

Modeling Notation Source

Styx

Version: 03-03-10

Document author: Radovan Cervenka

1 Introduction

Styx [Bush01] is an agent-oriented methodology, which guides the development of collaborative agent systems. The key areas covered by this methodology are the specification of communication concepts, inter-agent communication and each agent's behavior activation - but it does not address the development of application-specific parts of a system.

2 Notation Overview

2.1 Use Case Map

Styx exploits *Use Case Maps* (UCM) defined in [Buhr96]. UCM models possible processes in a system as paths which traverse various components of the system. Components are drawn as boxes, while paths are drawn as lines crossing various components. The start of a path is indicated by a solid circle, while the end point is indicated by a strong line. When a path crosses a component that component is assigned one or more responsibilities associated with the path. The UCM is labeled with component names, responsibility names and other explanatory notes. These maps provide a highly condensed notion suitable for modeling the high-level behavior of a system. A sample UCM diagram is depicted in Figure 1 a), and the responsibilities in this figure are expanded in Figure 1 b).

Rather than specification of individual agents, *Styx* specifies roles for the UCM component names.

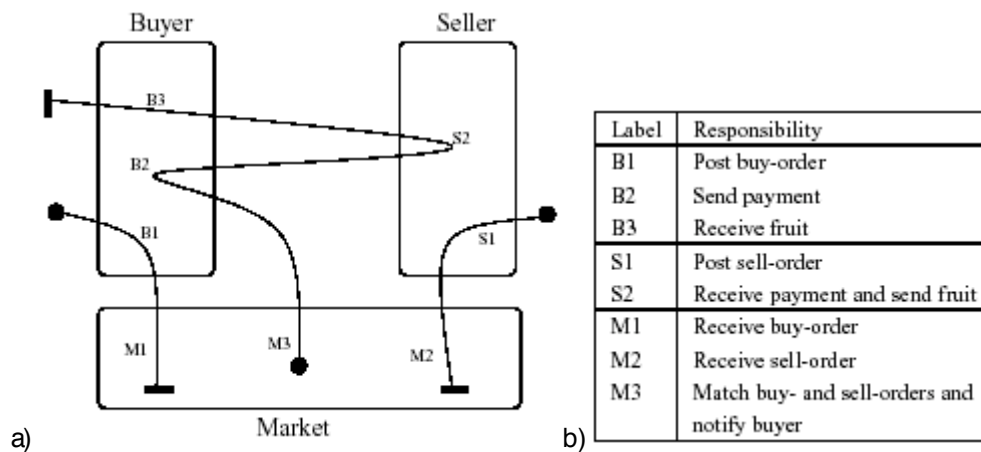


Figure 1: Use case map and corresponding responsibility table

2.2 Domain Concept Model

The *Domain Concepts Model* models each concept that agents will communicate about. Each concept is modeled as a particular object class. *Styx* uses UML class diagrams for modeling of domain concepts.

2.3 Role Responsibility Model

Role Responsibility Models are created for each component of the analysis level UCM, and take the form of a table with four columns: responsibility, pre-condition, post-condition, and action. For each responsibility in the analysis level UCM, an entry is made in the appropriate role's Role Responsibility Model.

2.4 Role Relationship Model

For each responsibility *The Role Relationship Model* further elaborates the relationships between roles. Each relationship is assigned a type of FIPA interaction protocol and an object from the Domain Concepts Model. The interpretation is that a conversation of the specified type will occur between the agents, where information is interchanged using the specified object.

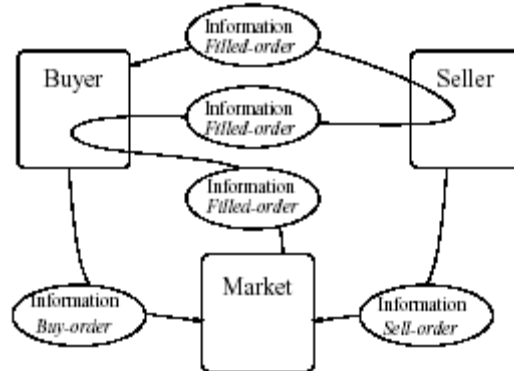


Figure 2: Example of role relationship model

2.5 Deployment Model

The *Deployment Model* is the most simple model in this methodology. It specifies a many-to-many mapping between agents and roles. This model specifies what agents will exist in the system, and what roles they will play. An example of the deployment model is given in Figure 5.



Figure 3: Example of deployment model

3 Unification Considerations

Problems:

- Styx does not consider systems that contain a large number of roles (recommended is less than 30).
- UCM for complicated scenarios (including many decisions, paths as well as actions) is not clear.
- Such notions as planning, scheduling, mobility, learning, etc. are not explicitly handled by Styx.
- Role relationship model describes interactions insufficiently, e.g. bounding of interaction protocols is not described well, specification of one object type for the whole interaction is insufficient, etc.
- Modeling techniques are UML incompliant, even if e.g. UML activity diagrams should be used instead of UCM, etc.

Recommendation:

Styx modeling language should represent an “inspiring work”, but I do not recommend direct adopting of Styx’s modeling constructs into FIPA AUML.

4 References

- [Bush01] G. Bush, S. Cranefield and M. Purvis: *The Styx Agent Methodology*. In The Information Science Discussion Paper Series, Number 2001/02, January 2001, ISSN 1172-6024
<http://citeseer.nj.nec.com/bush01styx.html>
- [Buhr96] R. J. A. Buhr and R. S. Casselman: *Use case maps for object-oriented systems*. Prentice Hall, 1996.
<http://citeseer.nj.nec.com/buhr96use.html>