A Semantics of Communicating Reactive Objects with Timing

Jozef Hooman & Mark van der Zwaag

University of Nijmegen

Katholieke *Universiteit* Nijmegen

The OMEGA Project

OMEGA project:

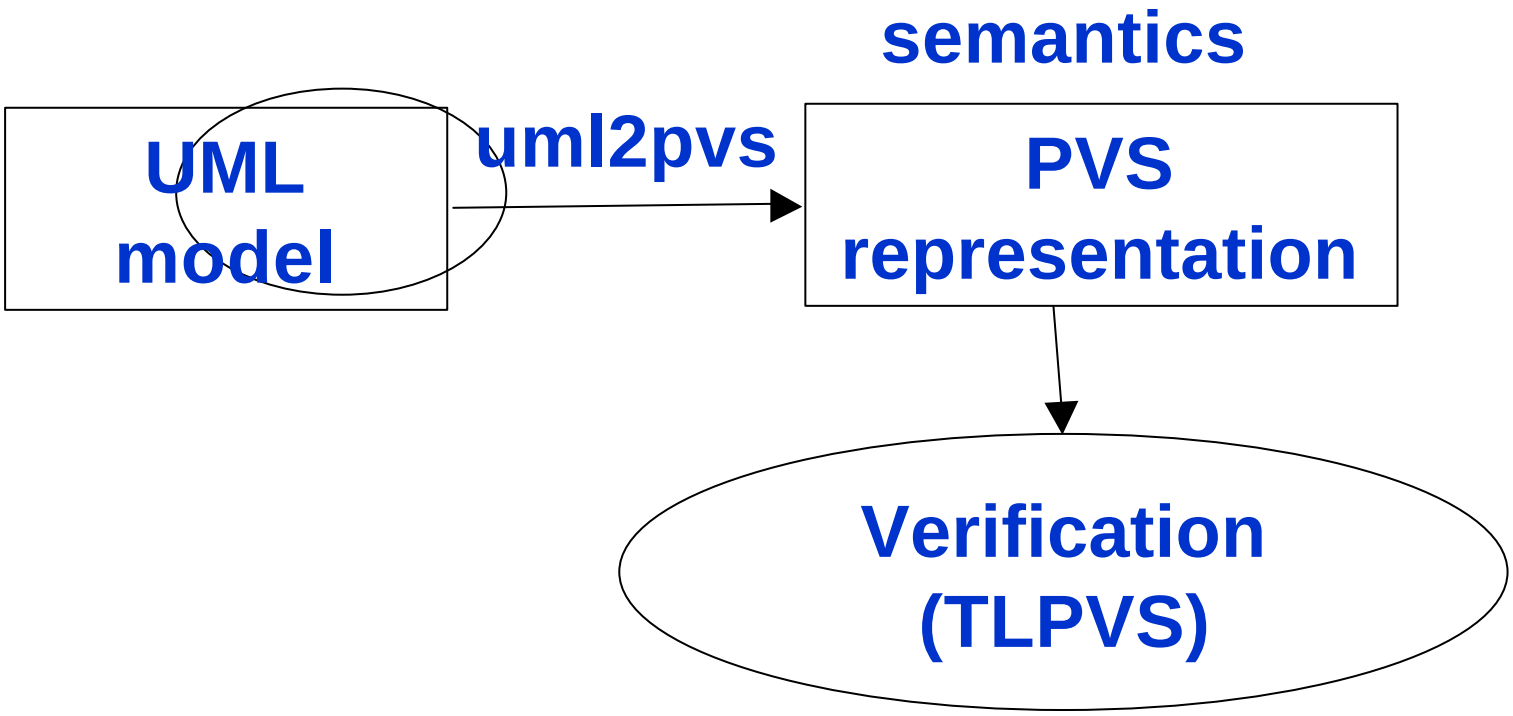
Correct Development of Real-Time Embedded Systems

Partners:

- **Verimag, Grenoble, France**
- **OFFIS, Oldenburg, Germany**
- **CWI, Amsterdam, Netherlands**
- **CAU, Kiel, Germany**
- **KUN, Nijmegen, Netherlands**
- **WIS, Rehovot, Israel**
- **Industrial partners (IAI, EADS, NLR, FT)**

Theorem Proving

CAU, CWI, WIS and KUN collaborate on tool support for theorem proving in PVS



Approach

- **Input: UML model expressed in OMEGA kernel language, i.e. class diagrams + state machines**

  This includes:
  - **Asynchronous signals (and queues)**
  - **Synchronous operation calls**
  - **Active classes (thread of control)**
  - **Timing annotations**
- **The semantics defines a Labelled Transition System (LTS) for the input model**
- **Verification: prove correctness properties (safety, liveness, timing) of execution traces (runs) of LTS**

System Semantics

System behaviour is concurrent behaviour of all state machines of the objects in the system

Global step corresponds to:

- **A state machine action of one of the objects (possibly involving other objects as well,**
  e,g, for synchronous operation calls)
- **The discarding of a signal by one of the objects**
- **A global time delay step (all other steps do not take time)**

Semantic issues

Main semantic issues discussed:

1. **Meaning of active objects and thread of control**
2. **Operations: triggered vs primitive, and relation with control**
3. **Signals: handling of queues and relation with operations and control**
4. **Timing**

Not elaborated here:

- **Object creation/destruction**
- **Generalization (inheritance)**

1. Activity Groups

Each object belongs to an activity group that has exactly one active object.
In every group exactly one object has control.
An object needs control to execute actions.

We have a "run to completion" semantics:
An object cannot loose control before it has reached a stable state (unless it calls a triggered operation of a group member).

2. Operations

- **Primitive operation: result value is completed locally in one atomic step which includes assignment of result value to attribute of caller**
  Decision: callee may be unstable or suspended

- **Triggered operation:**
  1. **Synchronization of caller and callee**
  2. **Caller becomes suspended until return value is received**
  3. **Callee must be stable**
  4. **No re-entrance**

Control & Triggered Operations

Control requirements and changes:

- **If caller and callee in same group:**

  control passed to callee,

  after return of result value, caller gets control back

  when callee has become stable

- **If caller and callee in different groups:**

  caller remains in control, callee must have control

Note: control requirements and control changes are rather complex; challenge to represent this conveniently in PVS to enable verification

3. Signals

Signals:

- Each object has its own signal queue

  (alternative: each activity group has queue)
- An emitted signal is placed in the signal queue of the addressed object; no requirements on receiver.
- A signal can be accepted (i.e. trigger a transition) from the queue if object is stable and has control
- A signal can be discarded from the FIFO queue if the object is stable and has control, and the signal does not trigger a transition from the current state

  (if it is deferrable, it remains in the queue)

4. Time

Timing: adding time is orthogonal to previous issues

We follow standard approach of Timed Automata:

- **Every object has local clocks which it can read and reset; guards may depend on clock values**
- **Global delay steps**
- **Clock invariants on state machine locations**

We allow the (global) passing of time in between the actions of a transition

Concluding Remarks

- Modeling of passing of control, in combination with operations and signals, in PVS revealed many ambiguities in the first OMEGA semantics and raised many new questions
- Adding time is orthogonal to the issues above

Current work includes:

- Verification of industrial case studies in PVS
- Interpretation of OMEGA timing annotations
- Compositional semantics