Quality Evaluation for Evolving Conceptual Database Design

Elvira Immacolata Locuratolo
CNR ISTI, Italy

INTRODUCTION

Research at boundary between *information systems* and *software development* is important for today's technology; (Buragga & Zaman, 2013). Before creating an information system, the business and organizational domain in which the information system is used must be represented. This representation, called *conceptual model*, is a critical point for the success of the information system development.

Model-driven engineering is an approach to software development that separates the specification of the system functionalities from its implementation on a particular platform. Models are the primary artifacts of software engineering. Two main categories of *mappings* between models are distinguished:

- Vertical mapping, relating system models at different abstraction levels;
- Horizontal mapping, relating models at the same abstraction level.

The requirements for engineering information modeling enclose complex objects, data exchange, data sharing, web based applications, imprecision uncertainty, and knowledge management. Techniques, such as web and artificial Intelligence, have been introduced in industrial applications.

Databases and Database systems play a main role in supporting data management and in implementing engineering information. Conceptual database design is defined by approaches to *mapping* from conceptual to logical database models (Elmasri & Navathe, 2015). Algorithms of conceptual database graph partitioning (Locuratolo & Rabitti, 1998; Spagnolo, 2000), originally introduced as algorithms of mapping from semantic data models to object database systems, have been also exploited as techniques for constructive conceptual database design (Locuratolo, 2013). *Information systems and software development* is a dynamic, interesting research area, which provides the finest and fastest way to solve typical problems.

Innovative evaluation of quality is introduced to design a database conceptual model which can result from a vertical mapping or a horizontal mapping (Locuratolo, 2014a). In the former case, the cost of specifications/proofs is given through the cardinality of variables and constants; in the latter case, the conceptual model is evaluated in terms of hidden classes. These represent the *saving* that is obtained for what has not been explicitly specified/proven. The hidden classes can also be exploited to test other quality desiderata/dimensions, such as *easiness of use* and *flexibility*. The numeric saving of consistency can be determined from the corresponding conceptual evaluations. The vice-versa is not possible.

In this article, a new approach to conceptual database design, called *evolving conceptual database design*, is introduced. The approach integrates a *structure for the preservation of classes/concepts* within the conceptual database design: starting from a database conceptual graph, an ontology defined at the boundary between concept theory and computer science, called structure for the *preservation of database classes/concepts,* is designed (Locuratolo, 2015b); the leaves of this structure are mapped to classes of a logical database model. A

DOI: 10.4018/978-1-5225-2255-3.ch175

constructive approach of logical database design is considered to handle some dynamic aspects of the conceptual database design and the quality of this approach is evaluated.

BACKGROUND

Model-driven engineering is an approach to software development that separates the specification of the system functionality from its implementation on a particular platform. A *model-driven architecture* makes models the primary artefacts of software engineering. The development consists in the definition of a platform-independent model (PIM) of a system and in the application of parameterized transformations to this PIM in order to obtain one or more specific platforms. Another important specification is *architecture-driven modernization*, which is the process of understanding and evolving existing software. It involves modifications, reuse and enterprise- application integrations. An approach to modernization is a way to revitalize existing applications and systems. It concerns a wide variety of models and mappings between models, allowing integration and transformation of those models. Over the last few years, *model-driven architecture* has been integrated with *enterprise information systems,* and other technologies, such as *semantic web, semantic web services, and ontology* techniques.

Graphs of classes representing *conceptual/logical database models* are based on concepts which are common in *enterprise information systems, model-driven engineering, and ontology*. New engineering requirements and current technologies favour the *conceptual database design evolution*. In the following, definitions useful to the purposes of the chapter are introduced: first a description of the *conceptual database design* is provided; then an extension of some definitions is given.

Conceptual Database Design

In this section, definitions useful to introduce the concept of conceptual database design are given:

- **[Database Conceptual Class]:** Couple (v, V), where v is a label denoting the class objects and connoting the corresponding concept, and V are the attributes associated with the objects.
- **[Is-a Relationship]:** Conceptual class (u, U) is in *is-a relationship* with conceptual class (v, V) *iff*
  - The objects of class (u, U) are enclosed into the objects of class (v, V);
  - The attributes of class (u, U) are inherited from class (v, V);
  - Class (u, U) can have specific attributes
- **[Database Conceptual Graph]:** Oriented acyclic graph G_c of database classes in *is-a* relationship.
- **[Revised Partitioning]:** Step-sequence of decompositions resulting in the *finest partition* of G_c. At each step of decomposition, the following properties hold:
  - **Root Partitioning:** The root objects of a conceptual graph G_c are partitioned into the root objects of the conceptual graphs obtained after the decomposition.
  - **Root Labeling:** The root labels of the graphs obtained from decomposition represent the partition. These labels are obtained combining the root label of the decomposed graph with the labels of the root directed descendants.
  - **Root Structuring:** The root labels can be decomposed into two parts separated by the "-"sign: that on the left of this sign consists of label intersections, whereas that on the right consists of label unions. One of the two parts can be empty.

Related Content

Metaheuristic Algorithms for Detect Communities in Social Networks: A Comparative Analysis Study
Aboul Ella Hassanienand Ramadan Babers (2018). *International Journal of Rough Sets and Data Analysis (pp. 25-45).*
www.irma-international.org/article/metaheuristic-algorithms-for-detect-communities-in-social-networks-a-comparative-analysis-study/197379

Characteristics of Resources and the Impact on Projects
Loredana Arana (2015). *Encyclopedia of Information Science and Technology, Third Edition (pp. 5029-5036).*
www.irma-international.org/chapter/characteristics-of-resources-and-the-impact-on-projects/112951

Climate Change: Inclusion of Gender and Cultural Diversity in Climate Change Actions
Mgbeodichinma Eucharia Onuoha Oragwa (2021). *Encyclopedia of Information Science and Technology, Fifth Edition (pp. 1398-1407).*
www.irma-international.org/chapter/climate-change/260274

Deep Mining Technology of Database Information Based on Artificial Intelligence Technology
Xiaoai Zhao (2023). *International Journal of Information Technologies and Systems Approach (pp. 1-13).*
www.irma-international.org/article/deep-mining-technology-of-database-information-based-on-artificial-intelligence-technology/316458

Visual Information Analysis for Interactive TV Applications
Evlampios Apostolidis, Panagiotis Sidiropoulos, Vasileios Mezarisand Ioannis Kompatsiaris (2015). *Encyclopedia of Information Science and Technology, Third Edition (pp. 2208-2218).*
www.irma-international.org/chapter/visual-information-analysis-for-interactive-tv-applications/112631