Chapter XII

Models for Cooperative Activities over the Web

Bogdan D. Czejdo, Maciej Zakrzewicz and Govindarao Sathyamoorthi
Loyola University, New Orleans, USA

ABSTRACT

The Chapter discusses the need and the problems associated with WEB based cooperative activities in which several team members work in parallel on a common task. Models for software systems supporting such cooperative activities are discussed. Our models describe structure of the cooperation object, cooperation modes and the network message synchronization, that are of prime importance when the system members work at different places and communicate over the Internet. We introduce and describe a component requirements graph and show how to translate it into an interaction graph. The state diagrams and the design graphs are the basis for the WEB software design. The discussion of software architecture for implementing cooperative activities over the Web is also provided.

INTRODUCTION

There is a well-recognized need to provide support on the WEB for cooperative activities in which several team members work in parallel on a common task. The common task usually involves a common object that we will refer to as a cooperation object. The cooperation object can have a complex structure. The members of the team can work on separate components of the cooperation object but the components are related to each other. The objective of this chapter is to describe various models of software supporting such activities, allowing not only
systematic creation of the software but, with further advances in those efforts, also automatic generations of the software.

The cooperation over the Internet is an area of very active research, and special workshops are organized to present the newest results and implementations (Georgia Bafoutsou, Gregory Mentzas, 2001). Since there are many distinct application areas of WEB based cooperation, some significantly different solutions are proposed for different application areas. The closest to our work is published by (Cornelia Haber, 2001). This work also addresses the cooperation between software applications and end-users but in a different context. Our work concentrate on WEB based cooperation systems with strong constraints. We take into consideration the structure and constraints of the cooperation object. We also address the need of various cooperation modes and the resulting cooperation constraints.

In our sub-area of WEB based cooperation, there are four main issues that need to be resolved. The first issue is the proper description of a cooperation object. It should include the part-subpart relationship since the cooperation object can consist of many components and each component, in turn, can consist of subcomponents. It should also include some type of characteristics (e.g., ideal characteristics) for both the cooperation object (referred to as global characteristics) and characteristics for all its direct and indirect components (component characteristics). In order for the components to work together, their characteristics should be within some constraints referred to as constraints on the component characteristics that are usually derived from global constraints for the cooperation object.

There are some application areas of cooperation where these characteristics and constraints are explicitly given and mapping between them is inherently related to cooperative activities. For example, in the design process of complex engineering systems (CES), both characteristics and constraints are explicitly described. In other application areas significant effort might be required in order to describe them. In each situation the goal of cooperation is to find values of all cooperation object parameters for which all constraints are satisfied and the characteristics are “as close as possible” to ideal characteristics (e.g., Baszun and Czejdo, 1995, 1996).

The second issue is the proper description of a cooperation mode describing roles of different team members. There are many possible cooperation modes and they can vary for different parameters of each component. In order to develop the formal description of cooperation modes, we suggest using the primitive modes: centralized and decentralized.

The centralized primitive mode could also be called autocratic since the decision is left only for the specified team member. The decentralized primitive mode could also be called democratic since the decision is reached if the members agree on parameter changes. Based on these two primitive modes, we can describe