Classifying and Reusing Conceptual Schemas

S. Castano V. De Antonellis
Dipartimento di Elettronica e Informazione - Politecnico di Milano
Piazza Leonardo da Vinci 32 - 20133 Milano
Email: {deantone,castano}@ipmel2.elet.polimi.it

B. Zonta
Dipartimento di Scienze dell’Informazione - Università di Milano
via Comelico 39/41 - 20135 Milano

Abstract. The paper presents a methodological approach to guide the application engineer to construct a Library of Entity-Relationship schemas, classified by means of indexing criteria and clustering techniques, and to extract reusable components from the existing selected schemas. Reusable components are defined as generic entities with associated meta-entities providing guidelines for reuse in a given application. Generic entities are derived from the analysis of the entities belonging to similar schemas within clusters of the Library. A set of reuse guidelines in form of meta-entities are given, suggesting how generic entities can be modified and tailored according to the needs of the application to be developed.

1 Introduction

The ever growing complexity of software applications, and the need to develop information systems that can be readily adapted to changes in user requirements, has led to an increasing demand for methods and tools for reusing knowledge, processes and results of previously developed applications in order to improve productivity and quality of the development and maintenance process [1, 4, 15]. In both the frameworks of software engineering and information systems development, the traditional approaches based on the design by transformation paradigm, according to refinement and/or abstraction strategies, are being extended to adopt, in an integrated way, the design by reuse paradigm. According to this paradigm, applications are developed not from scratch, but tailoring and personalizing reusable components [5, 21]. The possibility of reusing components at any development level (requirements, design, code) is strongly dependent on their capacity to behave as generic solvers of problems existing in a wide range of specific situations. Genericity is an essential property to support the reuse of components as pointed out in [21], and allows the definition of a minimal set of reusable components to represent the significant problems of the application domain under consideration [16, 18, 21]. The extraction of reusable components relies on the availability of a Library containing selected pre-existing conceptual schemas which can be readily accessed and analysed according to similarity criteria [12, 17]. Several approaches to reusability are being proposed and experimented in the object-oriented area [14, 25, 13, 22, 4, 9]. An approach to define reusable
components following the F-ORM object-oriented approach [7, 8], is presented in [5].

In this paper we consider reusability of Entity-Relationship schemas [3]. The problem is properly set since E-R schemas are widely and intensively produced in most organizations and their reuse can significantly improve design productivity. Precisely, we present a methodological approach for building a Library of Entity-Relationship schemas, classified by means of indexing criteria and clustering techniques, and for extracting reusable components from the existing selected schemas. Reusable components are defined as generic entities with associated meta-entities providing guidelines for reuse in a given application. Generic entities are derived from the analysis of the entities belonging to similar schemas within clusters of the Library. Guidelines associated to a reusable component provide a set of design suggestions about ways to incorporate that component in an application by means of possible adaptations and transformations. Responsible for the definition and maintenance of the Library and of the reusable components is the application engineer who is concerned with the following activities:

- organization of existing selected schemas in the Library according to a classification model, to support their search and retrieval by the application developer [11, 18, 20, 10];

- definition of the reusable components, extracting generic entities from schemas in the Library and defining corresponding meta-entities for their reuse [19, 25].

The proposed approach guides application engineers in both the activities. The paper is organized in the following way. Sect. 2 illustrates the model for the construction of the Library of E-R conceptual schemas. Sect. 3 focuses on the extraction of generic entities. In Sect. 4, methods to define reusable components are discussed. In Sect. 5, concluding remarks are given.

2 Construction of a Library of Conceptual Schemas

The analysis of about a hundred E-R schemas led us to conclude that an E-R schema is a kind of document suitable for automatic indexing. In fact, a schema appears to be a document, structured at both syntactic and semantic levels, ever belonging to a well identifiable (sub)domain. The structure of an E-R schema is constituted by interrelated elements labeled with meaningful names. These interrelated elements form a significant unit which, in turn, can be given a name, that is, the title of the schema. The linguistic labels of the schema elements, constitute, at least in principle, relevant terms for the description of the schema subject and are, therefore, good candidates for becoming descriptors of that schema. In practice, however, often the interpretation of a label cannot be immediate, because of the frequency of abbreviations, jargon, foreign words, or general terms which could be applied to any field and be specified only by their context (see, for instance, the schema in 2.1, where a credit dossier is called simply "dossier"). To overcome this obstacle, it is convenient, not to say necessary, to introduce normalizing interventions at the very moment of the label coinage [6]. Possible normalizations require to: eliminate articles, unnecessary prepositions, and the like; choose, among possible synonyms, those which are more pregnant in meaning (not “dossier” but “credit” or “credit dossier”); avoid, when possible, foreign words, acronyms and abbreviations.

In this section we present a methodological approach to the construction of a Library of conceptual schemas. The schemas descriptors, SDs, are extracted and used to