Towards a Language for Understanding Architectural Choices in Robotics

Robin Passama, OBASCO Research Group (Ecole des Mines de Nantes, LINA, INRIA)

David Andreu, LIRMM Robotics Department (Université Montpellier 2, LIRMM, CNRS)
Introduction

➢ Research Field: Robot Control Architecture Design

➢ Issue addressed: How to describe, communicate & compare architectural solutions?

➢ Proposal: A conceptual model as a basis for a domain specific modelling language
Software Engineering Challenge, Limitation and Direction

➢ Challenge: expressing and integrating various human expertises into the software architecture
  • control, navigation, vision, AI, etc.
  • medicine, space missions, etc.

➢ Limitation: software engineering paradigms centered on programmers not on domain experts.

➢ Direction: Domain specific languages(DSL) provide domain experts dedicated terminologies and abstractions.
  ✔ A DSL for robot architects
Issues for Control Architectures Description

➢ Variety of control architecture design methodologies
  • Behavioral
  • Reactive
  • Deliberative
  • Hybrid

✔ Concepts usable with all methodologies

➢ Different levels of technical details
  • Framework
  • Hardware aspects
  • Programming language/paradigm

✔ Concepts disconnected from implementation details
A Conceptual Model for Control Architectures

➢ Goal: defining concepts, terminology and graphical notations

➢ Focus: Decomposition of decisional process within control architecture
  • Abstract layered organization
  • Explicit relation between decisional process decomposition and controlled systems
  • Interactions as the core of decisional process decomposition
  • Qualifying domain knowledge used in decisional process

➢ Constraints:
  • Considering no implementation detail
  • Allowing all control design methodologies
Overview of Main Concepts
Knowledge

A Knowledge entity identifies a structured piece of information about the world within which the robot controller evolves.

**Categorization**, based on stable aspects [Brugali and Salvaneschi, 2006]

**Embodiment**: refers to the consciousness of having a body that allows the robot to experience the world directly.
*Types*: Body, Infrastructure

**Situatedness**: refers to evolving in a complex, dynamic and unstructured environment that strongly affects the robot behavior.
*Types*: Environment, Interaction, Phenomenon

**Intelligence**: refers to the ability of the robot to adopt adequate and useful behaviors while interacting with the dynamic environment.
*Types*: Behavior, Action, Strategy, Objective
An Activity entity is responsible for the achievement of a task that plays a role in the robot decisional process, using a set of Knowledge entities.

**Decision**: mechanism that computes coherent Reactions and Observations according to Intentions and Perceptions of any level of abstraction.

**Perceptions**: required pieces of information, or significant phenomena notification.

**Reactions**: results of decision that define command of any level of abstraction.

**Intentions**: goals that an activity intends to accomplish

**Observations**: interesting observed states of environment or robot.
A coordination entity is responsible for the way a set of Activity entities interact by exchanging or sharing Knowledge entities.

**Protocol**: mechanism that defines the way Activity entities interact and which Knowledge entities they exchange.
A System entity is an abstraction of the control of a physical entity (represented by a Body entity). It is responsible of the way a set of Activity and Coordination entities, concerning this physical entity, are organized in order to achieve a set of tasks.
Example: Aura

Environment -1-, Phenomena -2-, Interaction -3-, Body -4-, Behavior -5-, Action -6-, Objective -7-, Strategy -8-

Motors Schemas

Motors Schemas

Motors Schemas

Motor Schemas(4,1,5)

Perception Schemas(1,2)

Command Arbitration (5,7)

Composition Selection(1,5,7,8)

Stimuli Notification(1,5,7,8)

Request/reply (7)

Request/reply (1,7)

Request/reply (1,7)

Request/reply (6,2)

Command to actuators

Data from sensors

Infrastructure

Deliberative Layer

Mission Planner(1,7)

Spatial reasonner(1,7)

Plan Sequencer (2,6,1,7,8)

Schemas Controller (1,2,3,4,5,6,7,8)

Missions to actuators

Data from sensors

Long Term Environment Memory Sharing(1)

Reactive Layer

Request/reply (1,7)

Spatial reasonner(1,7)

Plan Sequencer (2,6,1,7,8)

Schemas Controller (1,2,3,4,5,6,7,8)

Mission Planner(1,7)
Conclusion

➢ A conceptual model for robotic architectures
  • Concepts used to decompose the robot decisional process
  • Basis of a domain specific modelling language (current work)

➢ A first step, but still many things to do:
  • Giving more description details? dedicated languages for:
    ✓ protocols and decisions?
    ✓ specific knowledge types?
    ✓ infrastructure specificities (sensors, actuators)?
  • Dealing with different degrees of genericity
    ✓ variability (optional concepts)
    ✓ specialisation (refinement of concepts)