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
ITB8079

Chapter XIII

Information System Design based on Reuse of Conceptual Components

P. Bertolazzi
IASI-CNR

M.G. Fugini and B. Pernici
Politecnico di Milano

INTRODUCTION

Current trends in modern Information System development are more and more based on the exploitation of the experience gained in previous developments of similar systems. While in the early years of Information System development each system was analyzed and designed from scratch, it alreadybecame clear in the 1980s that developing a system which meets user expectations implies being widely aware of the characteristics of the specific application domain and being able to build upon previous experience. Such "reuse of experience'' was initially performed informally. Meanwhile, in recent years, systematic approaches based on reuse of code and design experience have appeared in the literature of software engineering and in various commercial development environments (Biggerstaff, 1989; Krueger, 1992). For example, application frameworks (Johnson, 1997), seen as code and related specification and design artifacts, are typical structures that have been studied in various projects as large-grained components or application skeletons (see for instance (Fusaschi and Montini, 1997) and (D'Souza and Wills, 1998).

In general, it is now well acknowledged that reuse can bring benefits, but it is also an intensive, time-consuming task that needs to be undertaken systematically. In fact, it brings about issues such as:

1) the availability of structured component bases;
2) tools that help in selecting suitable candidates;
3) techniques for adapting those components that do not match the required functionalities perfectly, but "similarly" (and, therefore, useful);
4) techniques to *produce* reusable components at the correct granularity level (neither too small—and, therefore, specialized—nor too large—and, therefore, difficult to be tailored) and to *properly store* them in the component base (to facilitate retrieval of a useful component).

In summary, theories and tools are needed for component reuse and production that make the reuse effort feasible and effective. We will consider reuse in the context of Information System design as a *design process based on components* that must be *selected* (because identified as suitable to the current application), *aggregated* and *adapted* to the current requirements.

We define a *reusable component* as a unit of design for which the following elements are defined: 1) a *model*; 2) a *name*, identifying the component, and 3) *design guidelines*. Guidelines are in the form of design documentation provided in order to illustrate the context where the component can be reused. Guidelines include *constraints*, identifying, for instance, which other components must be used in combination with the one being considered.

Reuse based on components can be performed at different levels: 1) at conceptual level, 2) at design level, and 3) at implementation level. At *conceptual level*, reusable components are process skeletons and conceptual data structures (Castano et al, 1998). At *design level*, components are software modules to be selected and personalized; techniques and criteria for combining them have to be applied (Baumer et al, 1997). The *implementation level* is usually implied by the first two levels since the techniques for selecting and tailoring components at conceptual and design level are the basis for building an Information System from reusable code components (Bellinzona et al., 95; Damiani and Fugini, 1997).

Some approaches propose to reuse also the *design steps,* such as refinement steps applied during the design of a database schema (Castelli, 1999); this simplifies redesign that occurs when the conceptual schema evolves. This issue is discussed for instance in (Bellinzona et al., 1995), where conceptual design, logical design and implementation are linked through a sequence of steps that map the specification objects into design objects and eventually into the object-oriented code of the new application. Tools tracing the steps and the design history are described in the paper.

We define reuse from two different points of view:

- *Design for reuse*: approaches where software components (called *design components* in this chapter) and the related documentation in terms of conceptual schemas for data and processes (called *conceptual components* in this chapter) are developed in such a way that they can be reused in different contexts. Design for reuse can be performed by either generalizing so called "best practices" or by building components starting from the comparison and integration of many different systems having similar characteristics.
- *Design by reuse*: approaches where an Information System is designed by *assembling* and *personalizing* components both at the conceptual and at the design level.

Information Overload in Augmented Reality: The Outdoor Sports Environments

Rui Miguel Pascoaland Sérgio Luís Guerreiro (2017). *Information and Communication Overload in the Digital Age (pp. 271-301).*
www.irma-international.org/chapter/information-overload-in-augmented-reality/176575