Architecture for Autonomy

Sara Fleury & Félix Ingrand
(sara@laas.fr  felix@laas.fr)

LAAS-CNRS
(www.laas.fr)
7 avenue du Colonel Roche
F-31077 Toulouse FRANCE
Motivations

- **Methodologies** and **tools** to design embedded software architectures for autonomous systems
- Results taken from **robotic research**
- Application to new other systems:
  - **on board execution control and mission management**
    - ground station maintenance simplified
    - flexibility and high level interactions
- Autonomy => reactive + decision making capabilities
- Architecture properties:
  - programmability
  - adaptability
  - consistent behavior
  - robustness
  - reactivity
  - extensibility/reusability
The 3 levels LAAS architecture: from decision to action

1. Decision Level
   *(planning and supervision of action)*

2. Execution Control Level
   *(actions coordination)*

3. Functional Level
   *(actions execution)*

Logical System

Physical Platform

Sara Fleury & Félix Ingrand, LAAS/RIA, © 2000
From autonomous mobile robots ...

- Diligent at an exhibition
- an Hilare robot with a trailer and a 6dof arm
- Planetary exploration robotics
- Service robotics
- Transshipment robotics
- the rover Lama
- SNCF robot Commutor
- new harbour of Rotterdam
... to autonomous satellites
LAAS level 1: the Functional Level

Integrates all the operational functions
(hardware control, servo-control, data processing, …)

Structured as a set of independent **modules**
(dynamically controlled by the upper level)

Module: entity responsible for a physical or logical resource

upper level or operator

Modules

a module

data

to other modules or hardware devices

data

Request

Reply

Sara Fleury & Félix Ingrand, LAAS/RIA, © 2000
The Generator of Modules GenoM

- Automatic code synthesis
- No need to know the underlying OS
- One can concentrate on the functionalities
- Incremental design

1. module description

```plaintext
module Motion {
    number: 9600;
    SDI: MOTION_DATA;
}
request SetPos {
    type: control;
    input: pos::pos;
    control: controlPos;
    report: BAD_PARAM;
}
task Move {
    period: 25;
    priority: 15;
}
```

2. module generation

GenoM

- parser
- generic module instantiation
- compilation
- link editing

executable module (various OS)

Interface libraries (C, Propice, TCL, ...)

test programs

3. algorithms integration

4. tests

Sara Fleury & Félix Ingrand, LAAS/RIA, © 2000
LAAS level 2: the Execution Control level

- **Pivot** between functional/decision levels
- Purely **reactive system** that reacts to decision level requests and functional level replies
- **State controller** of function level:
  - maintains functional level state
  - filters decision level requests
  - detects and manages conflicts
  - check resources usage
- **Tool** **Transgen/Exogen**: automatic BDD from a set of constraints/rules (consistent, optimised)
- Offline model checking with temporal logic
LAAS level 3: the Decision Level

- All processes that require anticipation and global knowledge of the task and of the execution context.
- Structured in supervisor/executive-planner layers:

**Supervisor/Executive:**
- Interprets upper mission
- Selects action procedures (or call planner)
- Controls the procedures execution
- Reacts to events (replies) from lower level
- Tool: PROPICE

**Planner:**
- Queried by supervisor
- Deals with:
  - time constraints
  - resources constraints
  - predictable events
- Produces plan of actions
- Tool: IxTeT
Properties:
- high-level language
- parallel tasks + asynchronous events handling
- temporal properties

Main components:
- automatically updated database (view of the world)
- a library of procedures:
  - sequence of actions and tests
  - to achieve given goals, or
  - to react to certain situations
- a dynamic task graph

Example of a PROPICE procedure
IxTeT: IndeXed Time Table

- IxTeT kernel: an efficient time-map manager
- Time-point algebra relations and restricted interval algebra
- Used in situation recognition and plan synthesis
- Common knowledge representation: chronicles

Example of an IxTeT plan