Chapter 16

Representing and Reasoning with Scenarios within Information Systems Modeling

Choong-ho Yi
Swedish Defence Research Agency (FOI), Sweden

INTRODUCTION

Recent research in systems engineering shows the community’s interest in scenarios. Advantages of using scenarios have been reported in wide range of contexts within the area. Recalling that any systematic, and eventually, automated management of scenarios presupposes firm formal ground, research on formal theories for scenarios is highly motivated. However, only little work has been reported for that purpose, e.g., Hsia, Samuel, Gao, Kung, Toyoshina and Chen (1994). This chapter presents a formal approach to managing scenarios, using first order temporal logic extended with basic set theory, in the context of information systems modelling (ISM). In our framework scenarios are dealt with from two formal points of view, namely representation and reasoning.

Representation of scenarios: An information system (IS) is defined as a structure of object schema, actions, time and scenarios where each component of the structure is motivated to capture different aspects of the system, i.e., the static, the dynamic and the temporal aspect respectively, and scenarios are, roughly, “compounded” of the three components. Then formal semantics has been given to the components and other concepts associated with them, e.g., state and change. The in-built semantics of usual relationship diagrams, e.g., Entity-Relationship diagram (Chen, 1976), have been strengthened and formalised into basic predicates...
and functions which are used to express relationships between object types in logical formulae. The set of these formulae constitute an object schema. A state is defined on the basis of object schema, i.e., as an instance of object schema, and it describes the “snapshot” of the system at a moment of time. The static aspects, however, will not be presented in the paper for space reason.

We approach scenarios mainly in terms of actions, and so semantics of actions is of immediate importance in discussing scenarios. Actions describe the dynamic part of an IS such that system state is changed by the effects of actions. In order to represent action duration with varying length effectively we use time explicitly and associate each action occurrence with start and end time. Further each action is provided with, in addition to the objects that are involved in the action, two categories of agents participating in the action, i.e., the initiator and the receiver. The agents pair is used to represent communications between multiple agents, and to trace capabilities and responsibilities of the agents concerning the action.

A scenario is then defined as a pair of action occurrences set (describing what to do) and constraints set (how to do these actions). Some constraints are imposed on individual actions concerning, e.g., possible duration of the actions, or agents and other kinds of objects involved in the actions (e.g., when ordering a product, certain supplier is preferred to other suppliers and the price of the product should not be higher than certain level), and so on. At the same time constraints may be related to over several actions as well, e.g., performance order of the actions. Scenarios represented as such, using the whole expressive power of actions in combination with constraints, enjoy rich expressiveness and flexibility, and are suitable for, e.g., modelling various kinds of business processes or business policies (e.g., if a product is ordered by a special customer, then deliver a product within X time units after the order) in realistic environments.

Reasoning with scenarios: One of major benefits of formally representing scenarios is that it serves as base for formal reasoning with scenarios in various ways, e.g., prediction of future, simulation and evaluation of possible trajectories of the scenarios. Though several formal approaches have been proposed within ISM, they were mainly designed for representation and the connection to the phase of reasoning has not been studied enough. Consequently, reasoning is still, more or less, unfamiliar in that area. This chapter proposes “such a connection” and shows how reasoning, though straight, in the form of prediction for a given scenario, can be made in our framework based on the semantics introduced for the representation part.

**ACTIONS: THE DYNAMIC PART**

Actions describe the dynamic part of the system. Each action is provided with, in addition to the objects which are involved in the action, two categories of
Towards Understanding the Use of Patterns in Software Engineering
www.igi-global.com/chapter/towards-understanding-use-patterns-software/52880?camid=4v1a