

Chapter XIII

Interrogative Agents for Data Modeling

Vincenzo Deufemia

Università di Salerno, Italy

Giuseppe Polese

Università di Salerno, Italy

Mario Vacca

Università di Salerno, Italy

INTRODUCTION

The problem of changes in software development is a complex one, and it is almost impossible to avoid it. Indeed, the continuous evolution of the real world causes frequent changes in functional requirements, which entail frequent modifications to the software, yielding a gradual decay of its overall quality. To tackle this problem, two methodologies have been proposed: waterfall methodologies, and incremental/iterative methodologies. The formers try to prevent changes, whereas the

second ones consider system development as a step by step process.

The concept of software refactoring is at the base of iterative and incremental methodologies. According to Fowler software refactoring is “... *a disciplined technique for restructuring an existing body of code, altering its internal structure without changing its external behavior.*” (M. Fowler, <http://www.refactoring.com/>).

Analogously, changes to the database structure are also relatively frequent (Roddick, 1995). They are particularly critical, since they affect not

only the data, but also the application programs accessing them (Ambler & Sadalage, 2006; Karahasanovic, 2001). Therefore, similarly to software refactoring, database refactoring aims to modify the database schema, and to change the corresponding application programs accordingly.

Ambler & Sadalage (2006) gave the definition of database refactoring: It “*is a simple change to a database schema that improves its design while retaining both its behavioural and informational semantics*.” The database refactoring is the basis of evolutionary data modeling methodologies, which are the database analogous of the iterative and incremental ones. Ambler & Sadalage (2006) also observed that a disadvantage in the application of refactoring is the lack of mature supporting tools.

In this paper we deal with the problem of developing tools supporting the evolutionary data modeling process. First of all, we observe that the characteristics of the problem can be naturally framed in the agent paradigm, because the evolutionary data modeling can be seen as a process in active databases able to change their beliefs and structure. Moreover, the evolutionary data modeling can be compared to the design of an agent acting in an open environment: the environment can be represented by the user needs and requirements (which change in an unforeseeable way), while the database development process is represented by the evolution of a reactive agent. Then, by following the *AOSE* (Agent-Oriented Software Engineering) view, we show that the use of tools and techniques from AI (Artificial Intelligence) can help facing the problem of developing supporting tools to automate evolutionary data modeling. To this end, after a brief introduction to the basic concepts in agent theory, and the highlighting of relationships among agents, software engineering, and databases, we point out the correspondence between agents and data modeling by showing a suitable architecture based on the logic of interrogation (Hintikka *et al.*, 2002).

BACKGROUND

Before dealing with the problem of database refactoring, along with the more general data modeling, and the role of agents in solving it, we need to briefly introduce the concept of agent, highlighting advantages of agent technology.

The term agent has not reached a universally accepted definition, as pointed out by Franklin & Graesser (1996), but the essence of being an agent can be summarized as follows:

“*An **autonomous agent** is a system situated within and a part of an environment that senses that environment and acts on it, over time, in pursuit of its own agenda and so as to effect what it senses in the future.*” (Franklin & Graesser, 1996).

Luck *et al.* (2004) proposed to divide applications of agents in three main categories:

- *Assistant agents*, which replace humans in the execution of some task (e.g., agents for hotel reservation);
- *Multi-agent decision systems*, where the agents in the system make some joint decisions;
- *Multi-agent simulation systems*, used to simulate real-world domains like biological populations.

In this paper we are mainly concerned with particular applications of the first type, in which agents operate on databases (see, for example, de Carvalho Costa *et al.*, 2003; Magnanelli & Norrie, 2000). In particular, we are concerned with reactivity (the ability of the agent to respond in an appropriate way to the environment changes), and with the design of reactive systems needing to interact in open system environments^a.

Different architectures have been designed to realize the features of agents, the most famous of

7 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:
www.igi-global.com/chapter/interrogative-agents-data-modeling/20694

Related Content

Complementing Business Process Verification by Validity Analysis: A Theoretical and Empirical Evaluation

Pnina Soffer and Maya Kaner (2011). *Journal of Database Management* (pp. 1-23).

www.irma-international.org/article/complementing-business-process-verification-validity/55131

Databases Modeling of Engineering Information

Z.M. Ma (2009). *Selected Readings on Database Technologies and Applications* (pp. 65-84).

www.irma-international.org/chapter/databases-modeling-engineering-information/28546

Data Warehousing Interoperability for the Extended Enterprise

Aristides Triantafyllakis, Panagiotis Kanellis and Drakoulis Martakos (2004). *Journal of Database Management* (pp. 73-84).

www.irma-international.org/article/data-warehousing-interoperability-extended-enterprise/3317

Possibility Theory in Protecting National Information Infrastructure

Richard Baskerville and Victor Portougal (2005). *Advanced Topics in Database Research, Volume 4* (pp. 325-340).

www.irma-international.org/chapter/possibility-theory-protecting-national-information/4381

Logic Databases and Inconsistency Handling

José A. Alonso-Jiménez, Joaquín Borrego-Díaz and Antonia M. Chávez-González (2005). *Encyclopedia of Database Technologies and Applications* (pp. 336-340).

www.irma-international.org/chapter/logic-databases-inconsistency-handling/11169