Chapter X
Data Quality–Based Requirements Elicitation for Decision Support Systems

Alejandro Vaisman
Universidad de Buenos Aires, Argentina

ABSTRACT

Today, information and timely decisions are crucial for an organization’s success. A decision support system (DSS) is a software tool that provides information allowing its users to make decisions timely and cost effectively. This is highly conditioned by the quality of the data involved, usually stored in a data warehouse, and by a sound and complete requirements analysis. In this chapter we show that conventional techniques for requirements elicitation cannot be used in DSS, and present a methodology denoted DSS-METRIQ, aimed at providing a single data quality-based procedure for complete and consistent elicitation of functional (queries) and nonfunctional (data quality) requirements. The outcomes of the process are a set of requirement documents and a specification of the operational data sources that can satisfy such requirements. We review the state-of-the-art in the field, and show that in spite of the tools and methodologies already proposed for the modeling and design of decision support systems, DSS-METRIQ is the first one that supports the whole process by means of an integral technique.

INTRODUCTION

It is a well-known fact that, among the phases of the software development process, analysis and specification of functional and nonfunctional requirements is a crucial one. The lack of good requirements specification is a major cause of failure in software development (Thayer, 2002). The software engineering community has developed many useful tools for requirements analysis in transactional systems. These kinds of systems deal with the day-to-day operation of an organization. Decision support systems (DSS) are of a completely different kind: they are focused on integrating data and models in order to improve the decision-making process. The data that feed
Data Quality-Based Requirements Elicitation for Decision Support Systems

A DSS generally reside in a data warehouse. The software development cycle of DSS has particularities that require applying methodologies different than the ones used for operational systems. The reason for this is twofold: on the one hand, traditional methodologies have been thought and designed with transactional systems in mind; on the other hand, specific methodologies applicable to DSS arose as ad-hoc answers to practical needs, and most of them are just mere enumerations of activities that must take place in order to implement the system, focusing on populating the data repository while ignoring important issues like the impact of changes in the operational data sources, or worse, if these data sources satisfy the users’ information requirements. New sources of failure are present in DSS: correctness and trustworthiness of the information are the basis of the decision-making process. We do not only need to understand the user’s information needs, but also account for keeping the data repository up-to-date according to user specifications. Also, update processes and their frequency must be considered, as well as the analysis of the quality and completeness of the data sources.

It follows that there is a need for techniques that, besides accounting for the software process cycle and functional requirements, also consider the quality of the information the system will deliver. There are several reasons for this. For instance, most of the time, people developing information systems do not consider the impact of low quality data (Kimball, Reeves, Ross, & Thornthwaite, 1998). Low data quality is more a rule than an exception. Just to give an example, it has been detected in the U.S., that approximately 50 to 80% of the computerized criminal records are inaccurate, incomplete, or ambiguous (Strong, Yang, & Wang, 1997). So far, the contribution of software engineering for addressing the problems stated has been limited, although many techniques have been proposed in order to analyze and measure a data quality requirement. Some examples of these techniques are GQM (goal question metric) (Basili, Caldiera, & Rombach, 1992) and QFD (quality function deployment) (Akao, 1997).

In summary, traditional software development methodologies do not apply to DSS, and focus on software correctness, paying little attention to the problem of data quality and completeness, given that, in general, this is not considered an issue in the requirement analysis phase of the software development cycle for operational systems. Based on these points, we propose a methodology called DSS-METRIQ that integrates concepts of requirements engineering and data quality, in order to provide a comprehensive solution to the requirements elicitation process specifically oriented to DSS.

Case Study

Throughout the chapter we will discuss the following case study. We must collect requirements for a DSS for a wholesale chain called “Los Andes” (specialized in food products). The chain has three branches in the Argentina countryside. The project involves the development of a data warehouse and a DSS for supporting the daily tasks of decision makers. The company has many different sources of operational data. We must carry out the requirements elicitation process, with the following goals in mind: discovery and documentation of user queries, addressing the information quality required by our customer (that we must also help to define). It will also be our task to analyze data quality in each one of the data sources, indicating for each piece of data, the data source from which we will obtain it and the data quality we can expect. Thus, we must specify the queries (functional requirements) that could be addressed by the system (given the available data sources) satisfying the data quality levels imposed by our customer (nonfunctional requirements). There will be a requirements engineering team, composed of a project leader, a training team, a team for carrying out the interviews, a data processing team,