RIO : Model, Language and Tools for Agent-Oriented Software Engineering

BAALA Oumaya BAKHOUYA Mohamed BERDAI Abdelhamid CAMINADA Alexandre CREPUT Jean-Charles GABER Jaafar GALLAND Stéphane GARROUTY Gilles GRUER Pablo HAJJAM-EL-HASSANI Amir HARITI Mohamed HAYARI Noureddine HILAIRE Vincent KOUKAM Abderrafiaa LAMOTTE Olivier LAMROUS Sid Ahmed LISSAJOUX Thomas MABED Hakim RODRIGUEZ Sebastian RUICHEK Yassine SIMONIN Olivier

UTBM
Belfort Technopôle 90000 Belfort - France

Agent Link III TFG-AOSE

Outline

Introduction

The RIO meta-model

The OZS notation

Tools : prototyping and verification

Conclusion

Outline

Introduction

The RIO meta-model

The OZS notation

Tools : prototyping and verification

Conclusion

Statements

- Many models and architectures for agent and multiagent systems
- Heterogeneous engineering approaches
- Understanding model is difficult
- Comparison and reuse are difficults

Comparison critera

Model

- Concepts underlying the model
- Architecture dependant

Language

- Formal
- Semi-Formal

Tools

- Validation
- Verification
- Code generation

Model

- Organizational
  [Wooldridge et al., 1999, Ferber and Gutknecht, 1998, MESSAGE, 2000, Burrafato and Cossentino, 2002]
- Componential [Brazier et al., 1997]
- Architecture dependant
- Application range

Language

Semi-Formal

- Mainly extensions of Object-Oriented notation : AUML [Bergenti and Poggi, 2000], MASE [DeLoach, 1999], ADELFE [Bernon et al., 2002]

Formal

- Formal Method [Luck and D'Inverno, 1995],
- Modal and Temporal Logic GAIA [Wooldridge et al., 1999]
- Knowledge Representation DESIRE [Herlea et al., 1999]

Advantages of formal approach

- Non ambiguous semantics.
- Precise specifications.
- Proof, simulation, prototyping.

Drawbacks of formal approach

- Difficult to produce an implementation.
- Limited expressive power (reactive aspects, transformational aspects, ...).

Tools

- Model checking [Herlea et al., 1999, Gruer et al., 2000]
- Proofs
- Prototyping [Hilaire et al., 2000a, Hilaire et al., 2000b]

Needed for a methodology

Objectives

- Concepts suited for description of a wide range of MAS.
- Covering phases from formal specification to implementation.
- Validation and Verification.

Outline

Introduction

The RIO meta-model

The OZS notation

Tools : prototyping and verification

Conclusion

Definitions

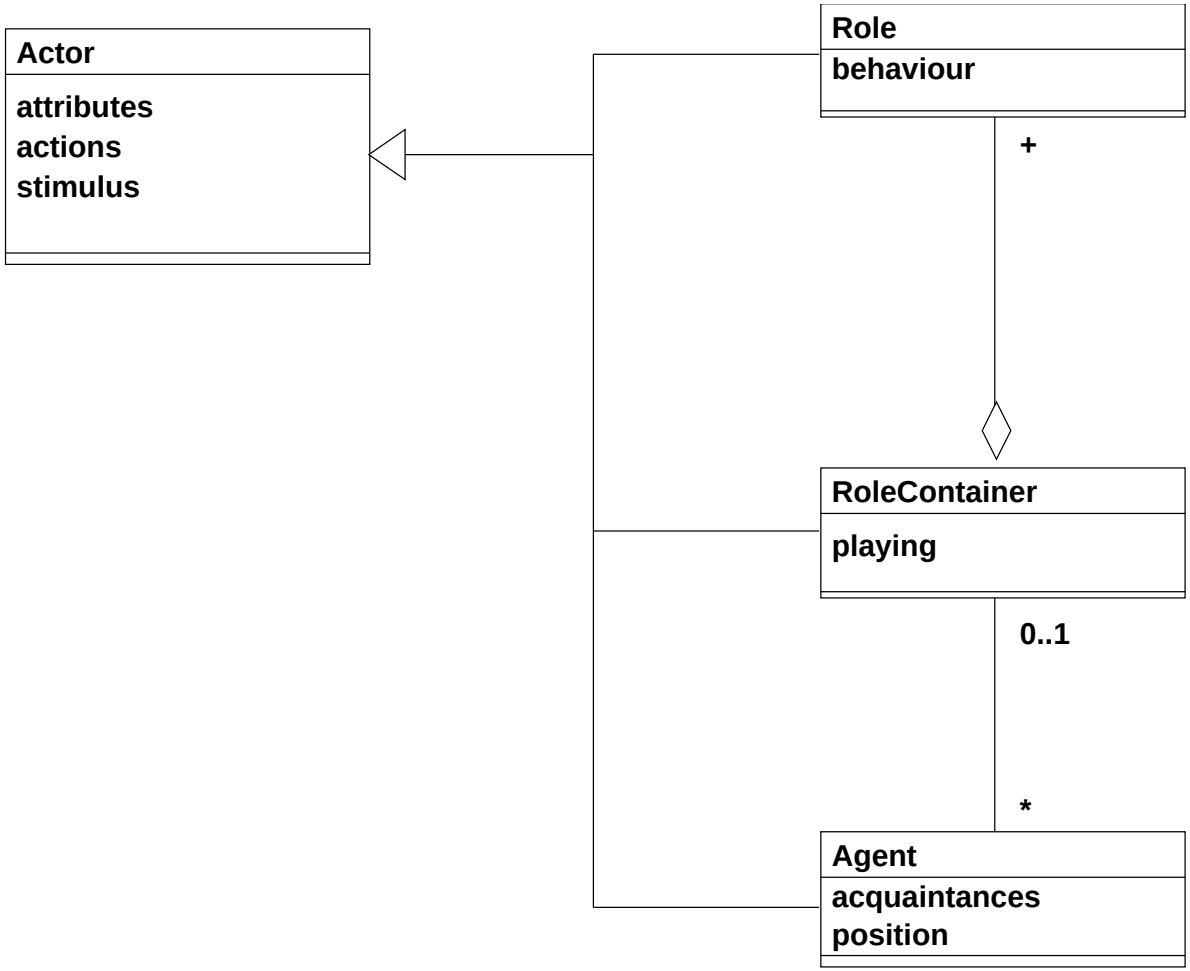
Definition

A role is an abstraction of a behaviour or a status in an organisation.

Definition

An organization is a set of roles such as every role interacts with at least one role in the organization.

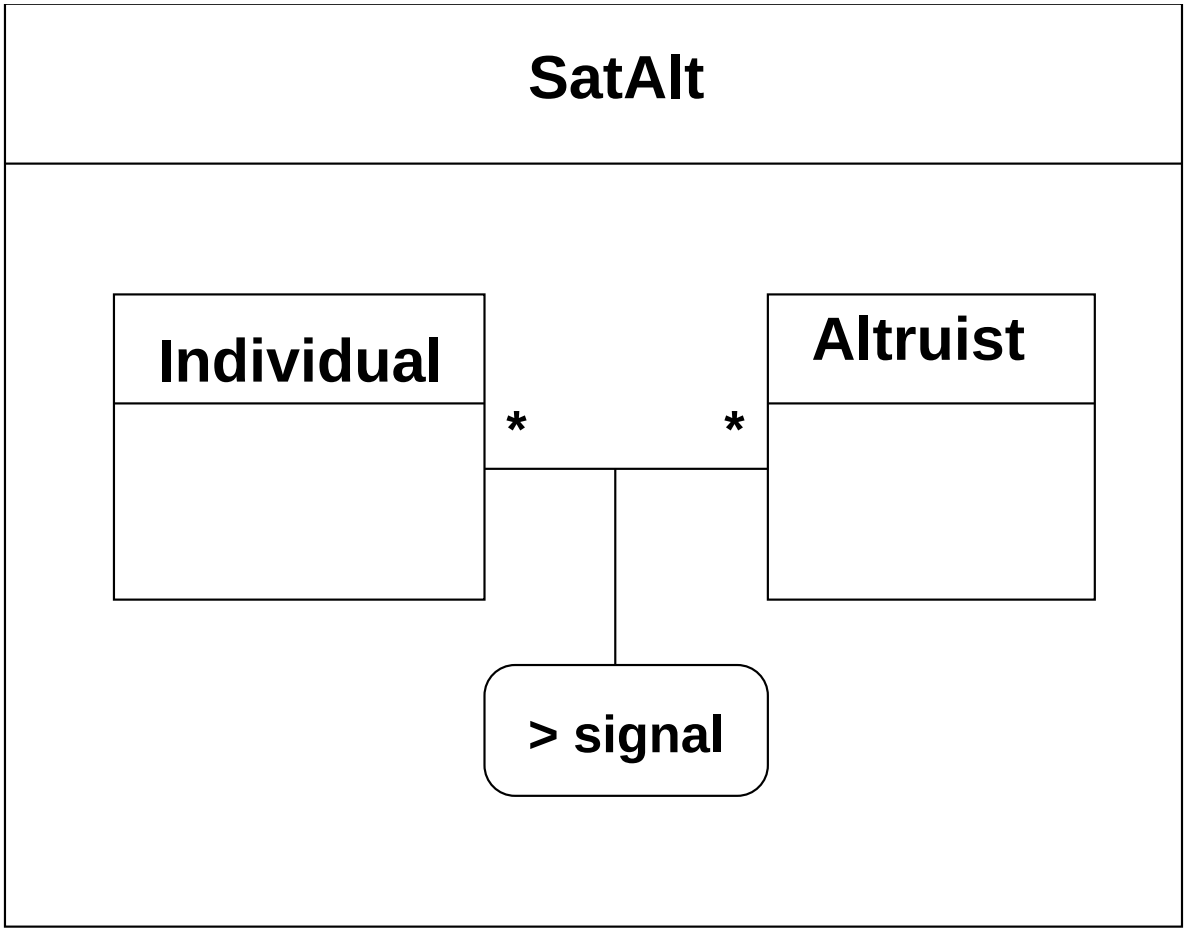
Meta-model overview



Organizational meta-model

- The meta-model uses concepts such as organization and role as first class citizen.
- Each concept (role, interaction, organization) is specified by a class with he OZS notation.
- Specification consists in specialization of framework classes.

Organization example



Outline

Introduction

The RIO meta-model

The OZS notation

Tools : prototyping and verification

Conclusion

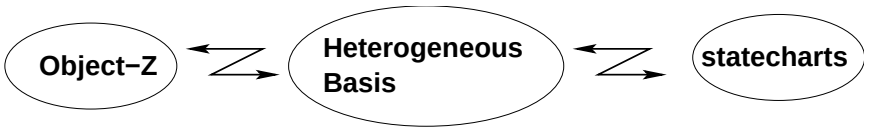
Multiformalisms notation

We have chosen two formalisms : Object-Z [Duke et al., 1991] and statecharts [Harel, 1987].

- Great expressive power.
- Specification simplified.
- Operational semantics.

Integration principles

[Gruer et al., 2004]

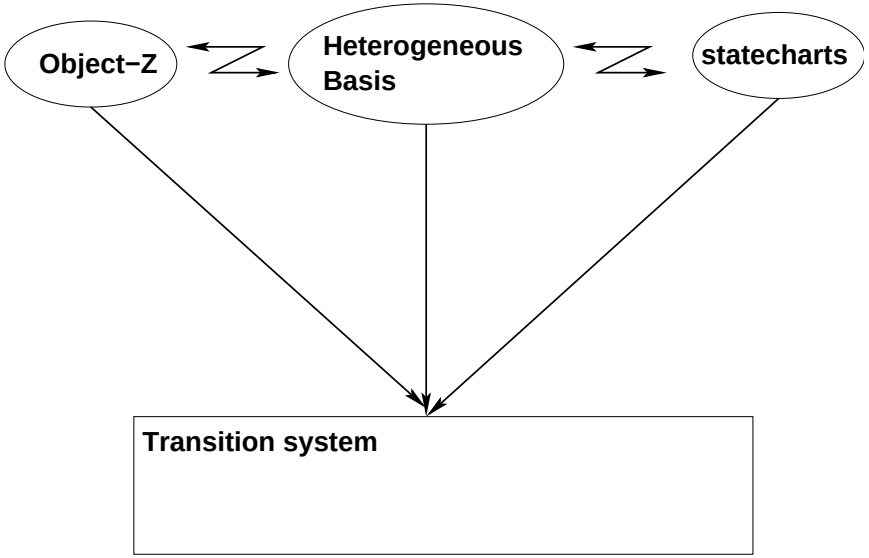


Syntaxic integration

Object-Z classes integrates a statechart which specifies their behaviours

Integration principles

[Gruer et al., 2004]



Semantic integration

Object-Z classes with a state-chart are given an operational semantics with transition systems.

Individual

Role

$current : Action$

$initialWeight, weight : Action \rightarrow [0, 1]$

$progressionReward : Action \rightarrow BMValue$

$s : DiscreteSensor$

$satisfaction, I : [-P_{max}, P_{max}]$

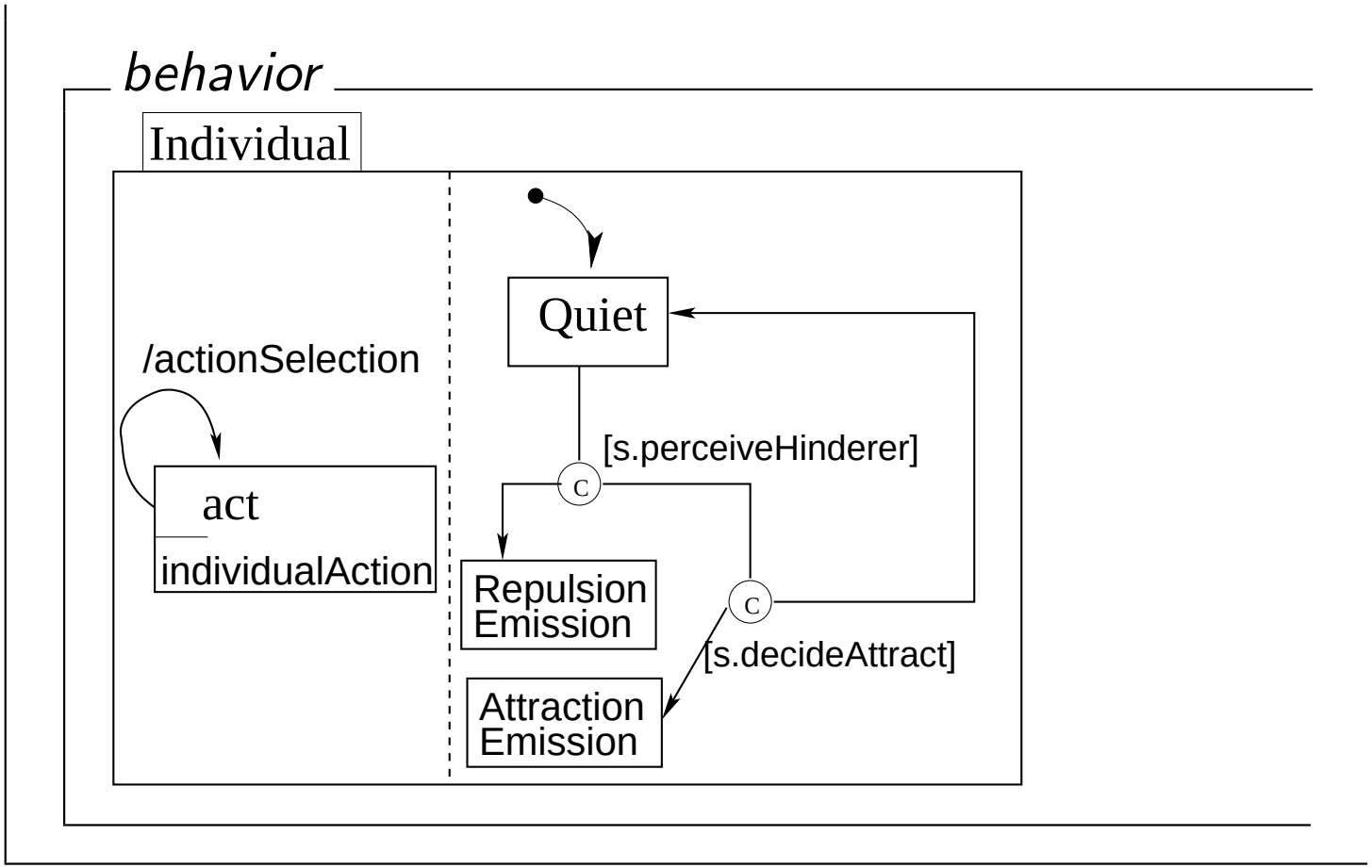
$\alpha : [0, 1]$

$current \subseteq actions$

$obtainCondition = \{|I_{ext}()| \leq P() \wedge |I_{ext}()| \leq I\}$

$leaveCondition = \{|I_{ext}()| \geq P() \wedge |I_{ext}()| \geq I\}$

$$\frac{ext! : \mathbb{R}}{ext! = s.getMax()}\ I_{ext}$$


behavior
Individual
Quiet
/actionSelection
[s.perceiveHinderer]
C
act
individualAction
Repulsion
Emission
C
[s.decideAttract]
Attraction
Emission

Altruist

Role

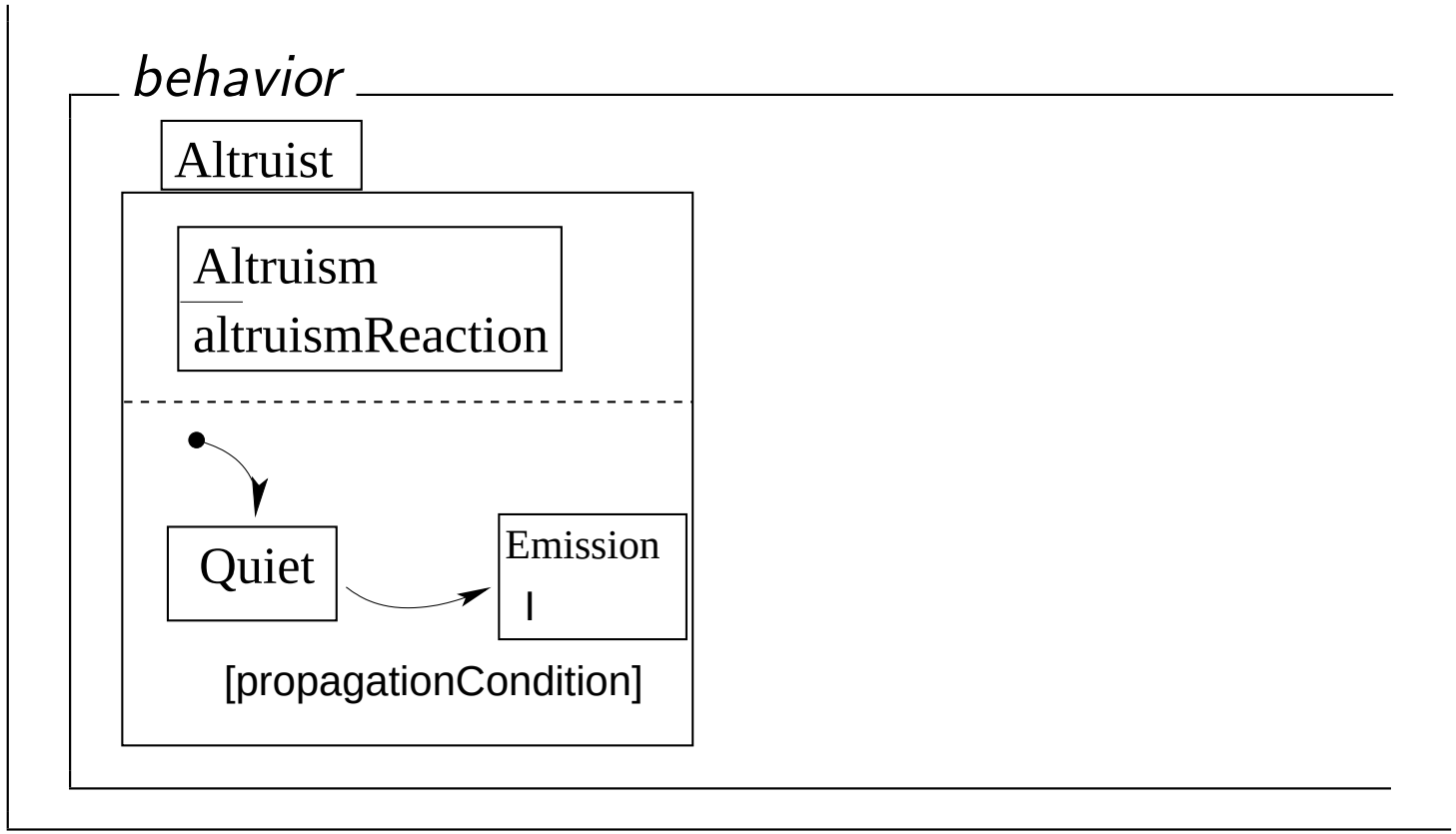
$$progressionReward : Action \rightarrow BMValue$$
$$s : DiscreteSensor$$
$$satisfaction, I : [-P_{max}, P_{max}]$$

$$current \subseteq actions$$
$$obtainCondition = \{|I_{ext}()| \geq P() \wedge |I_{ext}()| \geq I\}$$
$$leaveCondition = \{|I_{ext}()| \leq P() \wedge |I_{ext}()| \leq I\}$$

I_{ext}

$$ext! : \mathbb{R}$$

$$ext! = s.getMax()$$


behavior
Altruist
Altruism
altruismReaction
Quiet
Emission
I
[propagationCondition]

Outline

Introduction

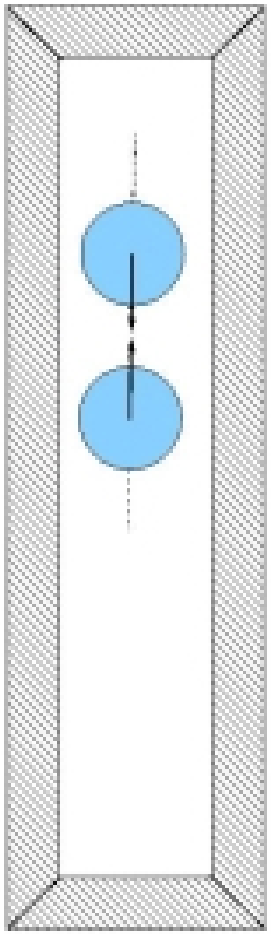
The RIO meta-model

The OZS notation

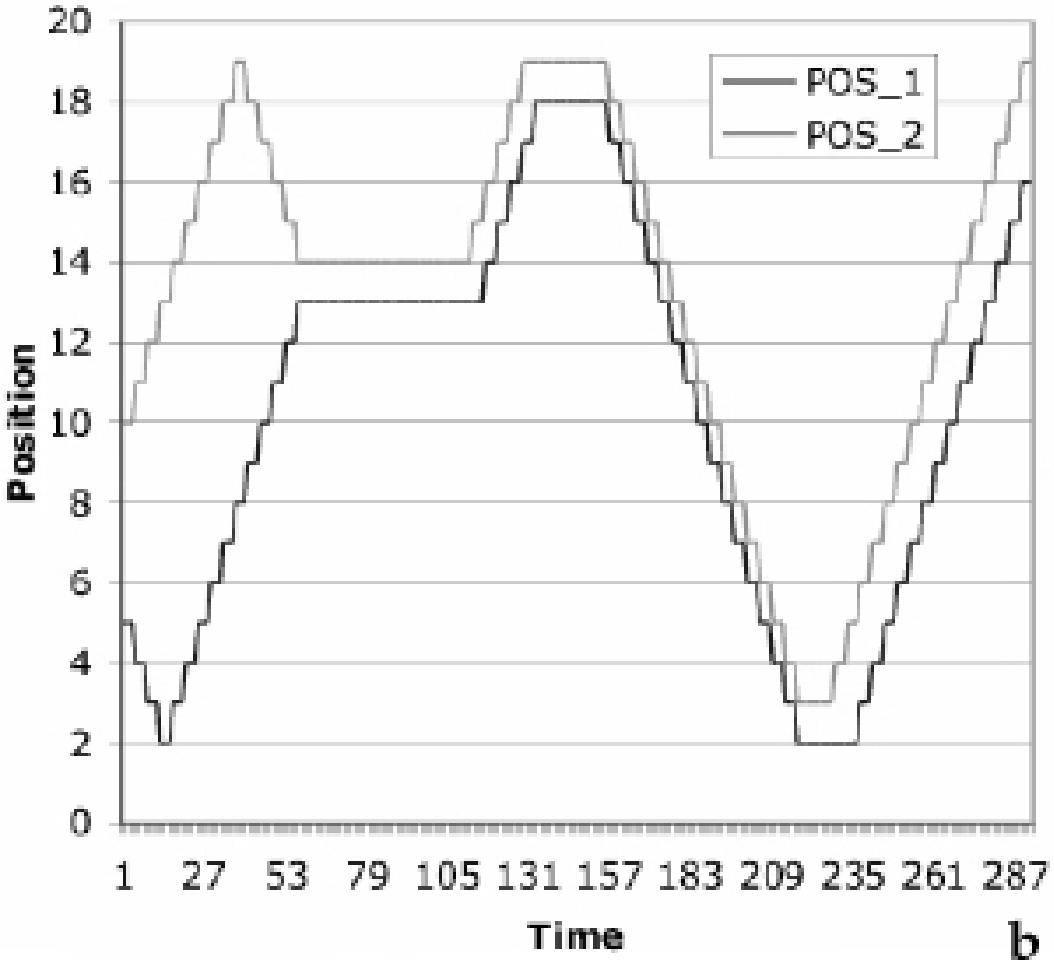
Tools : prototyping and verification

Conclusion

Prototyping



a



b

Verification

- Model-checking
- Verification

Outline

Introduction

The RIO meta-model

The OZS notation

Tools : prototyping and verification

Conclusion

Conclusion

- Formal specification linked to implementation.
- Modularity and abstraction,
- Simulation and animation.
- Proofs.

Perspectives

- CASE Tool currently in development.
- Compositional verification techniques.
- Refinement process.
- Integration with existing semi-formal notations.


Bergenti, F. and Poggi, A. (2000).
Exploiting uml in the design of multi-agent systems.
In Omicini, A., Tolksdorf, R., and Zambonelli, F., editors, *Engineering Societies in the Agents' World*, Lecture Notes in Artificial Intelligence. Springer Verlag.


Bernon, C., Gleizes, M.-P., Picard, G., and Glize, P. (2002).
The adelfe methodology for an intranet system design.


Brazier, F., Keplicz, B. D., Jennings, N., and Treur, J. (1997).
Desire: Modelling multi-agent systems in a compositional formal framework.
International Journal of Cooperative Information Systems, 6:67–94.


Burrafato, P. and Cossentino, M. (2002).

Designing a multi-agent solution for a bookstore with the passi methodology.


DeLoach, S. (1999).
Multiagent systems engineering: a methodology and language for designing agent systems.
In *Agent Oriented Information Systems '99*.


Duke, R., King, P., Rose, G., and Smith, G. (1991).
The Object-Z specification language.
Technical report, Software Verification Research Center, Departement of Computer Science, University of Queensland, AUSTRALIA.


Ferber, J. and Gutknecht, O. (1998).
A meta-model for the analysis and design of organizations in multi-agent systems.

In Demazeau, Y., Durfee, E., and Jennings, N., editors, *ICMAS'98*.


Gruer, P., Hilaire, V., and Koukam, A. (2000).
an Approach to the Verification of Multi-Agent Systems.
In *International Conference on Multi Agent Systems*. IEEE Computer Society Press.


Gruer, P., Hilaire, V., and Koukam, A. (2004).
Heterogeneous formal specification based on object-z and state charts: semantics and verification.
Journal of Systems and Software, 70(1-2):95–105.


Harel, D. (1987).
Statecharts: A visual formalism for complex systems.
Science of Computer Programming, 8(3):231–274.


Herlea, D. E., Jonker, C. M., Treur, J., and Wijngaards, N. J. E. (1999).
Specification of behavioural requirements within compositional multi-agent system design.
Lecture Notes in Computer Science, 1647:8–27.


Hilaire, V., Koukam, A., Gruer, P., and Müller, J.-P. (2000a).
Formal specification and prototyping of multi-agent systems.
In Omicini, A., Tolksdorf, R., and Zambonelli, F., editors, *Engineering Societies in the Agents' World*, number 1972 in Lecture Notes in Artificial Intelligence. Springer Verlag.


Hilaire, V., Koukam, A., Pablo, and Müller, G. J.-P. (2000b).
Formal Specification and Prototyping of Multi-Agent Systems.
Technical report, UTBM.
submitted to Agent Oriented Software Engineering'2000.


Luck, M. and D'Inverno, M. (1995).
Structuring a Z specification to provide a formal framework for autonomous agent systems.
Lecture Notes in Computer Science, 967:47–??


MESSAGE (2000).
Message website.


Wooldridge, M., Jennings, N. R., and Kinny, D. (1999).
A methodology for agent-oriented analysis and design.
In *Proceedings of the Third International Conference on Autonomous Agents (Agents'99)*, pages 69–76, Seattle, WA, USA. ACM Press.