System Behaviour Analysis with UML and Ptolemy

4th Biennial Ptolemy Miniconfence
22-23 March 2001

Xavier Warzee, Jean-Charles Causse
Thales Optronique
and
Dominique Marcadet,
Julien Raimbault, Mickaël Levier, Supelec

Scope and goals

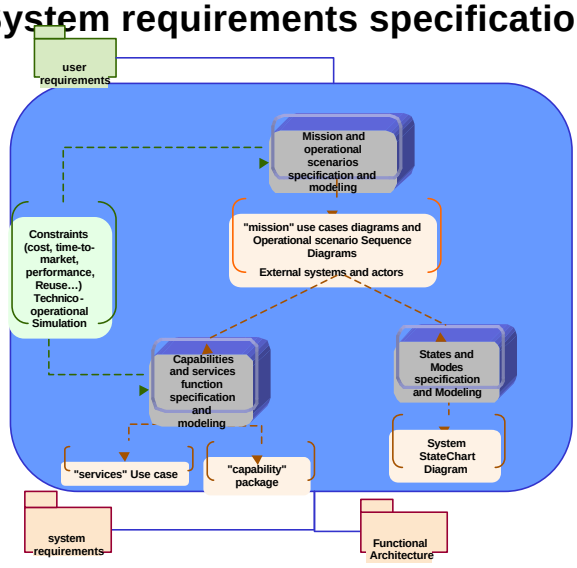
- Complex System analysis and design
  - Requirements hard to capture:
    - more and more missions assigned to systems (complex supervision),
    - more and more behavioral requirements,
    - more and more capabilities to support (heterogeneous systems).
  - Models used to capture requirements !
  - Current practices:
    - **UML** used to capture requirements,
    - DOORS/Telelogic used for tracability between requirements and models.
  - How to describe unambiguously operational scenarios ?
  - How to check dynamic properties such as Concurrency ?
- How to check early the correctness and the completeness of these models ?

System requirements specification with UML

- Capture missions of the system
  - Static models:
    - definition of scenarios with the following types of entity:
      - external actors,
      - system capabilities (e.g. target detection),
      - functions (e.g. target tracking, range computing),
      - components (e.g. laser, camera, sensors, boards).
  - Dynamic models:
    - Use Cases
      - interactions between the system and external actors.
    - Message Sequence Charts, Activity Charts
      - interaction between entities of the system.
    - StateCharts
      - States and modes.

System requirements specification

User View



Executable UML specifications

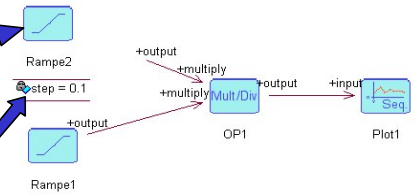
- Gap between the static and dynamic models:
  - how to check consistency and completeness ?
  - First answer with Rhapsody from Ilogix :
    - Animation of UML models,
    - Limitations:
      - ? background in software and Object-Oriented technologies needed !
      - ? Only one type of semantics for animation (StateCharts).
  - Esterel Studio
    - powerful solution for reactive systems,
    - formal verification of the control part based on the Esterel language,
    - Limitations:
      - ? Only one type of semantics (the Synchronous hypothesis).

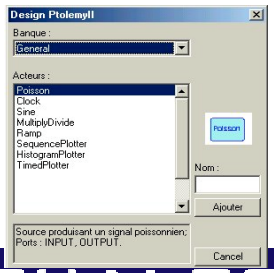
From UML to Ptolemy II

- Use of Ptolemy II to execute UML models
  - UML
    - MSCs to describe scenarios,
    - stereotypes to capture capabilities, functions,
    - statecharts to capture modes and states.
  - Generation of MoML files from the UML models
    - use of Rose/Rational
      - ? plug-in developed to parse the UML models,
    - generation of XML files,
    - launch of Ptolemy II to execute the models (now Ptolemy II models),
    - (optional) addition of blocks to generate inputs, and display outputs,
    - (optional) addition of other models if not done in UML,
      - ? Continuous Time models to simulate physical values evolving during simulations.

Rose Add-in : SDF and DE domains

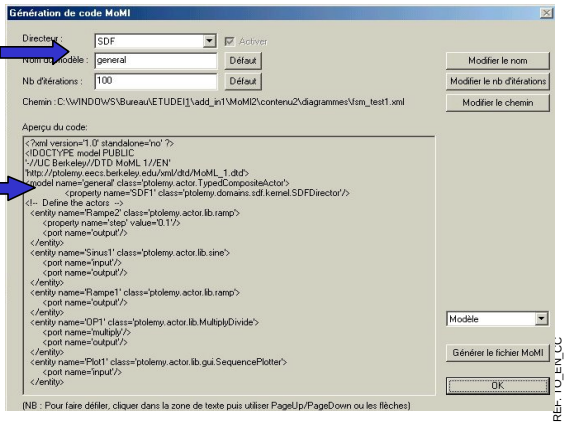
- Use of UML with specific rules
  - Choice of Class Diagram
  - UML Classes as Ptolemy actors
    - One UML class stereotype for each Ptolemy class
    - UML class attributes describe Ptolemy actor properties
  - UML associations as Ptolemy Links
    - UML association roles describe Ptolemy actor ports and port properties
- Creation of a user interface
  - Inserting actors into the diagram
  - Hierarchical organization of the actors
    - Modification
    - Definition
    - Deletion





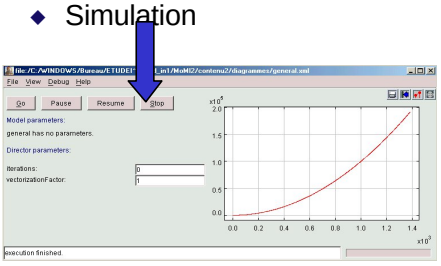
Rose Add-in : SDF and DE domains

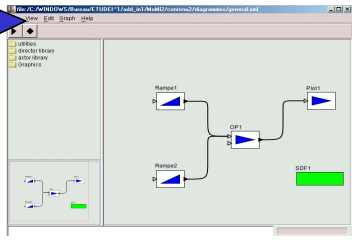
- Code generation
  - Diagram parsing
  - Parameters management
    - Choice of a domain director (SDF, DE)
    - Model or class
  - XML code overview
  - Exporting XML code to a file



Rose Add-in : SDF and DE domains

- Interaction with Ptolemy II
  - Importing XML file in Vergil
  - (optional) Modifications through Vergil interface
  - Simulation

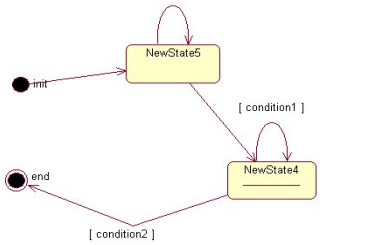


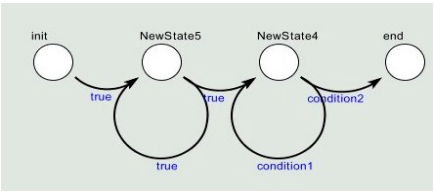


REF. TO_EN_CC

Rose Add-in : FSM domain

- Mapping between UML StateCharts and FSM domain
  - Under Construction
  - Whereas DE and SDF domains needed rules to be described in UML, a simple Mapping is possible between UML StateCharts and Ptolemy Finite State Machines

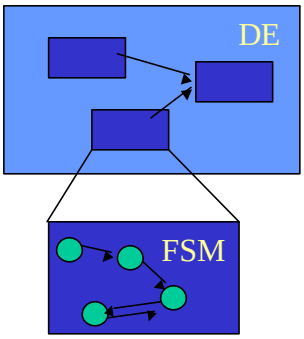




REF. TO_EN_CC

Rose Add-in : Goals

- Mapping between actions' meaning in StateCharts and FSM actions
- Management of hierarchical graphs using several domains (SDF, DE and FSM)



UML and Ptolemy II : perspectives

- Integration of other Ptolemy II domains
- XSLT to convert from XMI to MoML
- XSLT to convert from MoML to XMI
  - => From Ptolemy II to UML.
- Use of RoseRT instead of Rose ?
  - Mapping of UML-RT capsules and ports to Ptolemy II actors and ports.