Orccad Control Architecture

K. Kapellos TRASYS
R. Pissard-Gibollet, D. Simon, B. Espiau INRIA Rhone-Alpes
ORCCAD: A Three-Layered Control Architecture

- Discrete Event Systems
- Plans correct by construction

- Event based specification
- Deterministic behaviour
- Formal methods and tools

- Automatic control theory
- Efficient real-time implementation
Plan

- Functional Layer
  - Actions
  - Tools
- Execution Layer
  - Task
  - Tools
- Decision Layer
  - Concepts
  - Tools
Functional Layer – Action definition

• An Action is defined as the complete and parameterised specification of:
  – A control law
  – A local behavior rythmed by events
  – Temporal constraints
• Examples: move_mast_to(), take_image(), …
Functional Layer – Action Implementation

- Real-Time Multi-rate, multi-task Implementation
  - Computation durations
  - Activation period
  - Priorities
  - Assigned processor
Functional Layer – The Actions specification Tool

- Development under Eclipse
  - Plug-in for specification and code generation, connections with external modelling tools, …. 
  - Use of XML format for data specification
The Execution Layer

Decision Layer: Plans

Execution Layer: Tasks

Start_A1, Abort_A1, ...,
Exceptions_A1

Start_An, Abort_An, ..., Exceptions_An

Functional Layer: Actions
Execution Layer – Task Definition

- Task is defined as a logical and temporal composition of Actions and Tasks.
  - Examples: CriticalDeployment, Travel, Drill, …

```java
do
[
  DrillMoveTo();
  CloseImagerMoveTo();
  [
    CloseImagerMonitor()
    ||
    DrillExtractSample()
  ]
] watching Alarm do
....
```
Execution Layer – Formal Framework

- Reactive systems
  
  ![Reactive system diagram]
  
  - Behavior of the system
    - All allowed sequences of the input/output events
    - Synchronous approach
      - The duration of the system reaction is negligible wrt the input signal occurrences

    - ESTEREL language
      - Dedicated for reactive systems programming
      - Compilation to finite state automata
      - Tool for simulation, verification and code generation
Execution Layer – The FORMID Tool

- Specification
  - Selection of Actions/Tasks
  - Provide the ESTEREL code that specifies the sequencing of the involved Actions/Tasks

- Esterel modules definition, declarations, … are automatically produced
Execution Layer – The FORMID Tool

- Task Simulation and Formal Verification
- Relationship between Actions/Events and Actions
  - The execution of the Action/Task A1/T1
    ... triggers ...
    ... is always preceded by ...
    ... always takes place during ...
    ... always implies ...
    the execution of the Action/Task A2/T2
  - The occurrence of the Event event
    ... triggers ...
    the execution of the Action/Task A2/T2
Fault detection and Recovery

- **Decision Layer: Plans**
- **Execution Layer: Tasks**
- **Functional Layer: Actions**

Types of faults and recovery strategies:
- **exc type 1**
- **exc type 2**
- **exc type 3**

Replanning process:
- Detect fault
- Decide on replanning
- Execute replanning

Diagram illustrating the layers and fault types.
Decision Layer: Supervisory Control Theory

- **Objective**: Create a new sequence of Tasks (a Plan) in case of a (re)-planning request ensuring its correctness wrt a set of constraints

- **Models**: Discrete Event Systems
- **Properties & Objectives**: Transition systems that express constraints on the behaviour of the system
- **Events**: distinguished between controllable and uncontrollable

*Derivation of a controller such that the resulting behavior of the closed-loop system meets the control objectives*
Decision Layer: The basic entities

Decision Layer

 Execution Layer: Tasks

---

1 Task 1

2 Task n
Decisional Layer: Tools

- Supremica [Prof. K. Akesson – Chalmers University]

- HMI for generating the model of the system, the constraints and the objectives
- Algorithms to synthesise the controller  (*monolithic, BDD based and compositional*)
Orccad and Associated Supervision Tools

- Linked with the generated code via DREAMS