A foundation for MDE

Ernesto Posse   Juergen Dingel

Applied Formal Methods Group
Software Technology Lab
School of Computing
Queen's University
Kingston, Ontario, Canada

September 28th, 2008

Outline

1. Introduction

2. An example: UML-RT/RoseRT

3. Conclusions

Outline

1. Introduction
2. An example: UML-RT/RoseRT
3. Conclusions

Introduction

- MDE promises to improve software development by focusing on models
- Many successes and failures
- Problems:
  - Lack of adoption
  - *Ad hoc* methods, tools and applications

    ... exacerbated by ...
  - a lack of foundations
- Grand challenge: develop a foundation for MDE

Introduction

- Foundations of MDE ⟹foundations of modelling languages
- Foundations of modelling languages:
  - syntax,
  - semantics,
  - pragmatics
- Consider UML-RT and RoseRT as an example of:
  - what can be achieved
  - what is missing

Outline

1 Introduction

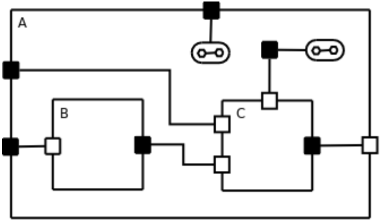
2 An example: UML-RT/RoseRT

3 Conclusions

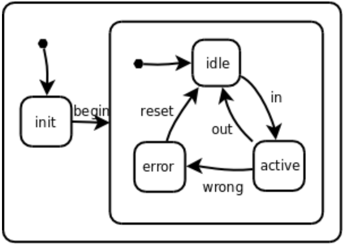
UML-RT and RoseRT

- UML-RT: a UML profile for real-time, distributed and embedded systems.
- RoseRT (now Rational Technical Developer): UML-RT implementation
  - modelling environment,
  - code generation,
  - (some) analysis

UML-RT models



A capsule diagram



A state machine diagram

Why is UML-RT/RoseRT successful?

- Simple notation/syntax
- Manageable mix of effective features which guarantee strong properties that simplify design and analysis:
  - strong encapsulation with interaction via ports,
  - run-to-completion semantics
  - no orthogonal regions
  - etc...
- Tool support
  - good use of models: automatic code generation, (limited) analysis,
  - flexibility

What is missing?

- Features:
  - Action language
  - Behavioural interface specifications
  - "Real" real-time, schedulability analysis,
  - ...
- Tools:
  - Better analysis
  - Debugging/animation, etc.
- Questions:
  - How does multi-threading interact with RTC semantics?
  - ...
- Formal semantics

Outline

1. Introduction
2. An example: UML-RT/RoseRT
3. Conclusions

Solutions

- Identification of core features
- Clarify their semantics and fundamental properties
  - our approach for real-time systems: real-time process algebra
  - UML: a λ-calculus of software modelling
- Study impact of model transformations on properties
- Study language patterns

Conclusions

- Even successful approaches to MDE have a lot of room for improvement
- We need more research in foundations
  - … more specifically we need research on modelling languages
  - … with special focus on formal semantics and pragmatics

Acknowledgments

We would like to thank:

- Bran Selic
- Major Ron Smith
- National Sciences and Engineering Research Council of Canada (NSERC)
- Ontario Centers of Excellence (OCE)
- IBM