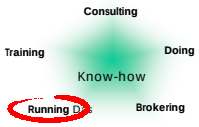
CASSANDRA/xUML

Executable UML Specifications

UML today

The Unified Modeling Language (UML™) not only allows us to model a specific implementation of a software system, but also to specify its functionality in an implementation-independent way. However, in practice such analysis models become all-to-often heavily "polluted" by design and implementation elements, since the software finally must be implemented in a specific programming language and on a specific operating system.

UML tomorrow



By turning UML models into precise **xUML** models (**Executable UML**), it becomes possible to define an environment in which such implementation-independent analysis models can be executed without considering technical details of the target platform. This results in the following advantages:

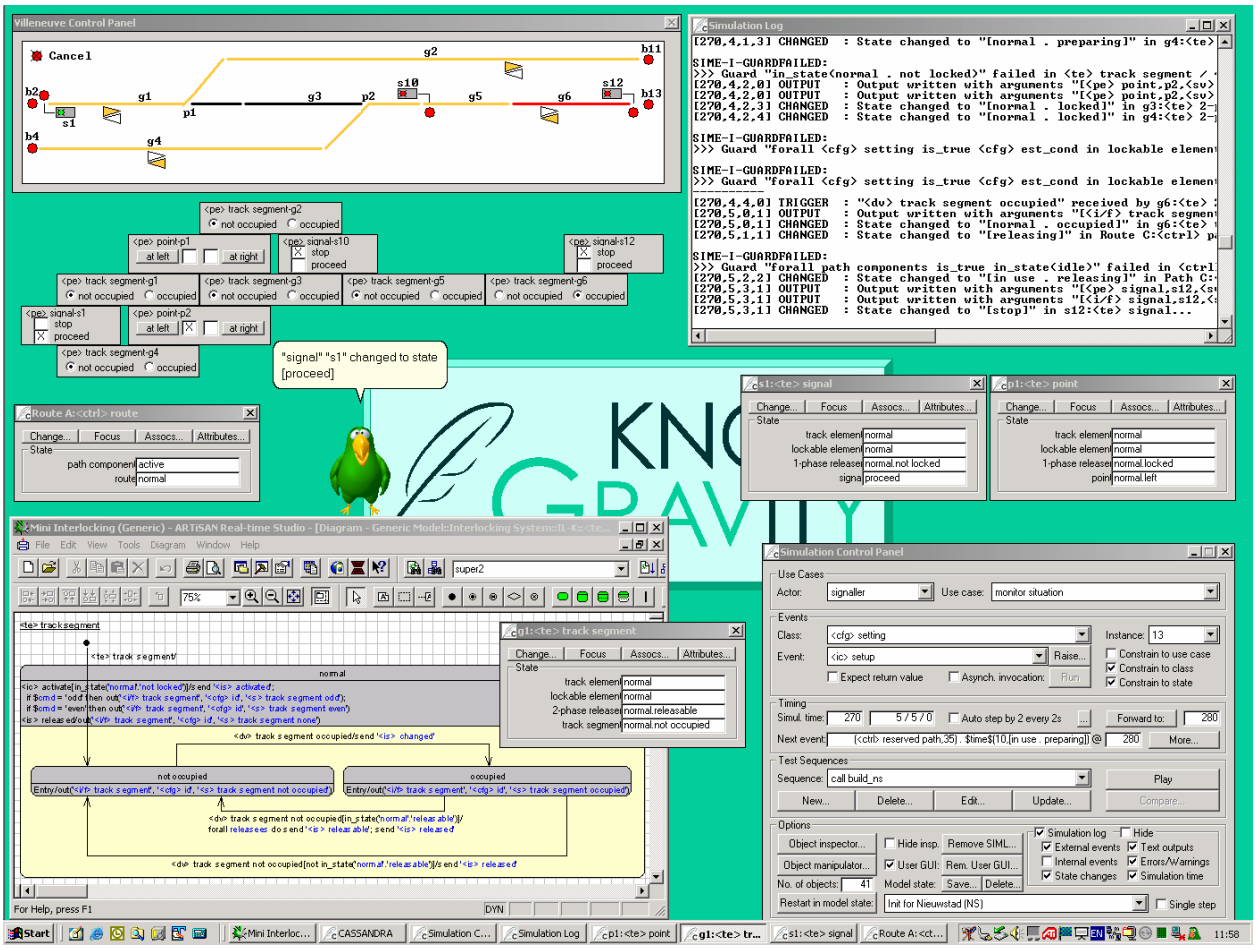
- The functionality of the system can be **tested, before the first line of code has been written**.
- **Searching and fixing errors** in the application logic can be carried-out **on the abstraction level of UML models** instead of the level of program code.
- The **semantics of UML models** can be "felt" and thus becomes much better understandable and **learnable**.

CASSANDRA/xUML

Based on our software engineering platform CASSANDRA we developed a simulation environment that allows the convenient execution of xUML models. This environment provides the following features:

- **Simulation of**
  - the user view represented by use case models and sequence diagrams
  - the static system structure by class models with attributes, associations and (multiple) inheritance
  - the system behavior by means of state diagrams, including super states and concurrent states.
- A powerful **Action Language** for state transitions supports
  - creation and destruction of instances
  - establishing and loosing association-links between instances
  - navigation along associations (and association classes)
  - boolean, arithmetic and set-expressions
  - all and existence quantifiers
  - synchronous and asynchronous communication via events
  - event broadcasts via associations
  - time and condition-events.
- The following tools are provided for **Model-Level Debugging**:
  - a logger to record important events during a simulation run
  - object inspectors for in-situ observation of instances
  - an object manipulator to manipulate instances and their association links
  - single step and autostep simulation
  - a **regression tester** to record test scenarios and replay them automatically including automated verification of the system behavior.
- Simple integration of a **dedicated user interface**.

The screen shot below shows a session with CASSANDRA/xUML to simulate a **railway interlocking system** (with kindly permission of the EURO-Interlocking Project of the International Union of Railways, UIC).



What's next?

As soon as a precise and executable functional specification of a system is available, a whole set of new opportunities emerge:

- The specification can be considered as a PIM (Platform Independent Model) in the sense of OMG's **Model Driven Architecture (MDA®)** and thus be translated into a Platform Specific Model (PSM), i.e. into a final implementation.
- Based on the explicit information contained in such a specification, some less explicit information about the model can be derived that in turn can be used to **validate the specification**.
- A precise specification may be used as a base to apply **formal verification techniques** to prove certain safety-critical properties of the system.

Currently these types of CASSANDRA applications are subject to our research.

Requirements

To run CASSANDRA/xUML the following technical requirements must be fulfilled:

- CASSANDRA, V1.5
- CASE Tool ARTiSAN RtS, V4.1 or V4.2

Support for other CASE Tools on request.

Contact

KnowGravity Inc
Badenerstrasse 808
8048 Zürich
Switzerland

Phone +41 (0) 1 434 20 00
Fax +41 (0) 1 434 20 09
Internet www.knowgravity.com
Email info@knowgravity.com