Chapter VIII

The Role of Standards for Interoperating Information Systems

Wilhelm Hasselbring
INFOLAB, Tilburg University

INTRODUCTION

For integrating heterogeneous information systems, semantic interoperability is necessary to ensure that exchange of information makes sense — that the provider and requester of information have a common understanding of the ‘meaning’ of the requested services and data. Effective exchange of information between heterogeneous systems needs to be based on a common understanding of the transferred data. This chapter discusses the role of domain-specific standards for managing semantic heterogeneity among dissimilar information sources. The process of integrating such heterogeneous information systems is also discussed in this context, whereby standards play a central role for ‘initiating’ top-down processes by means of defining common data models for the involved information sources.

BACKGROUND

Traditionally, the integration of heterogeneous information systems proceeds in a bottom-up process. Information stored in existing legacy systems is analyzed with respect to potential overlaps, whereby overlapping data in dissimilar systems describes the same or related information. The overlapping areas of related information sources are subsequently integrated. The integration is usually realized by means of mediators, federated database systems or such—like system architectures. Typical goals for the integration of existing systems are the development of global applications that access the data from multiple sources as well as consistency management of information that is stored in related systems.

Let us consider digital libraries as an example domain where the integration of existing information sources is one of the central problems to be solved (Schatz & Chen, 1996). As one result of a bottom-up integration of those existing informa-
tion sources, the structure of the merged common data model (schema) is determined by the overlaps among the local data models, and not by the requirements of global applications. The maintenance of such integrated models is a problem, because those merged models rapidly become very complex; usually more complex than required for the actual integration goals. This situation can lead to severe scalability problems with respect to execution performance, usability and maintenance.

ISSUES, CONTROVERSIES AND PROBLEMS WITH THE TRADITIONAL BOTTOM-UP INTEGRATION PROCESS

Various approaches for integrating heterogeneous information systems — e.g., federated database systems or mediator and agent architectures — have been proposed (Hurson, Bright & Pakzad, 1993, Elmagarmid, Rusinkiewicz & Sheth, 1998, Sheth, 1998, Wiederhold, 1996; Jennings & Wooldridge, 1998). We illustrate the traditional bottom-up process of integrating such heterogeneous information systems, by means of schema integration in federated database systems (Sheth & Larson, 1990). A federated database system is an integration of autonomous database systems, where both local applications and global applications accessing multiple database systems are supported. For federated database systems, the traditional three-level schema architecture must be extended to support the dimensions of distribution, heterogeneity, and autonomy. The five-level-schema-architecture of Sheth & Larson (1990) is generally accepted as the basic structure for schema integration in federated database systems or at least for comparison with other architectures of specific federated database systems (Conrad, Eaglestone, Hasselbring, Roantree, Saltor, Schönhoff, Strässler & Vermeer, 1997).

Figure 1 illustrates the bottom-up process for constructing the schema architecture in federated database systems which starts with an analysis of information stored in the local systems. To explain the schema types displayed in Figure 1: A local schema is the conceptual schema of a component database system, which is expressed in the (native) data model of that component database system. In a first step, the local schemas are translated into component schemas in the canonical data model of the federation layer. Then, export schemas are filtered from the component schemas and merged into a common federated schema. A component schema is a local schema transformed into the so-called canonical data model of the federation layer. The component, export and federated schemas are defined in this canonical data model. An export schema is derived from a component schema and defines an interface to the local data that is made available to the federation. A federated schema is the result of the integration of multiple export schemas, and thus provides a uniform interface for global applications. An external schema is a specific view on a federated schema or on a local schema, which serves as a specific interface for applications (local or global). To keep it simple, no external schemas are shown in Figure 1. For local applications, external schemas can be filtered from the local schemas. For global applications, external schemas are filtered from the federated schema.
Developing Secure Software Using UML Patterns
www.igi-global.com/chapter/developing-secure-software-using-uml-patterns/125296?camid=4v1a