Functional Architectures

Pedro U. Lima
M. Isabel Ribeiro

Institute for Systems and Robotics
Instituto Superior Técnico
Lisbon, Portugal

March 2007
References

INTEGRATION OF SUBSYSTEMS

- TASK PLANNING
  - MOTION PLANNING
    - desired trajectory
      - global map
      - local map
  - SELF-LOCALIZATION
    - estimated trajectory
      - environment features
      - map update
  - MAPPING
    - map update
      - environment features
  - SENSOR FUSION / COOP. PERCEPTION
    - obstacle detection
      - desired position and/or velocity
  - GUIDANCE
    - processed sensor information
  - CONTROL
    - actuator commands

- SENSORS
- VEHICLE
- ACTUATORS

Mobile robots only
INTEGRATION OF SUBSYSTEMS

- Autonomous mobile robots must generate plans to accomplish a mission or task
- Plans are composed of several primitive actions
- Primitive actions execution must be coordinated
- Plan generation, primitive actions coordination and primitive actions execution are integrated by a functional architecture
- Two major approaches to functional architectures for autonomous mobile robots:
  - Heterarchical (or horizontal, or subsumption)
  - Hierarchical
### INTEGRATION OF SUBSYSTEMS

<table>
<thead>
<tr>
<th>DELIBERATIVE</th>
<th>REACTIVE</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Reflexive</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Speed of Response</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Predictive Capabilities</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Dependence on Accurate, Complete World Models</strong></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DELIBERATIVE</th>
<th>REACTIVE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Representation-dependent</td>
<td>Representation-free</td>
</tr>
<tr>
<td>Slower response</td>
<td>Real-time response</td>
</tr>
<tr>
<td>High-level intelligence (cognitive)</td>
<td>Low-level intelligence</td>
</tr>
<tr>
<td>Variable latency</td>
<td>Simple computation</td>
</tr>
</tbody>
</table>

Reprinted from (Arkin, 1998)
HIERARCHICAL ARCHITECTURE

Mission command

PLANNING

TASK DECOMPOSITION

ACTUATORS

PERCEPTION

SENSORS
HETERARCHICAL ARCHITECTURE

- reason about behavior of objects
- plan changes to the world
- identify objects
- monitor changes
- build maps
- explore
- wander
- avoid objects

Sensors → Actuators
HETERARCHICAL ARCHITECTURES

Assembling Behaviors

Diagram: Perception/Stimuli $s$ leading to Behavior 1, Behavior 2, Behavior 3, ..., Behavior $n$, which are then coordinated by the Coordination Function. The output of the Coordination Function is the Response $R$. Each $g_i$ represents a state or decision point in the coordination process.
HETERARCHICAL ARCHITECTURES

Assembling Behaviors

Examples of Coordination Functions:

- **competitive**
  \[ R = \max\left(\text{inputs}\right) \]
  (e.g., max activation value, max number of votes)

- **cooperative**
  \[ R = \sum\left(\text{inputs}\right) \]
• Control is *layered* with higher levels subsuming the role of lower level layers when they wish to take control.

• The system can be partitioned at any level, and the layers below form a complete operational control system.
BROOK’S SUBSUMPTION ARCHITECTURE

Behavior module
BROOK’S SUBSUMPTION ARCHITECTURE

Level 0 control system
BROOK’S SUBSUMPTION ARCHITECTURE

Level 0 control system augmented with level 1
SARIDIS' 3-LEVEL HIERARCHICAL ARCHITECTURE

Petri Net Model for Coordination and Execution Levels
(Wang, Kyriakopoulos, Tsolkas and Saridis, 1990)
SARIDIS’ 3-LEVEL HIERARCHICAL ARCHITECTURE

PRINCIPLE OF INCREASING PRECISION WITH DECREASING INTELLIGENCE

precision and complexity increase when we move downwards along the hierarchy, while intelligence decreases.

MACHINE-INTELLIGENCE/PERFORMANCE IS MEASURED BY ENTROPY

entropy (Shannon) is a measure of information conveyed in a message. If we are more uncertain about the contents of a message, the message conveys more information. The higher level in Saridis’ Architecture (Organization Level) deals with more abstract (uncertain) knowledge, thus it deals with more information: entropy increases when we move upwards along the hierarchy. If we reduce the global entropy of an Hierarchical Controller, we are increasing its intelligence.

COMPLEXITY increases downwards because the time and space scales decrease: the time available for an operation is reduced, while the spatial resolution required is increased.
HIERARCHICAL CONTROL is an old and proven organizing concept, based on management theory and used by military, government and business. It is based in successive partitions of the problem domain, leading to a more efficient division of labor

SPATIAL RESOLUTION manifests itself in the span of control and resolution of perception.

TEMPORAL RESOLUTION manifests itself in terms of bandwidth, sampling interval, planning horizon and temporal length of historical traces.
RCS APPLICATION TO AUVs

(Albus, 1987)

MULTIPLE AUTONOMOUS UNDERSEA VEHICLES (MAUV) PROJECT
(developed at the U.S. National Institute of Standards and Technology, 1987)

Objective: to demonstrate intelligent cooperative behavior in multiple autonomous undersea vehicles

Approach: to build a control system architecture which fully integrates concepts of artificial intelligence and game theory with those of modern control theory

Research Issues:
• hierarchical distributed control
• knowledge-based systems
• real-time planning
• world modelling
• value-driven reasoning
• intelligent sensing and communication
• gaming
• cooperative problem solving by two intelligent vehicles in a natural and potentially hostile environment
RCS APPLICATION TO AUVs

(Albus, 1987)

Demonstration scenarios:

1. Cooperative search and map:
   - one vehicle searches the area while other relays messages about what has been found

2. Cooperative search and attack:
   - one vehicle illuminates the target while the other takes action against the target
   - one vehicle actively hunts for the target, while the other lies in wait
   - one vehicle attracts the target attention, while the other closes in for the kill
   - one vehicle occupies the enemy defences, while the other slips past unnoticed
   - one vehicle draws attention to itself, while the other escapes with valuable information
RCS APPLICATION TO AUVs

(Albus, 1987)

- 3- (4-, in a later version) legged, 6 level (in the example) hierarchy of computing modules for
  - task decomposition
  - world modelling
  - sensory processing
  - value judgement

- the hierarchy is serviced by a communications system and a distributed common memory (blackboard)

**task decomposition modules**: real-time planning and task monitoring. Tasks goals are decomposed both spatially and temporally.

**sensory processing modules**: filter, correlate and integrate sensory information over both space and time, so as to detect, recognize and measure patterns, features, objects, events and relationships in the external world.

**world modelling modules**: answer queries, make predictions and compute evaluation functions on the state space defined by the information stored in global memory. They serve both the task decomposition and sensory processing modules. The *global memory* is a database which contains the system best estimate of the state of the external world. The world modelling modules keep the global memory database current and consistent.
RCS APPLICATION TO AUVs

(Albus, 1987)

FIGURE 5: High level block diagram of MAUV RCS-3 control system architecture.
RCS APPLICATION TO AUVs

(Albus, 1987)

FIGURE 6: Each task decomposition H module decomposes task both spatially and temporally.

FIGURE 7: Each sensory processing G module compares observed signals with world model predictions and performs temporal and spatial integration.
RCS APPLICATION TO AUVs

(Albus, 1987)
RCS APPLICATION TO AUVs

(Albus, 1987)
The architecture is **hierarchic** in the sense that commands and status feedback flow hierarchically up and down the chain of command, and also in the sense that sensory information is organized in increasingly higher levels of abstraction, and that information stored in the world model is organized hierarchically.

The architecture is **heterarchic** in the sense that data is shared horizontally between heterogeneous modules at the same hierarchical level, i.e., between *task decomposition*, *sensory processing* and *world modelling* modules.
RCS APPLICATION TO AUVs

(Albus, 1987)

FIGURE 9: A block diagram of the MAUV task decomposition hierarchy.

GROUP TASKS

5

GROUP 1

GROUP 2

GROUP M

VEHICLE TASKS

4

VEHICLE 1

VEHICLE 2

VEHICLE K

E-MOVE KEY POSES

3

PILOT

SONAR

COMMUNICATIONS

CLEAR PATH POINTS

2

VEHICLE DYNAMICS

BEAM POINTING

MESSAGE STRING

DYNAMIC xyz TRAJECTORIES

1

THRUSTERS

TRANSMIT RECEIVE

TRANSMIT RECEIVE

ACTUATOR POWER

ACTUATORS
RCS APPLICATION TO AUVs

(Albus, 1987)

DEFINITIONS

Task: Activity which begins with a start-event and is directed towards a goal-event.

Goal: is an event which successfully terminates a task.

Command: instruction to perform a task.

Plan: Set of activity-event pairs which is designed to accomplish a task and produce a goal event. Each activity in the plan leading to the goal is a planned subtask, and the event terminating each of the planned subtasks is a subgoal.
RCS APPLICATION TO AUVs

(Albus, 1987)

PLAN REPRESENTATION:

- Gantt notation
- Petri-nets
- State-machines (state-graphs here)

Petri-nets implicitly allow generation of concurrent activities. The language they can generate is more general than the language state-machines can. However, state-machines are easier to implement, and we may define a separate state-machine for each parallel subsystem.

Example: simple plan for a search-and-attack mission, represented as a state machine (graph).
RCS APPLICATION TO AUVs

(Albus, 1987)

FIGURE 12: A plan may consist of several concurrent strings of subtasks which collectively achieve the goal event.
RCS APPLICATION TO AUVs

(Albus, 1987)

FIGURE 15. A state-graph plan for decomposition of the <Fetch (A)> command. (From reference 10)
RCS APPLICATION TO AUVs

(Albus, 1987)

FIGURE 19. Role of world model in planning. Hypothesized actions are "What if?" questions.
RCS APPLICATION TO AUVs

(Albus, 1987)

FIGURE 33. Role of M modules in responding to H module executor “What is?” questions.
RCS APPLICATION TO AUVs

(Albus, 1987)

FIGURE 32. Role of M module in predicting sensory input and in updating knowledge base based on correlations and differences between predictions and observations.
RCS APPLICATION TO AUVs

(Albus, 1987)

**FIGURE 24.** The H module at each level has three parts. A planner manager module PM, planners PL and set of executors EX.

**FIGURE 25.** Internal structure of the planner manager and planners.
RCS APPLICATION TO AUVs

(Albus, 1987)

Hierarchical Planning

FIGURE 28. Three levels of planning activity in RCS-3.
SOFTWARE ARCHITECTURES FOR MULTI-ROBOT SYSTEMS

REQUISITES:

- Flexibility in module replacement
- Multi-plataform (OS and mechanical hardware)
- Concurrent programming (soft) real-time support
- Event-driven
- Usability
  - Generic modules for robotic applications
  - Behaviour-Based programming with Graphical-User Interface (GUI)
ARCHITECTURES FOR MULTI-ROBOT SYSTEMS

HIGH-LEVEL ARCHITECTURE

- reduce programming effort
- constrain solutions to chosen framework
- use high-level graphical programming interfaces

MIDDLEWARE

- concurrent multi-platform programming
- modularity
  - for flexible module replacement
  - for easy module edition/modification
- performance
- independence from robot hardware
MeRMaID ARCHITECTURE

(Barbosa, Ramos, Lima, 2007)

User “code”

MeRMaID

MeRMaID::Support

Operating System

Task-specific code and behavior specification

High-Level Architecture generic software components

OS abstraction, Communications and other supporting Frameworks

Linux, Mac OS X, Windows
MeRMaID ARCHITECTURE

(Barbosa, Ramos, Lima, 2007)

User “code”

Task-specific code and behavior specification

MeRMaID

High-Level Architecture generic software components

MeRMaID::Support

OS abstraction, Communications and other supporting Frameworks

Operating System

Linux, Mac OS X, Windows
MeRMaID ARCHITECTURE

WISDOM

World Info

Event Generator

CORTEX

Team Organizer

Behavior Coordinator

Behavior Executor

ATLAS

Information Fusion

Sensors

Impact on environment

Primitive Actions

Navigation Primitives

Raw data from environment (includes internal robot data)