Chapter XVIII
Artifacts for Collaborative Software Development
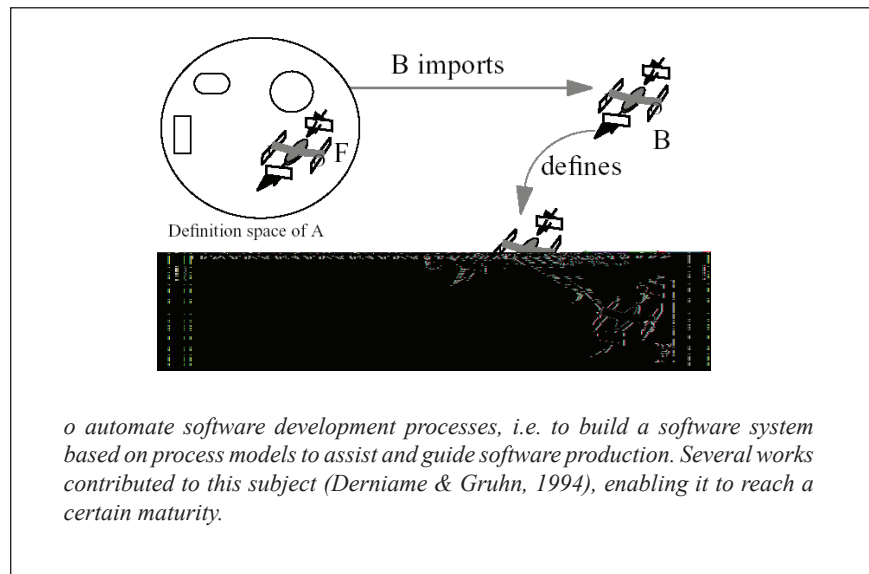
Pierre F. Tiako
Langston University, USA

INTRODUCTION

The development of software applications generally requires the following: hardware resources (computers, networks, peripherals, etc.), software resources (data, tools, etc.), human resources (individuals with various qualifications), and working methods. These resources are distributed in different autonomous software development environments. A single environment does not always have all the necessary resources to realize some large and/or complex projects. Therefore, collaboration between the environment in charge of the project (coordinator) and others (contractors) will be required to do the job.

While several research projects have contributed to various aspects of collaboration among software development environments during the past decade, little has been done on explicitly defining and modeling processes and environment artifacts involved in such partnerships. That is what this chapter is about. In the context of this study, environments work together by assigning tasks and sharing working methods. Tasks and working methods can be defined explicitly using process models. Process models, already the main focus in monolithic software development, will still be an important factor in our approach of collaborative software development. Because they are process-based, all software development environments considered here will be qualified in the continuation of Process-sensitive Software Engineering Environments (PSEEs).

Table 1. Brief process modeling history and limits

o automate software development processes, i.e. to build a software system based on process models to assist and guide software production. Several works contributed to this subject (Derniame & Gruhn, 1994), enabling it to reach a certain maturity.

BACKGROUND

The objective of developing large software projects can be reached more easily when the work to be done is divided into tasks (Smith, Hale & Parish, 2001) and assigned to various PSEEs. Tasks and working methods can be explicitly defined using process models. Table 1 proposes a brief history of process modeling and its limits.

Software processes are modeled before being assigned to PSEEs for performance. Collaborations to deal with these issues have also been discussed in information systems (Hahn, Jarke & Rose, 1991; Mookerjee & Chiang, 2002).

A simple solution toward this goal is to provide PSEEs with protocols that allow them to communicate in order to import or export process components as well as other PSEE artifacts. First investigations presented hereafter have been made in the subject by Oz (Ben-Shaul & Kaiser, 1995, 1998) and Federated PSEE (Basile, Calanna, Nitto, Fuggetta & Gemo, 1996) approaches.

Oz proposes to compose different instances of PSEEs where each of them is devoted to supporting the development process executed by a single organization. All these PSEEs run autonomously according to their own processes, but they can interact to accomplish common activities such as the integration test of software components that have been independently developed by various organizations. In Oz, the strategy through which a common activity is executed is called a summit. In a summit, one site acts as a coordinator. It receives from the other sites all data needed to execute the common activity, executes the activity, and sends the results back to the other sites. This behavior is obtained by directly implementing the summit protocol and procedures as a basic mechanism of the PSEEs.

Federated PSEEs takes Oz as a starting point and allows several interorganization policies to be implemented and combined. The solution of this issue is to provide a set of basic operations to specify any interorganization policy. Examples of operations are a search for the physical location of

Related Content

An Efficient Index Structure for Spatial Databases
Kap S. Bangand Huizhu Lu (1996). *Journal of Database Management (pp. 3-16).*
www.irma-international.org/article/efficient-index-structure-spatial-databases/51164

Predicting Software Abnormal State by using Classification Algorithm
Yongquan Yanand Ping Guo (2016). *Journal of Database Management (pp. 49-65).*
www.irma-international.org/article/predicting-software-abnormal-state-by-using-classification-algorithm/165162

Visualization of Predictive Modeling for Big Data Using Various Approaches When There Are Rare Events at Differing Levels
Alan Olinsky, John Thomas Quinnand Phyllis A. Schumacher (2018). *Handbook of Research on Big Data Storage and Visualization Techniques (pp. 604-631).*
www.irma-international.org/chapter/visualization-of-predictive-modeling-for-big-data-using-various-approaches-when-there-are-rare-events-at-differing-levels/198779

Data Quality: An Assessment
Jeretta Horn Nord, G. Daryl Nord, Hongjiang Xuand Elizabeth S. Myrin (2007). *Contemporary Issues in Database Design and Information Systems Development (pp. 265-286).*
www.irma-international.org/chapter/data-quality-assessment/7027

INDUSTRY AND PRACTICE: Another Look at On Time and Within Budget: An Agency Theory Explanation
Robert C. Mahaneyand Albert L. Lederer (1999). *Journal of Database Management (pp. 41-42).*
www.irma-international.org/article/industry-practice-another-look-time/51221