IDEA GROUP PUBLISHING
701 E. Chocolate Avenue, Hershey PA 17033-1117, USA
Tel: 717/533-8845; Fax 717/533-8661; URL-http://www.idea-group.com
ITB7995

Chapter III

Metrics for Controlling Database Complexity

Coral Calero, Mario Piattini and Marcela Genero
Grupo ALARCOS – Departamento de Informática
Universidad de Castilla-La Mancha, Spain

INTRODUCTION

Software engineers have been proposing large quantities of metrics for software products, processes and resources (Fenton and Pfleeger, 1997; Melton, 1996; Zuse, 1998). Metrics are useful mechanisms in improving the quality of software products and also for determining the best ways to help practitioners and researchers (Pfleeger, 1997). Unfortunately, almost all the metrics put forward focus on program characteristics (e.g., McCabe, 1976, cyclomatic number) disregarding databases (Sneed and Foshag, 1998). As far as databases are concerned, metrics have been used for comparing data models rather than the schemata itself. Several authors (Batra et al., 1990; Jarvenpaa and Machesky, 1986; Juhn and Naumann, 1985; Kim and March, 1995; Rossi and Brinkemper, 1996; Shoval and Even-Chaime, 1987) have compared the most well-known models--such as E/R, NIAM and relational--using different metrics. Although we think this work is interesting, metrics for comparing schemata are needed most for practical purposes, like choosing between different design alternatives or giving designers limit values for certain characteristics (analogously to value 10 for McCabe complexity of programs). Some recent proposals have been published for conceptual schemata (MacDonell et al., 1997; Moody, 1998; Piattini et al., 2001), but for conventional databases, such as relational ones, nothing has been proposed, excepting normalization theory.

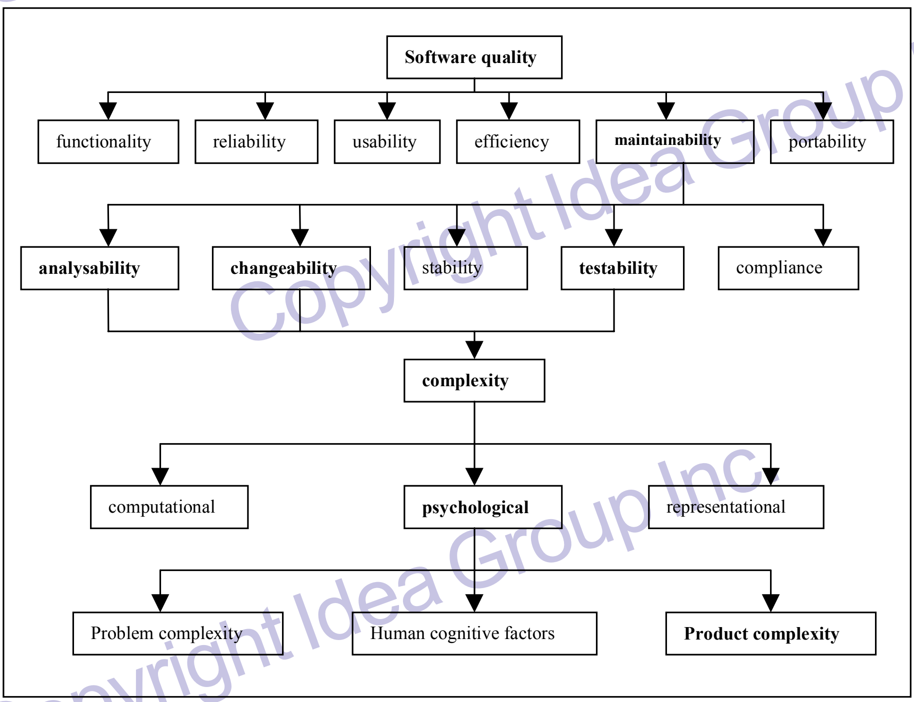
This lack of metrics support could be explained, as databases have been until recently just simple files/tables with minor contributions to the complexity of the overall system. However, this is no longer the case because databases now play a new role with information systems being their core, and a new database generation is necessary for supporting all the new applications. This new generation is the "third database generation" (Carey et al., 1990; Cattell, 1991), where new data types, rules, generalizations, complex objects and functions (Stonebraker and Brown, 1999) are being supported within the database realm.

Databases are becoming more complex, and it is necessary to measure schemata complexity in order to understand, monitor, control, predict and improve database development and maintenance projects. In modern information systems (IS), the database has become a crucial component, so there is a need to propose and study some measures to assess its quality.

Database quality depends on several factors: functionality, reliability, usability, efficiency, maintainability and portability (ISO, 1999). Our focus is on maintainability, because maintenance accounts for 60 to 90% of lifecycle costs, and it is considered the most important concern for modern IS departments (Frazer, 1992; McClure, 1992; Pigoski, 1997).

The International Standard, ISO/IEC 9126, distinguishes five subcharacteristics for maintainability: analysability, changeability, stability, testability and compliance (see Figure 1). Analysability, changeability and testability are in turn influenced by complexity (Li and Cheng, 1987). However, a general complexity measure is *the impossible holy grail* (Fenton, 1994), i.e., it is impossible to get one value that captures all the complexity factors of a database. Henderson-Sellers (1996) distinguishes three types of complexity: computational, psychological and representa-

Figure 1. Relation between software quality and product complexity



Related Content