2

BDI\textsubscript{5} \quad S3''

BDI\textsubscript{2} \quad S3

BDI\textsubscript{3} \quad S1

BDI\textsubscript{9} \quad S1'

BDI\textsubscript{4} \quad S3'

BDI\textsubscript{7} \quad S1'

BDI\textsubscript{11} \quad S3'

G\textsubscript{1} \quad G\textsubscript{2} \quad G\textsubscript{3} \quad \ldots \quad G\textsubscript{m}

Space of Representable States

Space of Goal States
Example SCARAB BDI Graph at time t1
So why the BDI for SCARAB?

- Non-linear, dynamic nature of environment
  - Non-linear, dynamic nature of reasoning
- Adaptive reasoning model associated with BDI
  - Motivator history allows tuning of behaviours that fail
  - BDI Metrics allow for reinforcement learning

At Time t1
- 14 Beliefs, 35 Goals, 39 Intentions,
- 8 possible BDI combinations, 56 possible BDI Chains

At Time t4
- 40 Beliefs, 35 Goals, 39 Intentions,
- 21 possible BDI combinations, 83 possible BDI Chains
Summary

• SCARAB and ACA being developed
  – Prolog ⊕ C++ ⊕ Aria ⊕ MobileSim
  – Behaviours then mapped onto real robots
    • Typically parameter tuning for real robots (but Issues!)
    • OpenCV for Vision – Just Blob detection on the Pi’s
  – Design/Implementation Experiments ⇒ Journal Paper
  – ACA now being developed for SCARAB
    • SOCA Design/Implementation Experiments ⇒ Journal Paper

• CAMAL to be redeveloped
  – Linking in with SCARAB and ACA
  – Perception and Spatial Reasoning once up and running

• H2020 Funding opportunities for this and group
Coda: Representing Time and Change

- Whose time framework?
  - Maintaining consistency across multiple frameworks

- Discrete Time (e.g. in symbolic reasoning)
  - move from instant to instant with gaps
    - granularity of time instances
  - move from interval to interval
    - what about time between intervals

- Continuous Time (e.g. in real-time i.e. Robot)
  - Time as continuous function - No gaps in time
  - useful for continuous variables
    - e.g. orientation, motor activity etc.
Event-based Time
- time steps not necessarily uniform
- intervals are periods between interesting events
  - time agent entered GameLevel
  - time agent picks up amulet
  - time an agent uses amulet

State Space
- non-explicit representation of time
- actions as mapping between states
  - state1 object on floor next to agent
  - state2 amulet held by agent
  - state3 amulet used to enter portal
- Represent time as points
  - events occur by specific time point
  - propositions T or F at a time point

- Naively - state indeterminate in between points
Representing Time

- Represent time as intervals
  - events occur over an interval
  - propositions T or F over the interval
  - are the end points part of the interval?

```
p, q, ¬r
   t1
       p, ¬q, r
```

```
p, q, r
   t2
       p, q, r
```
Mapping between point and intervals

- Point-wise properties
  - T of interval when T of every point in interval
  - P is true of interval $t_0$ through to $t_2$
  - Q is true of interval $t_1$ through to $t_2$
  - R is true of interval $t_0$ through to $t_2$
Gestalt properties

- T of interval but may not be of any sub-interval

- P is true of interval covering $t_0$ to $t_2$
- But NOT of sub-interval $t_0$ to $t_1$
Reasoning About Events That Take Time To Achieve

- Many agents in environments where operators do not invoke immediate change in state space
- Agent invokes plan with two operators
  - operator1 - recharge batteries - two cycles plus
  - operator2 - sense environment - one cycle
  - Variation: do operator2 until operator1 complete

\[
\text{recharge}(s_i) \quad \text{sense}(s_{i+1})
\]
Goals and Rules allow Goal interruption and completion
- Goal potentially lost if using Scripted rules
Full Recharge takes two cycles
Cannot recharge if hiding from sensed enemy