

A Formal Development and Validation Methodology applied to Agent-Based Systems

Giovanna Di Marzo Serugendo

CERN/IT Division

CH-1211 Geneva 23, Switzerland

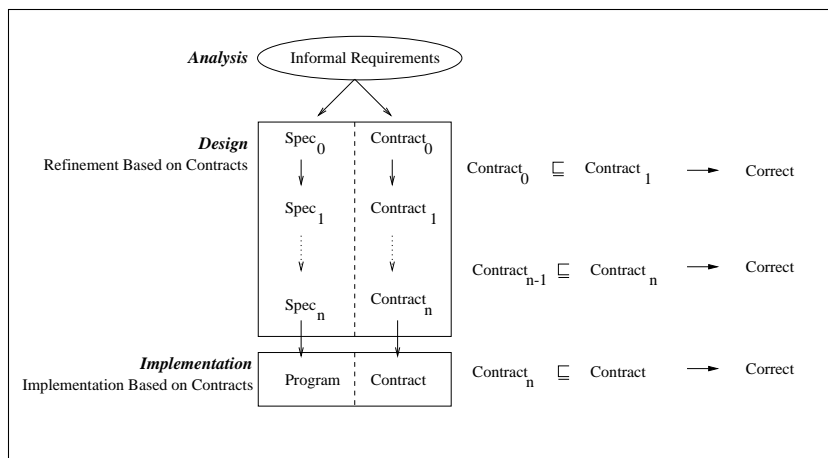
Outline

- Methodology: Development and Validation
- Application to Agent-Based Systems
- Example: Agent market place
- Refinement Patterns

Methodology (1)

- Stepwise Refinement + Formal Validation of Model-Oriented Formal Specifications
 - System Model (Behaviour)
 - ⇒ Model-oriented specifications language
 - Contract (Properties to preserve)
 - ⇒ Logical Language
- Stepwise Refinement of CO-OPN/2 Specifications
 - CO-OPN/2
 - Hennessy-Milner Logic

Methodology (2)



Agent-Based Systems

- System performs *functionality* for final user
- System = *Collection* of Agents + *relationships* among collections (negotiation techniques, cooperation protocols, coordination models)
- Agents *interact* with each other (message passing, blackboard, etc.)
- Agent = *problem-solving* entity (own algorithm)

Development Guidelines

- *Informal Requirements*
- *System's Functionality* (Contract: Functionality)
- *Agent Collections* (Contract: Functionality of Collection + Properties of interactions)
- *Collections Design* (Contract: Functionality of Agents)
- *Agent Design* (Contract: Internal Behaviour of Agents)
- *Actual Communications Means* (Contract: Properties of Communication)
- *Program* (Contract: same as last specification)

Market Place: Contract

$$\phi_{I_1} = \langle MP.create \rangle \langle MP.new_buyer(b) \rangle \langle MP.new_seller(s) \rangle$$

$$\langle MP.buy(b, g, p, h) \rangle \langle MP.sell(s, g, p', l) \rangle$$

$$\langle MP.sold_goods(s, b, g, p1) \rangle$$

$$\phi_{I_2} = \langle MP.create \rangle \langle MP.new_buyer(b) \rangle \langle MP.new_seller(s) \rangle$$

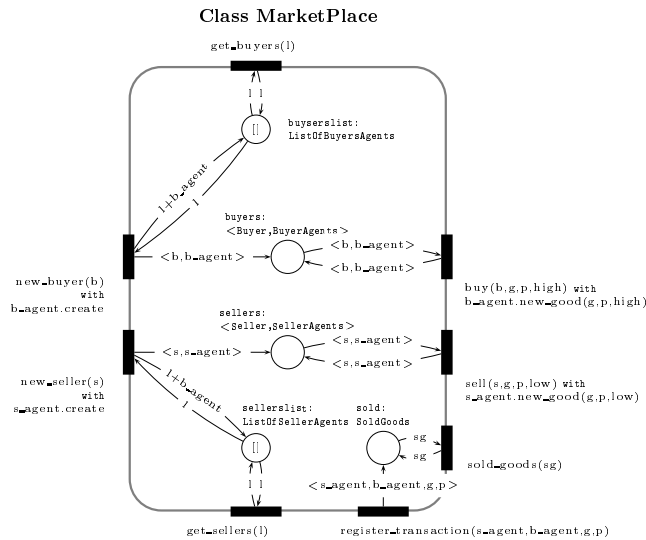
$$\langle MP.buy(b, g, p, h) \rangle \langle MP.sell(s, g, p', l) \rangle$$

$$(\neg \langle MP.sold_goods(s, b, g, p2) \rangle \wedge$$

$$\neg \langle MP.sold_goods(s, b, g, p3) \rangle)$$

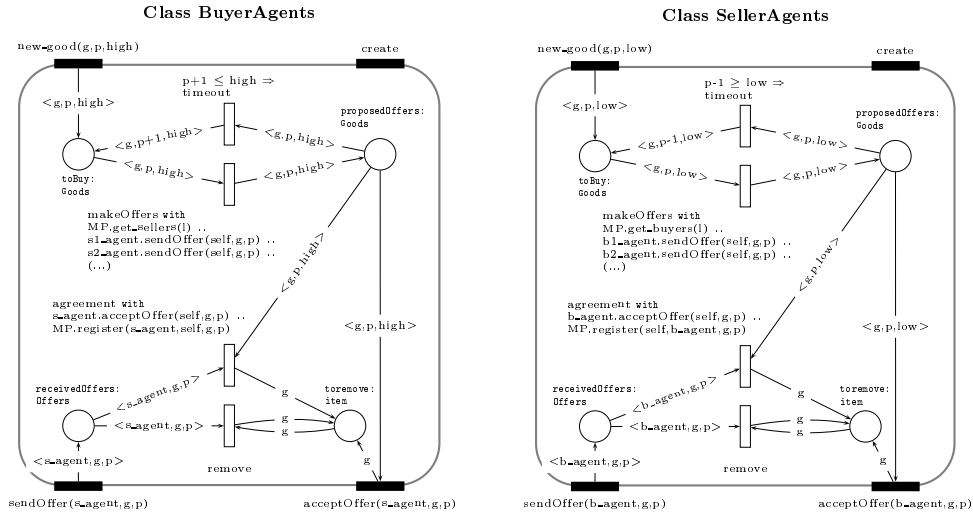
$$l \leq p_1 \leq h, p_2 \leq l, p_3 \geq h$$

Market Place: Refinement R1 (1)



Agent decomposition and Interactions

Market Place: Refinement R1 (2)



Refinement R1: Contract

$$\phi_{R1_1} = \phi_{I_1}$$

$$\phi_{R1_2} = \phi_{I_2}$$

$$\begin{aligned} \phi_{R1_3} = & \langle MP.create \rangle \langle s_agent.create \rangle \langle b1_agent.create \rangle \langle b2_agent.create \rangle \\ & \langle s_agent.sendOffer(b1_agent, g, p) \rangle \langle s_agent.sendOffer(b2_agent, g, p) \rangle \\ & \langle b1_agent.sendOffer(s_agent, g, p) \rangle \langle b2_agent.sendOffer(s_agent, g, p) \rangle \\ & ((\langle b1_agent.acceptOffer(s_agent, g, p) \rangle + \langle b2_agent.acceptOffer(s_agent, g, p) \rangle) \wedge \\ & \neg (\langle b1_agent.acceptOffer(s_agent, g, p) \rangle \langle b2_agent.acceptOffer(s_agent, g, p) \rangle) \wedge \\ & \neg (\langle b1_agent.acceptOffer(s_agent, g, p) \rangle // \langle b2_agent.acceptOffer(s_agent, g, p) \rangle)) \end{aligned}$$

Market Place: Refinement R2 + Program

Refinement R2: Actual Communications Means

- Market place = server + RMI class
- Agents = RMI Objects
- RMI registry (RMI references available)
- Contract: RMI features

Program

- Java Implementation of Refinement R2
- Contract: same as Refinement R2

Refinement Patterns

Client/Server

- Requirements
- Centralised
- Distributed Data
- C/S architecture
- Communication Layer
- Program

Dependable Applications

- Requirements
- No Fault-Tolerance
- Fault-Tolerant Design
- Program

Conclusion

- *General Theory of Refinement*
Refinement Formalisation, Formal Validation
- *Refinement Patterns*
(MAS, C/S, Dependable Applications)
- *Future*
Temporal Logic for CO-OPN/2
Refinement of Real-time specifications