



ForMAAD: Towards A Model Driven Approach For Agent Based Application Design

Zeineb Graja, Amira Regayeg
and Ahmed Hadj Kacem

ReDCAD Laboratory

<http://www.redcad.org>

University of Sfax - Tunisia

Context

- Agent technology has emerged as a strong solution for large-scale industrial and commercial applications
 - ⇒ a need to ensure that the systems being developed are robust, reliable and fit for purpose
 - ⇒ use of formal methods
 - ⇒ DESIRE
 - ⇒ OPERAS
 - ⇒ ForMAAD

Context

- ForMAAD: **F**ormal **M**ethod for **A**gent based **A**pplication **D**esign
 - uses the TemporalZ formal language
 - integrates linear temporal logic into the Z notation
 - based on a verified refinement conception process
 - composed of a specification phase and a design phase
 - offers principals and hints that guide the designer during conception

■ Motivations

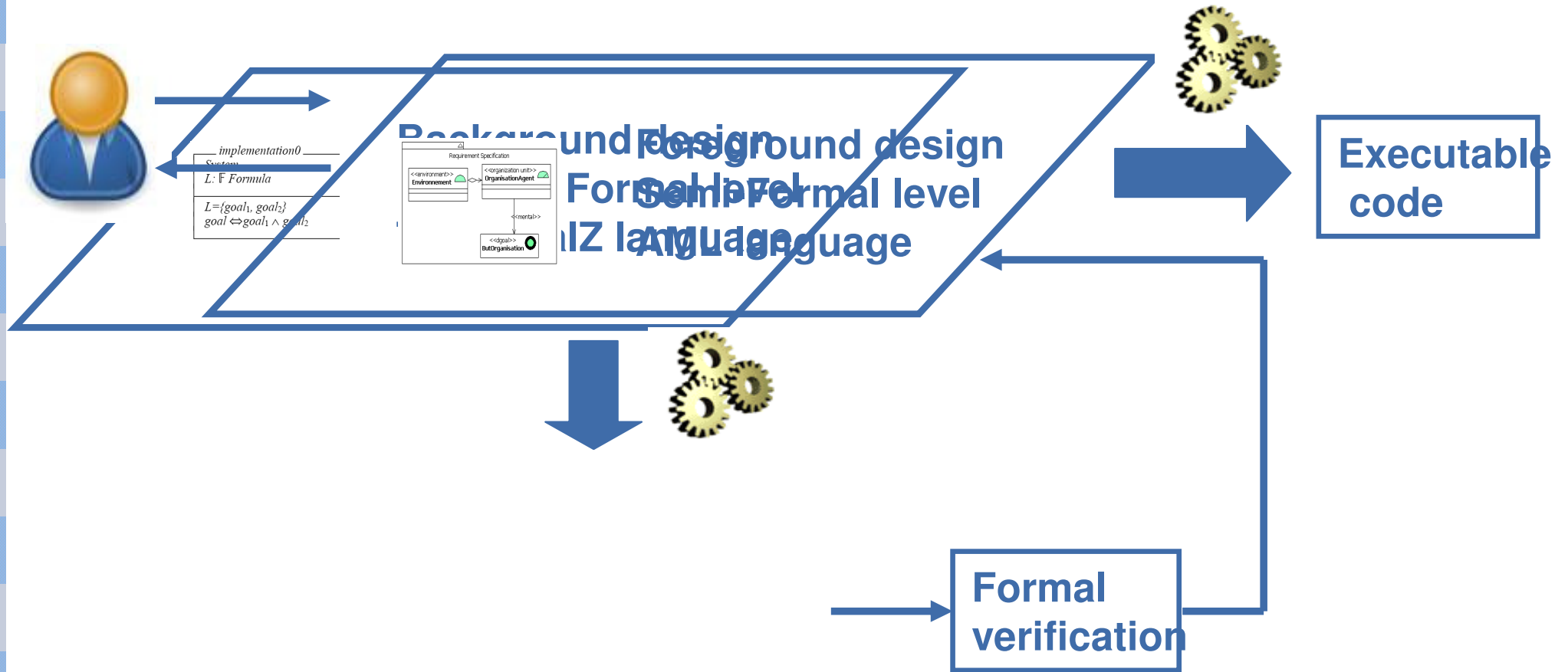
- The ForMAAD' usage is restricted to experts of formal languages
- The design result is at a very high level of abstraction
 - it is hard to generate an executable code
 - a deep gap between design and implementation

■ Motivations

- MDA (Model Driven Architecture) recommends
 - the use of semi-formal language (UML)
 - the use of different model for different level of abstraction
 - Computation Independent Model
 - Platform Independent Model
 - Platform Specific Model
 - the definition of transformation between models to

The idea is to reformulate the ForMAAD methodology in terms of the MDA paradigm

Objectives



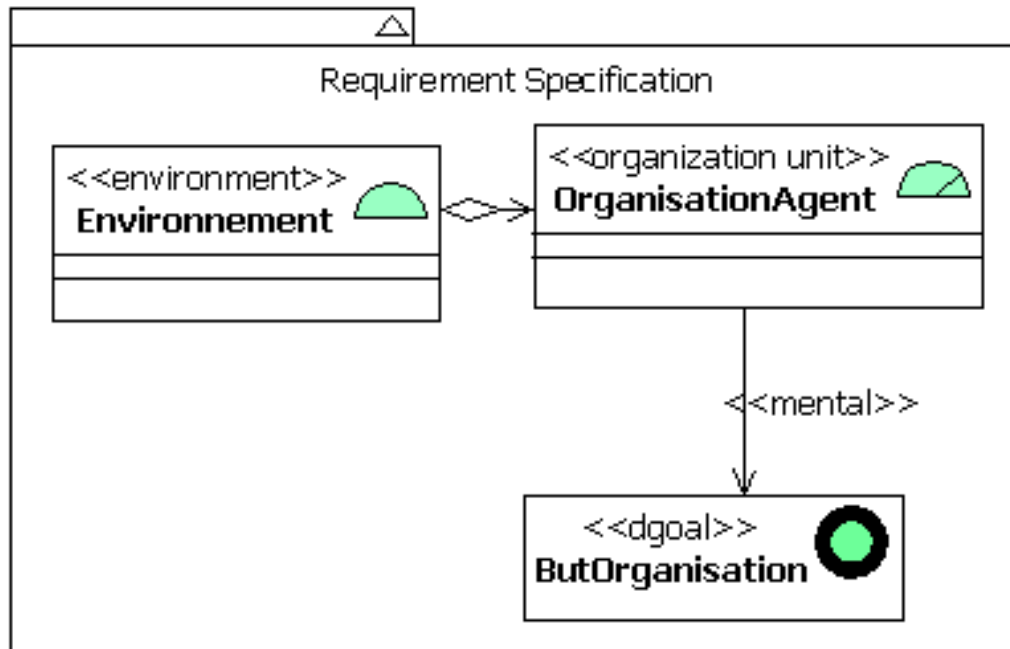


■ Outline

- Description of the foreground design
- Transformation rules
- Conclusion

Description of the foreground design

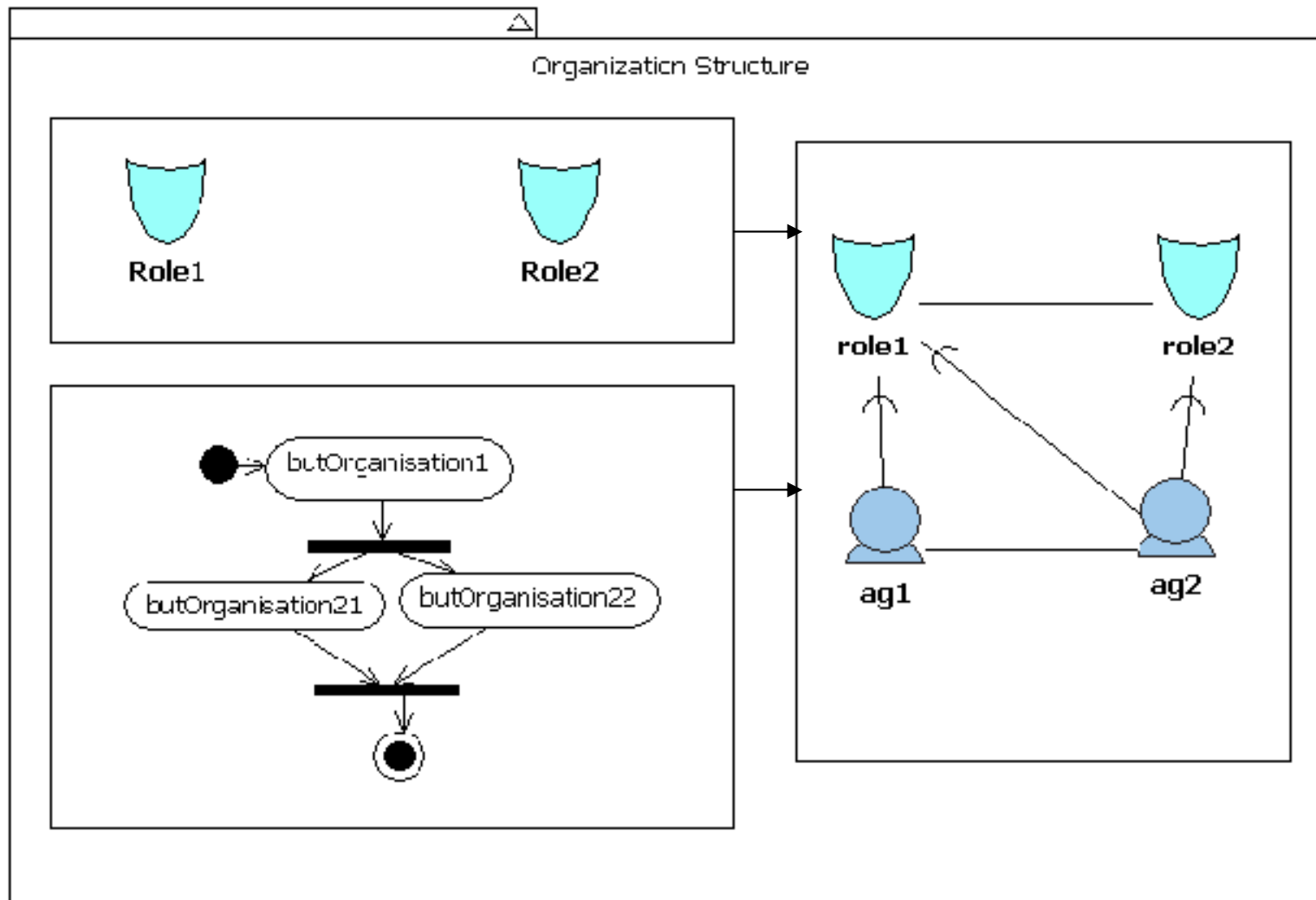
(1)



Requirement
specification

Requirement specification
model

Description of the foreground design (2)

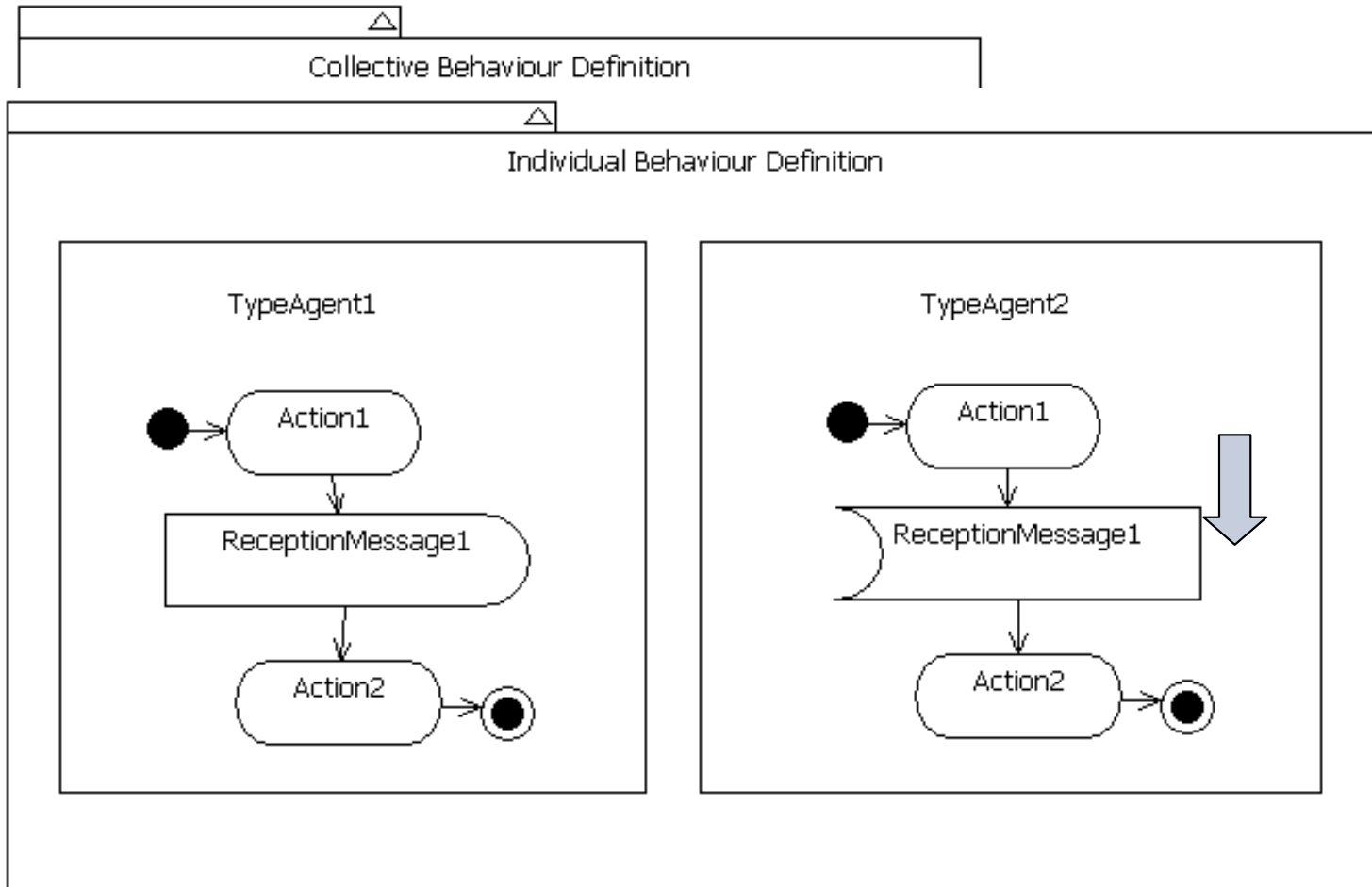


Cooperation strategy
definition model

Organization structure
definition model

Description of the foreground design

(3)

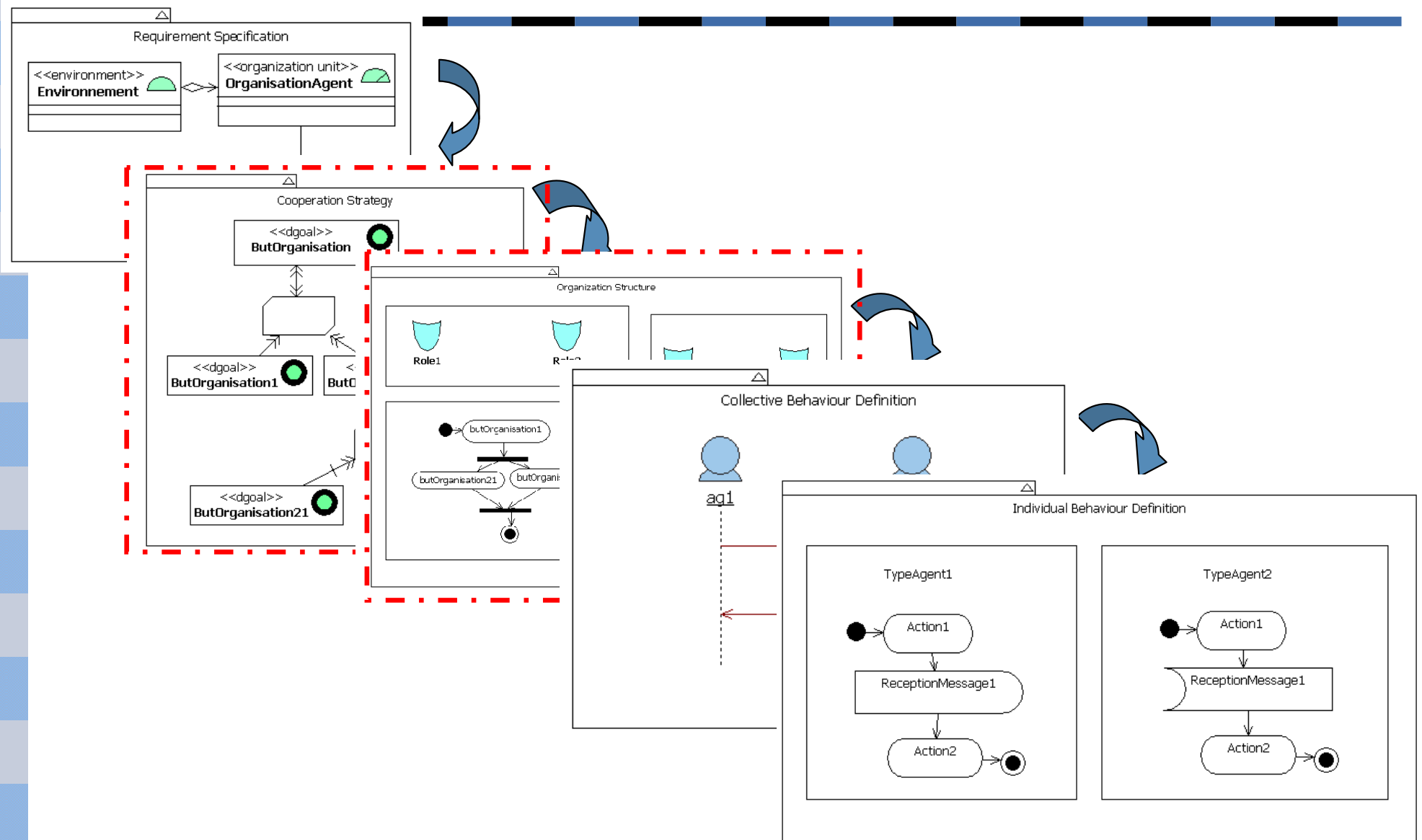


**Collective behavior
definition model**

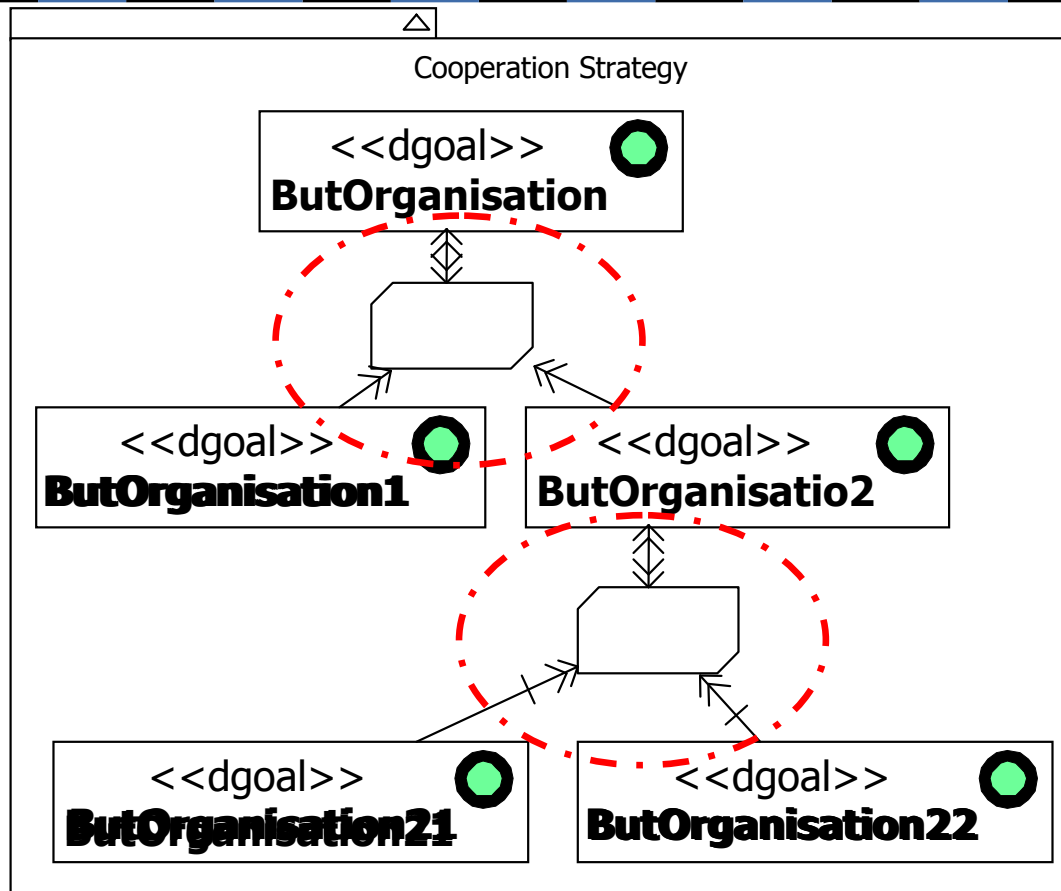
**Individual behavior
Definition model**

Description of the foreground design

(4)



Transformation rules (1)



Formula ::= ... |

implementation0 _____

System

L: F Formula

...

L = {

,

,

}

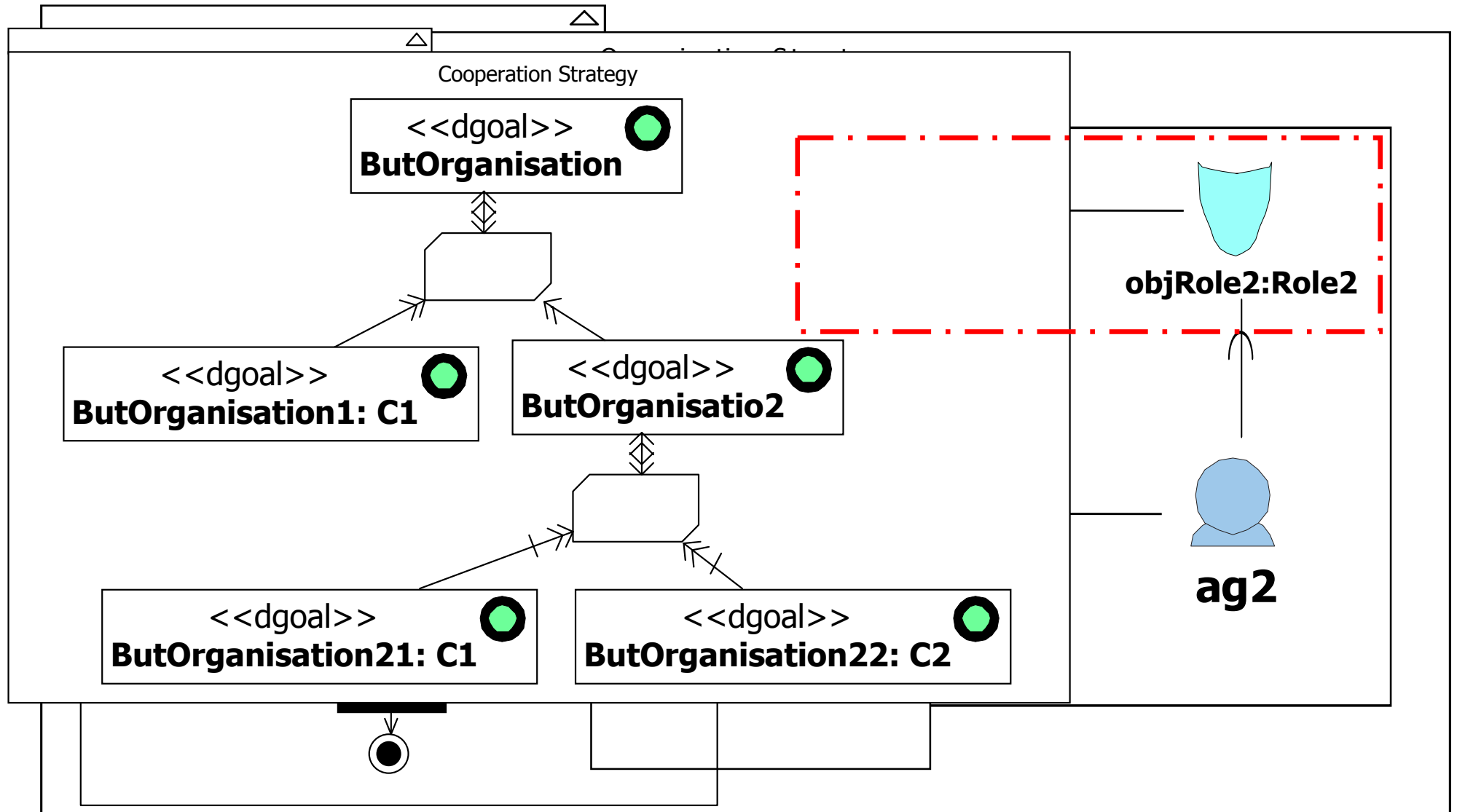
ButOrganisation \Leftrightarrow *ButOrganisation1* \vee *ButOrganisation2*

ButOrganisation2 \Leftrightarrow *ButOrganisation21* \wedge *ButOrganisation22*

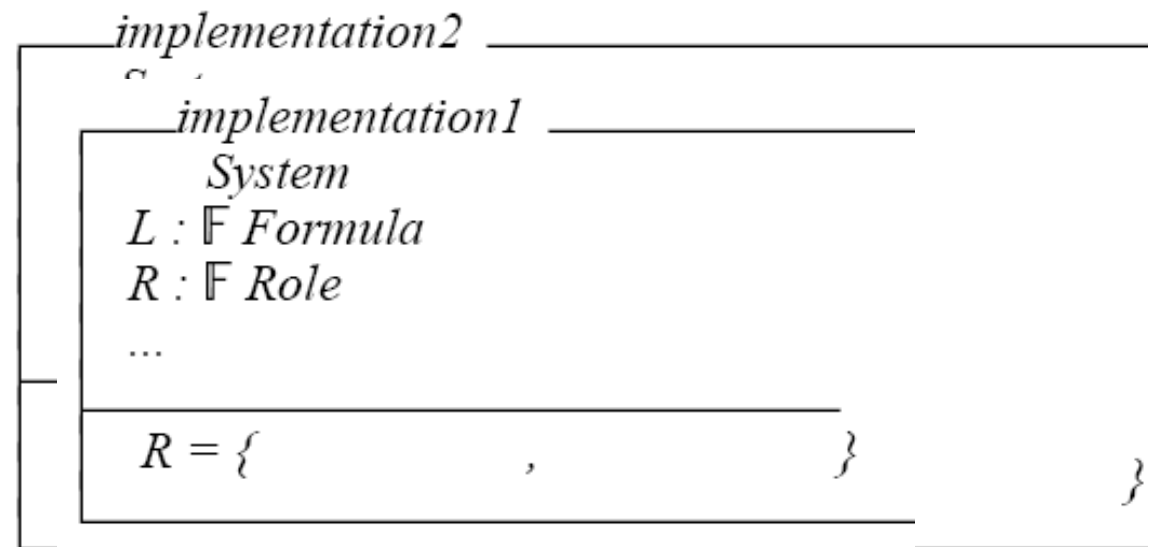
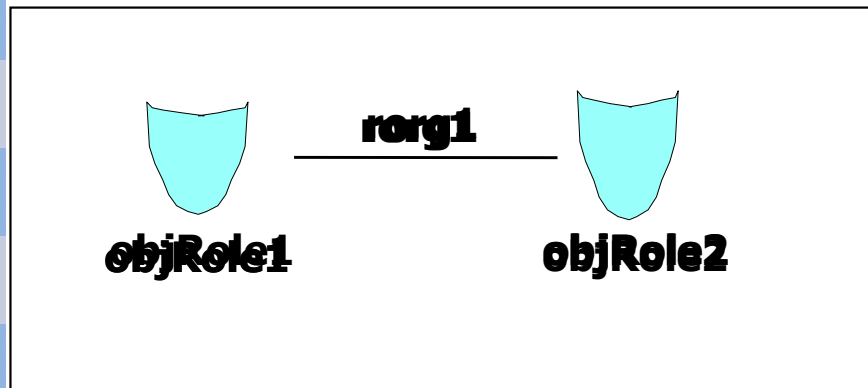
theorem CoopStrategy

$c_1, c_2, \dots, c_m \vdash \mathcal{C}$

Transformation rules (2)

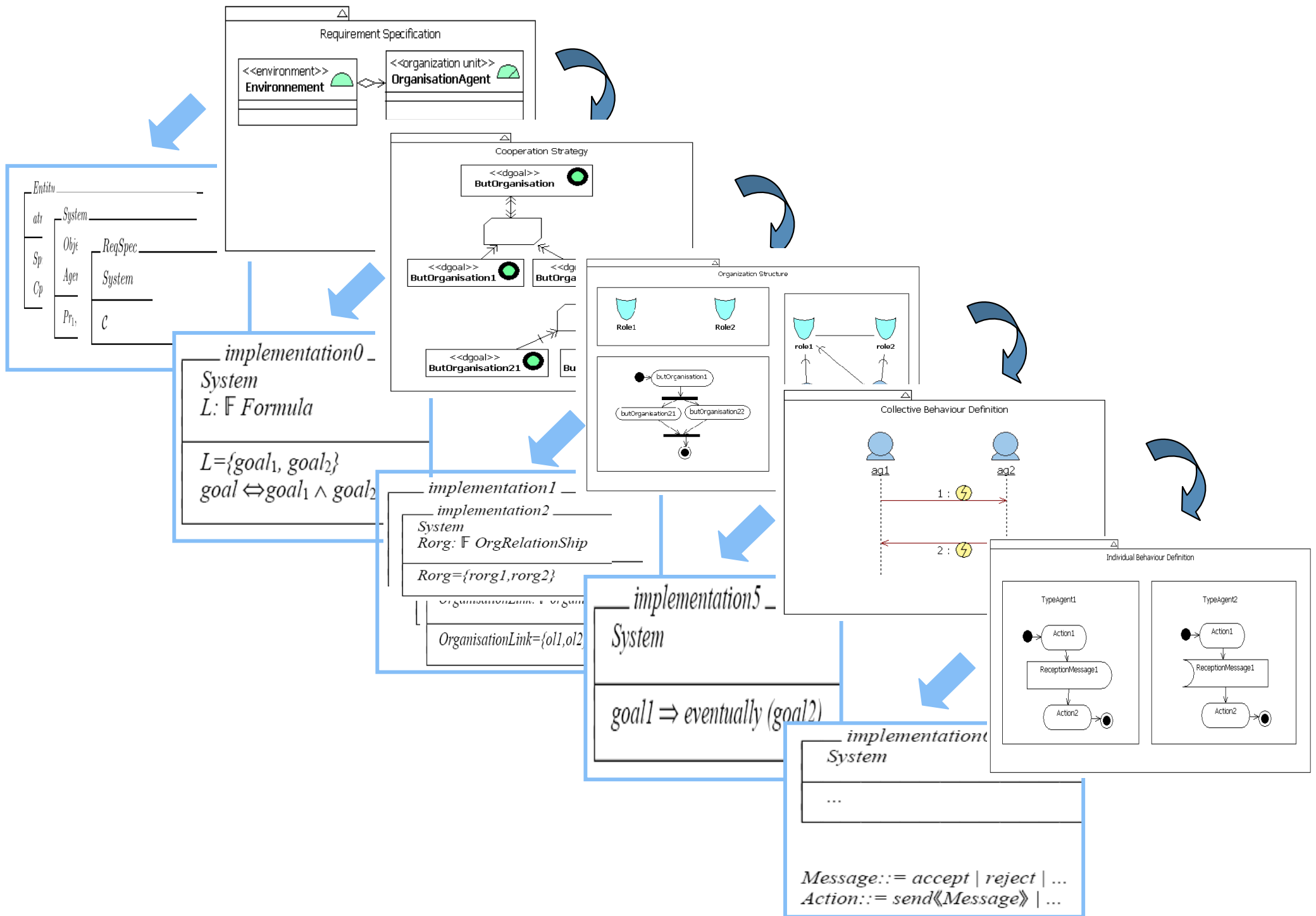


Transformation rules (3)



theorem RoleParticipant

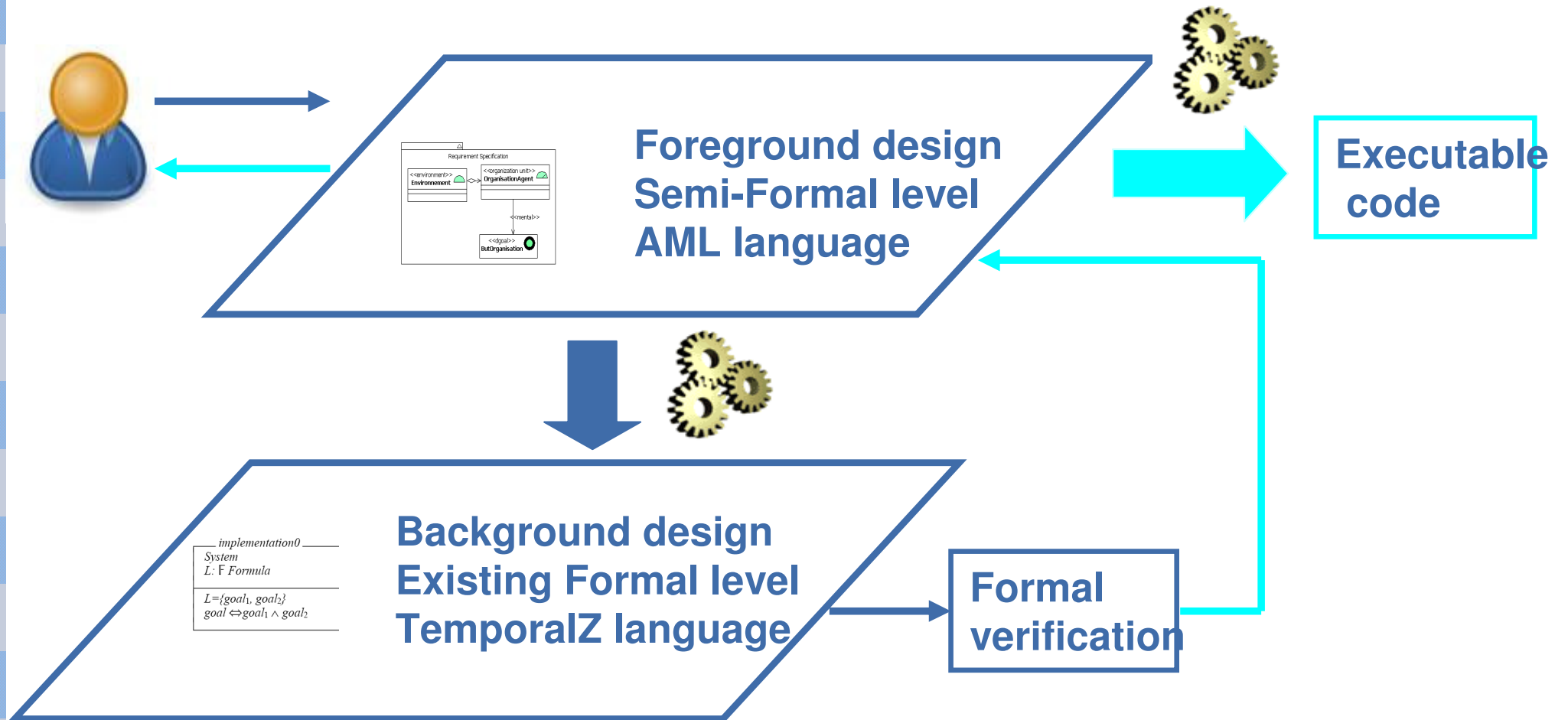
$$\text{Implementation}_2 \Rightarrow \bigcup_{\text{org} \in R_{\text{org}}} \text{org.participants} = R$$



■ Conclusion (1)

- Code generation using MDA techniques
 - implementation platform choice
 - transformation rules definition
- Transformation rules definition from the background design to the foreground one

Conclusion (2)



Thank you for your
attention