Chapter XIV
From Business Process Model to Information Systems Model:
Integrating DEMO and UML

Peter Rittgen
University College of Borås, Sweden

ABSTRACT

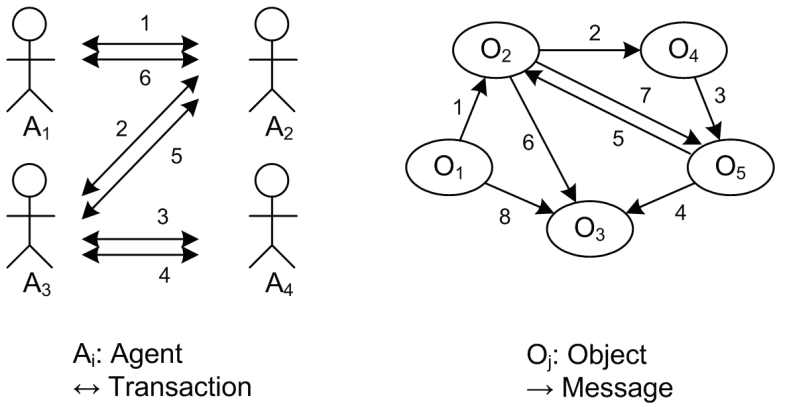
The main purpose of a corporate information system is the support of the company's business processes. The development of information systems is therefore typically preceded by an analysis of the business processes it is supposed to support. The tasks of analysing business processes and designing information systems are governed by two seemingly incompatible perspectives related to the interaction between human actors or inanimate agents (objects), respectively. As a consequence, the corresponding modeling languages also differ. DEMO (dynamic essential modeling of organization) is a typical language for modeling business processes, the UML is the predominant language for information systems modeling. We challenge the assumption of incompatibility of the perspectives by providing a framework for the integration of these languages.

INTRODUCTION

In the action view, a system consists of a number of agents (people or organizational units) who interact with each other by communicating. The basic unit of communication is a speech act (Austin, 1962; Searle, 1969). A transaction (Weigand & van den Heuvel, 1998) is the smallest sequence of actions that has an effect in the social world (e.g., establishing a commitment). It typically consists of two speech acts: an utterance and the response (e.g., a request and the promise). On the third level, the workflow loop (or action workflow, Medina-Mora, Winograd, Flores, & Flores, 1992) describes a communicative pattern consisting of two consecutive transactions that aim at reaching an agreement about (1) the execution of an action and (2) the result of that execution. The left side of Figure 1 shows three examples of workflow loops. Higher levels can be defined such as contract and scenario but the first three are sufficient for the purpose of this chapter. More details on the action view are given in the section "Dynamic Essential Modeling of Organization."

In the reaction view, object orientation prevails today. It has largely replaced the functional paradigm that characterized early approaches to software en-

Figure 1. Action view and reaction view



gineering (and is still used in certain areas such as databases). In object orientation, a system is seen as a collection of objects exchanging messages. Each object encapsulates data and functionality (or structure and behaviour, or attributes and operations). An object is in principal a passive (or reactive) unit that only acts if it receives a message. It will then carry out the appropriate operation which might involve sending messages to other objects. Finally, it will deliver the result as a reply to the original message but "communication" is essentially one-way (see Figure 1, right). More details on the reaction view can be found in the object-oriented literature, for example (Dori, 2002).

The major conceptual differences between the views are:

1. The action view describes social systems that consist of human beings that can both act of their own accord and react to stimuli from the environment, whereas an object can only react.
2. By performing speech acts, agents create obligations for themselves or others. Having a conscience, they are fully aware of the consequences of entering into a commitment and also of not fulfilling an obligation. An object is not equipped with a conscience so it cannot commit itself. If an object behaves in the "desired" way, this is due to a pre-programmed automatism and not the result of an individual decision based on free will. An object cannot be responsible for its "actions."
3. Communicating is not just exchanging messages. We communicate to achieve a certain purpose for which we need the help of others. An object sends a message because its code prescribes this behaviour and the message is received, processed, and "answered" for precisely the same reason. An object has no intentions.

BACKGROUND

Regarding the reaction view, the task of finding an appropriate language is not difficult. The software engineering community has subjected itself to a rigorous standardization process that resulted in the unified modeling language (UML). It follows the object-oriented paradigm and is widely used in the design of information systems. Adhering to the reaction view, its focus is more on the technical part of the information systems than on the organizational (i.e., social) part, but the proponents of UML claim that it can also be used for the latter. As evidence for this standpoint, they mention use cases and business processes. For the former, UML