Goal-Oriented Requirement Analysis for Data Warehouse Design

Paolo Giorgini
University of Trento, Italy

Stefano Rizzi
University of Bologna, Italy

Maddalena Garzetti
University of Trento, Italy

Abstract

Several surveys indicate that a significant percentage of data warehouses fail to meet business objectives or are outright failures. One of the reasons for this is that requirement analysis is typically overlooked in real projects. In this paper we propose a goal-oriented approach to requirement analysis for data warehouses, based on the Tropos methodology. Two different perspectives are integrated for requirement analysis: organizational modeling, centered on stakeholders, and decisional modeling, focused on decision makers. Our approach can be employed within both a demand-driven and a mixed supply/demand-driven design framework: in the second case, while the operational sources are still explored to shape hierarchies, user requirements play a fundamental role in restricting the area of interest for analysis and in choosing facts, dimensions, and measures. The methodology proposed, supported by a prototype, is described with reference to a real case study.

1 Introduction

Several surveys indicate that a significant percentage of data warehouses (DWs) fail to meet business objectives or are outright failures. One of the reasons for this is that requirement analysis is typically overlooked in real projects, mainly since [21]:

- Warehousing projects are long-term ones, and most requirements cannot be stated from the beginning.
- Information requirements for DW applications are difficult to specify since decision processes are flexibly structured, poorly shared across large organizations, jealously guarded by managers, and unstable in time to keep pace with evolving business processes.
- Requirements for decision making often refer to information that does not exist in the required form, and must be derived from data sources.

Though most of the methodologies for DW design claim there must be a phase dedicated to analyzing the business requirements (e.g., [9, 13, 16]), there is no consensus on what relevance should be assigned to such phase. Indeed, the approaches to DW design are usually classified in two categories [21]:

- *Supply-driven* (also called *data-driven*) approaches design the DW starting from a detailed analysis of the data sources [11, 8, 18]. User requirements impact on design by allowing the designer to select which chunks of data are relevant for the decision making process and by determining their structuring according to the multidimensional model.

- *Demand-driven* (or *requirement-driven*) approaches start from determining the information requirements of DW users [19, 3]. The problem of mapping these requirements onto the available data sources is faced only *a posteriori*.

While supply-driven approaches somehow simplify the design of ETL, since each data in the DW is rooted in one or more attributes of the sources, they give user requirements a secondary role in determining the information contents for analysis, and give the designer little support in identifying facts, dimensions, and measures. Conversely, demand-driven approaches bring user requirements to the foreground, but require a larger effort when designing ETL.

Supply-driven approaches are feasible when all of the following are true: (1) detailed knowledge of data sources is available *a priori* or easily achievable; (2) the source schemata exhibit a good degree of normalization; (3) the complexity of source schemata is not high. In this case, conceptual design is heavily rooted on source schemata and can be largely automated (e.g. see [8]). Our on-the-field experience shows that requirement analysis can then be carried out informally, based on simple requirement glossaries (such as in [14]) rather than on formal diagrams. On the other hand, we believe that such an informal approach is unsuitable for other design frameworks.

In this paper we propose a goal-oriented technique to requirement analysis for DWs, based on the Tropos methodology [2]. The technique can be employed:

- within a demand-driven framework, that is the only alternative whenever a deep analysis of data sources is unfeasible (e.g. if the DW is fed from an ERP system, whose logical schema is huge and hardly understandable), or data sources reside on legacy systems whose inspection and normalization is not recommendable. In this case, conceptual design will be directly based on requirements.

- within a mixed supply/demand-driven framework. In this case, requirement analysis and source inspection are carried out in parallel; conceptual design is still carried out in a semi-automated way, like in the supply-driven framework, but leaning on user requirements to reduce its complexity. The mixed framework is recommendable when source schemata are well-known but their size and complexity are substantial. In fact, the cost for a more careful and formal analysis of requirement is balanced by the quickening of conceptual design.

Our technique adopts two different perspectives for requirement analysis: *organizational modeling*, centered on stakeholders, and *decisional modeling*, focused on decision makers. Decisional modeling is directly related to the information needs of decision makers; with reference to the terminology introduced in [21], it achieves *to be* analysis. On the other hand, organizational modeling is aimed at *as is* analysis; it has a primary role in enabling identification of facts and in supporting the supply-driven component of the approach. The diagrams produced, that relate enterprise goals to facts, dimensions, and measures, are then used during conceptual design: within a demand-driven design framework, the requirements are translated into a conceptual schema to be mapped on data sources *a posteriori*; within a mixed framework, while the data sources are still explored to shape hierarchies, user requirements play a fundamental role in restricting the area of interest for analysis and in determining facts, dimensions, and measures.

The paper is structured as follows. In Section 2 we summarize the most relevant literature related to requirement analysis in DW design. Section 3 illustrates the technique we propose for requirement analysis by discussing organizational and decisional modeling. Section 4 shows how our technique relates to conceptual design within both demand-driven and mixed design frameworks. Finally, Section 5 draws the conclusions and introduces the prototype supporting our approach.

2 Related Literature

In the field of DW design, it is necessary to distinguish between supply- and demand-driven approaches. The prototypic supply-driven approach dates back to 1992, when Inmon claimed that the development of DWs is data-driven, as opposed to the requirement-driven development of operational systems [12]. Other supply-driven approaches were proposed in [11], [8], and [18], where conceptual design of the DW is rooted in the schema of operational sources and is carried out starting, respectively, from the identification of measures, from the selection of facts, and from a classification of the operational entities. Also the comprehensive design method described in [16] leans on a conceptual model; a mixed approach to conceptual design is recommended, but no details are given.

In demand-driven approaches, collecting user requirements is given more relevance. In [21], a wish-list for DW design methodologies is proposed, and a multi-stage technique for requirement analysis is outlined. Here, two different phases are interlaced: *as is analysis*, aimed at analyzing and describing the actual information supply, and *to be analysis*, aimed at analyzing the information demand and matching it with the supply. In [19], a goal-oriented approach based on the *goal-decision-information* model is proposed. Though this approach shares some similarities with ours, it mainly focuses on requirement analysis and does not show how to move from requirements to design. A process-oriented approach is presented in [3], where three different perspectives at increasing levels of detail, each associated to a specific requirement template, are used. Though the authors recommend to iteratively and incrementally gather requirements with use cases, a few details are given and no examples are provided, so a comparison is very hard.

In [1], a goal-oriented method to support the identification and design of DWs is presented. This approach can be regarded, like ours, as mixed demand/supply driven. The main difference is that organizational modeling is not supported and that requirement analysis starts from the goals of decision makers. Goals are analyzed separately using abstractions sheets, and general considerations about how they relate to the organization activities are given in natural language. Conversely, in our approach an explicit goal model of the organization is given and the analysis of decision makers' goals is directly related to such a model. Moreover, in our goal analysis, goals are decomposed in subgoals and specific relationships between goals are specified. Another important difference with our approach, is that we support early requirements analysis [6, 22] that allows for modeling and analyzing processes that involve multiple participants (both humans and software systems) and the intentions that these processes are supposed to fulfill. By so doing, one can relate the functional and non-functional requirements of the system-to-be to relevant stakeholders and their intentions.

An interesting case-based comparison of supply- and demand-driven approaches can be found in [15]. Remarkably, it is concluded that data-oriented and goal-oriented techniques are complementary, and may be used in parallel to achieve optimal design.

Finally, it is worth to mention that a few CASE tools for DW design have been implemented, either from software vendors or as research protoypes. In ADAPT [4] and in GOLD [17] the conceptual schema for the DW is directly drawn by the designer, thus a demand-driven approach is implied—though no active support for requirement analysis is given. Conversely, in WAND [10] the conceptual schema is semi-automatically derived from the operational schemata, thus implementing *de facto* a supply-driven approach.

3 Requirement Analysis

Tropos [2, 5] is an agent-oriented software development methodology, based on the $i*$ conceptual framework [22], where the concepts of *agent*, *goal*, and related mentalistic notions are used to support all software development phases, from early requirement analysis to implementation. Tropos differs from other goal-oriented methodologies since it moves the notions of agent and goal to the early stages of software development. During early requirement analysis, the requirements engineer identifies the domain stakeholders and models them as social actors, who depend on one another for goals to be fulfilled, tasks to be performed, and resources to be furnished. Through these dependencies, one can answer *why* questions, besides *what* and *how*, regarding system functionality. Answers to *why* questions ultimately link system functionality to stakeholder needs, preferences, and objectives.

The Tropos methodology has been successfully applied in different domains. In the following we summarize the part of the Tropos notation that can be used in the DW context:

- *Actors.* An actor represents an enterprise stakeholder. More precisely, it can model a physical or software *agent* (e.g., Mr. Brown), a *role*, meant as an abstract characterization of the behavior one or more agents take in a specific context (e.g., sale analyst), or a *position*, i.e. a set of roles generally played by a single agent (e.g., marketing manager). Graphically, actors are represented by circles.

- *Strategic dependencies.* A dependency represents an "agreement" between two actors, one depending on the other to respect the agreement. The agreement can be a goal to be fulfilled, a task to be performed, or a resource to be delivered. In our context, the main interest is on *goals*, that are represented as ovals.

- *Actor diagram.* It is a graph of actors related by dependencies, used to model how actors depend on each other.

- *Rationale diagram.* It is used to represent the logical foundations that rule the relationships between actors. It appears as a balloon within which goals of a specific actor are analyzed and dependencies with other actors are established. Goals are decomposed into subgoals, with either AND (all subgoals must be achieved) or OR (any of the subgoals must be achieved) semantics, possibly specifying the positive/negative contributions of subgoals to goals. The intuitive meaning of a positive (negative) contribution is that the satisfaction of a goal encourages (discourages) the satisfaction of another goal. Notations + and ++ (− and − −) specify the different strength of the contribution.

When analyzing user requirements for DWs, two perspectives should be taken into account. Firstly, it is important to model and analyze the organizational setting in which the DW will operate (*organizational modeling*); this includes designing the actor diagram as well as the rationale diagrams for each stakeholder. Secondly, in order to capture the functional and non-functional requirements of the DW, we need to design rationale diagrams for the decision makers, who are the main actors in the decisional process (*decisional modeling*).

In the following subsections these two perspectives are described in detail, together with the analysis phases they encompass, with reference to real case study, the BI-BANK project, developed at the University of Trento in collaboration with *DeltaDator S.p.a.*. BI-BANK is a project for developing a Banking Business Intelligence System able to support the decisional process with a set of basic banking analyses. For simplicity, in this paper we only focus on banking transaction analysis.

As concerns notation, using Tropos in the DW context requires some new concepts to be introduced:

- *Facts.* In organizational modeling, a fact models a set of events that happen when a goal is achieved. In decisional modeling, a fact is more properly meant as a possible focus of analysis related to an analysis goal. Graphically, facts are represented as rectangles connected to a goal.

- *Attributes.* They are fields whose value is provided when a fact is recorded to fulfill a goal. They are denoted as small diamonds connected to goals.

- *Dimensions.* A dimension is a fact property that describes a possible coordinate of analysis, i.e. a possible perspective for looking at the fact to fulfill an analysis goal. Dimension are represented as small circles connected to goals.

- *Measures.* A measure is a numerical property of a fact that describes a quantitative aspect that is relevant for decision making. Graphically, measures are represented as small squares connected to goals.

Table 1: Notation for actor and rationale diagrams

Symbol	*Meaning*
	actor
	goal
	dependency
	fact
	AND decomposition
	OR decomposition
	attribute
	dimension
	measure

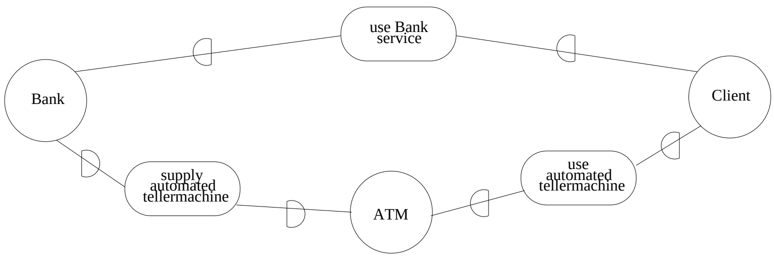

Figure 1: An actor diagram for the BI-BANK case study

The graphical notation is summarized in Table 1.

3.1 Organizational Modeling

Organizational modeling consists of three different phases: (i) *goal analysis*, in which actor and rationale diagrams are produced; (ii) *fact analysis*, in which rationale diagrams are extended with facts; and (iii) *attribute analysis*, in which rationale diagrams are further extended with attributes. Each phase is a different iterative process taking in input the diagrams produced by the previous one.

3.1.1 Goal Analysis

The first step for goal analysis is to represent the relevant stakeholders for the organization and their social dependencies. This is done by means of an actor diagram, in which actors can represent agents, roles, or positions within the organization.

Figure 1 shows a partial actor diagram for the BI-BANK case study. The Client depends on the

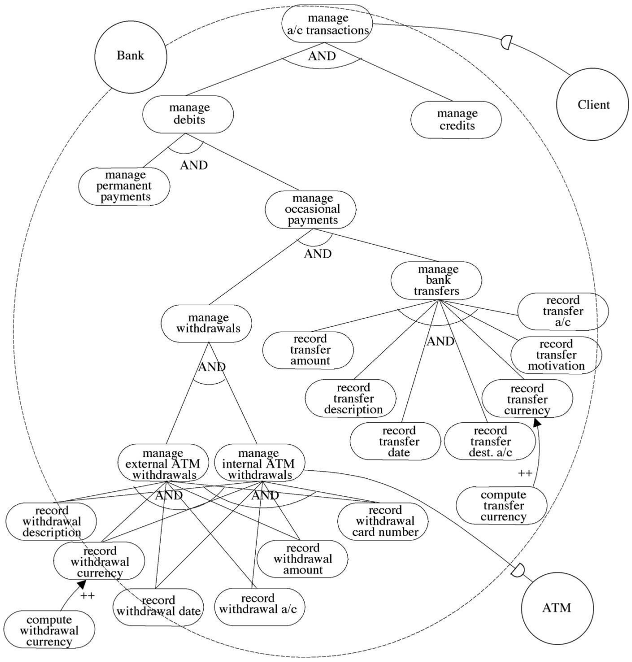

Figure 2: Rationale diagram for the Bank actor from the organizational perspective

Bank for achieving the goal use Bank service, and on the ATM for the goal use automate teller machine. Moreover, the Bank depends on the ATM actor for the goal supply automate teller machine.

The second step consists in analyzing each goal of each actor in more detail to produce a rationale diagram for each actor. Goals are AND-decomposed and contribution links between goals are discovered. See for instance [7, 20] for details on how goal analysis can be carried out. Goal analysis ends when all the relevant goals of each actor have been analyzed and all the dependencies between actors are established.

Figure 2 presents a part of the rationale diagram for the Bank actor focusing on the goal of managing transactions. The goal manage a/c transactions is decomposed into manage debits and manage credits, and in turn manage debits is decomposed into manage permanent payments and manage occasional payments. New dependencies may be discovered at this point, for example the Client depends on the Bank to manage a/c transaction.

3.1.2 Fact Analysis

The objective of fact analysis is to identify all the relevant facts for the organization. The analyst navigates the rationale diagram of each actor and extends it by associating goals with facts that model the set of events to be recorded when goals are achieved.

Figure 3 shows an extended rationale diagram for the Bank actor, still focusing on goal manage a/c transactions (note that the figure also covers attribute analysis, that we will see in detail in next paragraph). The fact transaction is associated to the main goal manage a/c transactions, the fact debit transaction to the goal manage debits, and so on.

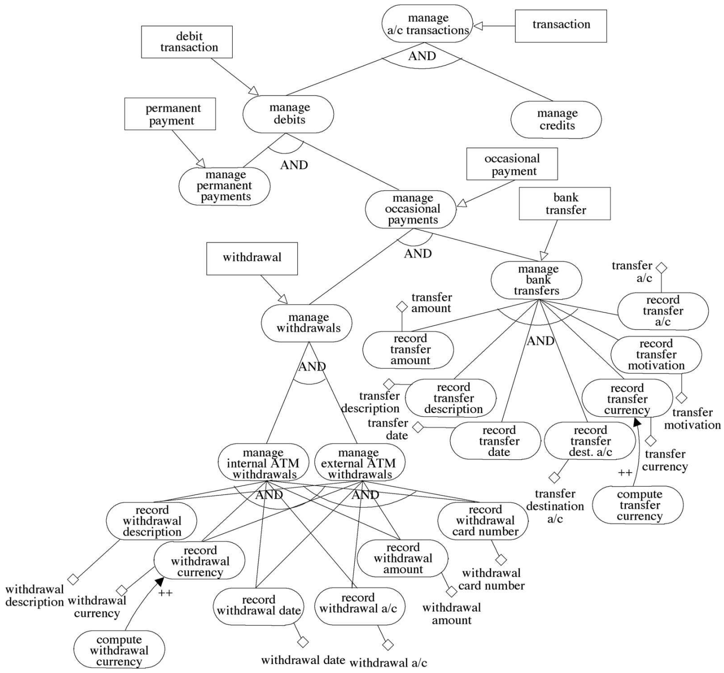

Figure 3: Extended rationale diagram for the Bank actor from the organizational perspective

3.1.3 Attribute Analysis

Attribute analysis is aimed at identifying all the attributes that are given a value when facts are recorded. Starting from the extended rationale diagrams produced in the previous phase, the analyst explores all the subgraphs to associate goals with the attributes they use. Note that, in this phase, the attributes are identified without specifying their possible role as dimensions or measures; from the organizational point of view, attributes are simply data associated to goals.

Figure 3 shows an extended rationale diagram for the Bank actor. For instance, by analyzing the subgraph of the goal manage withdrawals that fact withdrawal is associated with, we introduce attributes withdrawal currency, withdrawal date, withdrawal card number, withdrawal amount, etc.

3.2 Decisional Modeling

After organization modeling, the methodology proposes a second type of analysis focused on the goals of decision makers, i.e., the actors that play the most relevant role in the decisional process. Firstly, all the decision makers are identified; then, for each of them, four steps are carried out: (i) *goal analysis*, that produces rationale diagrams; (ii) *fact analysis*, that extends them with facts; (iii) *dimension analysis*, that further extends them with dimensions; and (iv) *measure analysis*, that further extends them with measures.

3.2.1 Goal Analysis

As for organizational modeling, goal analysis starts by analyzing the actor diagram for the decision makers. Decision makers are identified and initial social dependencies between them are established.

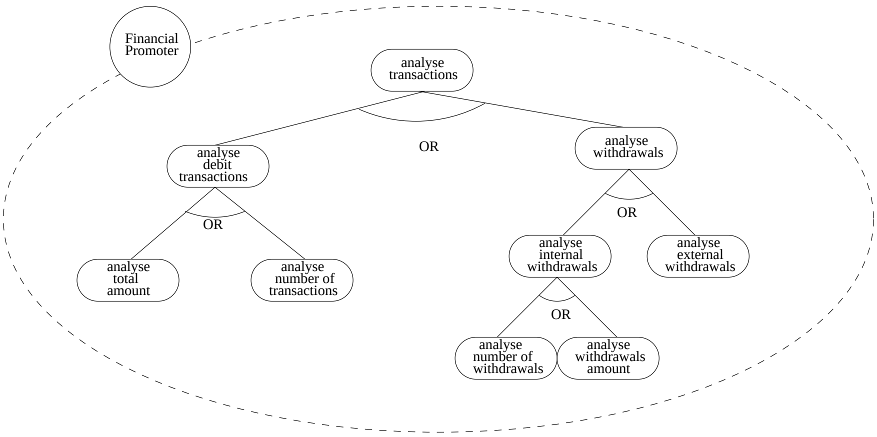

Figure 4: Rationale diagram for the Financial Promoter decision maker from the decisional perspective

The goals associated to each decision maker are then decomposed and analyzed in detail, to produce a set of rationale diagrams. Goals may be completely different from those analyzed during organization modeling, indeed they are part of the decision process and might be not included in the operative process of the organization.

Figure 4 shows a rationale diagram for decision maker Financial Promoter, focusing on the goal of analyzing transactions. The goal analyze transactions is OR-decomposed into analyze debit transactions and analyze withdrawals, which in turn are further decomposed. So, for instance, the goal analyze debit transactions is OR-decomposed into analyze total amount and analyze number of transactions.

3.2.2 Fact Analysis

Like for organization modeling, rationale diagrams are extended by identifying facts and associating them to the goals of decision makers. Facts are possible objects of analysis, and correspond to business events that dynamically happen within the enterprise. Facts are normally imported from the extended rationale diagrams produced during organization modeling. Indeed, very often the goals of decision makers are related to the information produced in the operational process, so the facts associated to the organization activities are fundamental for fulfilling the decision makers' goals. In some cases, the analyst can also introduce some new facts by directly analyzing the decision maker rationale diagrams. For instance, in Figure 5 the analyst associates fact transaction, identified during organizational modeling (see Figure 3), to the goal analyze transactions (the figure also includes dimensions and measures, that we introduce later).

3.2.3 Dimension Analysis

In this phase, each fact is related to the dimensions that decision makers consider necessary in order to satisfy their decisional goals. Dimensions are connected to the goals associated to the fact as shown in Figure 5, where dimensions account number and month are associated to goal analyze total amount.

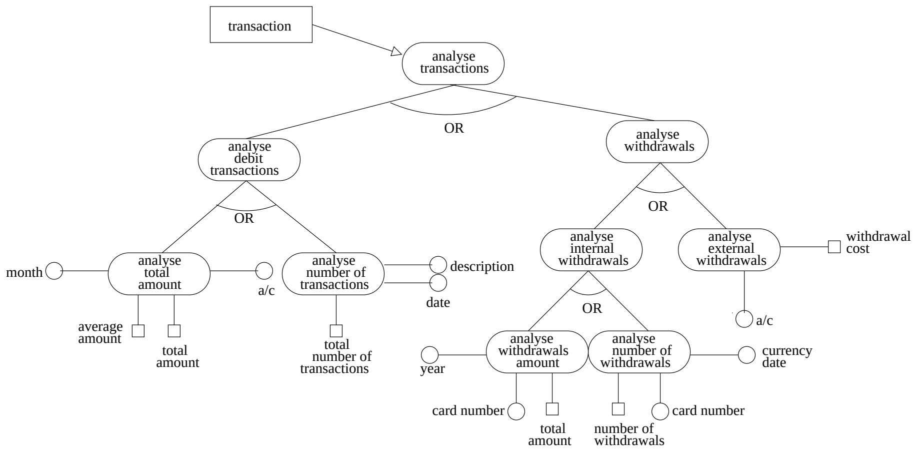

Figure 5: Extended rationale diagram for the Financial Promoter decision maker from the decisional perspective

3.2.4 Measure Analysis

Finally, the analyst associates a set of measures to each fact previously identified. For example, two measures are identified for goal analyze total amount in Figure 4: total amount and average amount.

4 From Requirement Analysis to Conceptual Design

The organizational model we produced by requirement analysis represents the main data on which the enterprise operation is based, thus roughly mapping the most relevant attributes that are presumably part of the source database. On the other hand, the decisional model summarizes the role played, in glossary-based requirement analysis, by the glossaries of facts, dimensions, and measures and by the preliminary workload. In this section we explain how these diagram are used within, respectively, a mixed and a demand-driven design framework.

4.1 Mixed Design Framework

The mixed framework joins the facilities of supply-driven approaches with the guarantees of demand-driven ones. In fact, the requirements derived during organizational and decisional modeling are matched with the schema of the operational database to generate the conceptual schema for the DW. Three phases are involved: (i) *requirement mapping*, where facts, dimensions, and measures identified during decisional modeling are mapped onto entities in the operational schema; (ii) *hierarchy construction*, where a basic conceptual schema is generated by navigating the operational schema; and (iii) *refinement*, where the basic conceptual schema is edited and trimmed to fully meet the user expectations.

4.1.1 Requirement Mapping

During this phase, the facts, dimensions, and measures included in the extended rationale diagrams produced by decisional modeling are mapped, where possible, onto the source schema. More precisely:

1. The facts of decisional modeling are mapped onto entities or n-ary associations (if sources are modeled by an Entity/Relationship schema) or onto relations (if sources are modeled by a relational schema). Thus, in the bank example, fact transaction will be mapped on some TRANSACTIONS table in the source schema.

2. As to dimensions and measures, mapping is achieved by using the attributes represented during organizational modeling as a bridge. A double mapping is established between such attributes and the attributes in the source schema on the one hand, the dimensions and measures in the decisional model on the other. For instance, attribute card number associated to goal record card number in Figure 3 corresponds to dimension card number associated to the analysis goals analyze withdrawals amount and analyze number of withdrawals in Figure 5; the same attribute card number might for instance correspond, on the source schema, to an attribute cardNumber within a WITHDRAWALS table. Similarly, attribute withdrawal amount of goal record withdrawal amount corresponds to measure total amount of the analysis goal analyze withdrawals amount and to attribute amount on the WITHDRAWALS table.

Interestingly, if the names in the extended rationale diagrams are chosen by the analyst consistently with those in the operational schema, this phase can be partially automated. In particular, if the operational schema was actually obtained by normalizing and integrating different sources—which very often is the case, especially when complex cleaning and transformation procedures are necessary to improve data quality—its name space is largely under the designer's control. Otherwise a Thesaurus must be built, as suggested in [1].

4.1.2 Hierarchy Construction

This phase implements the supply-driven part of our approach. For each fact identified during decisional modeling and successfully mapped, the many-to-one associations—expressed in the operational schema by foreign keys—are navigated to build the attribute hierarchies and create a basic conceptual schema, e.g. in the form of a *fact schema*. Fact schemata are the conceptual artifacts provided by the Dimensional Fact Model, proposed in [8] as a support for conceptual design of DWs. Note that any other conceptual model for multidimensional databases could be equivalently adopted.

This phase can be largely automated; the details of the algorithm can be found in [8]. Remarkably, while in the supply-driven approach described in [8] navigation is "blind", meaning that all the attributes connected to the fact by a path of many-to-one associations are reached and included in hierarchies, here navigation is actively biased by the user requirements. In fact:

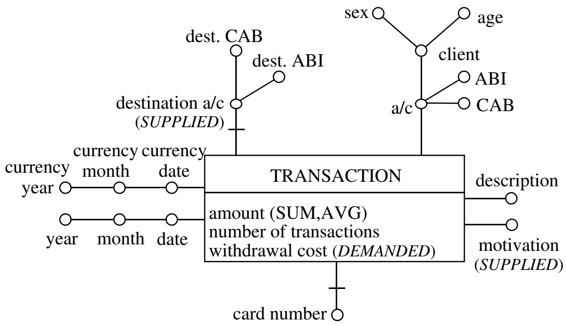

Figure 6: Preliminary fact schema for fact TRANSACTION in a mixed framework

1. Every dimension d successfully mapped from an extended rationale diagram to the operational schema is included in the fact schema, and the full hierarchy rooted in d is generated by navigation.
2. Every measure m successfully mapped from an extended rationale diagram to the operational schema is included in the fact schema, and no hierarchy is generated for it.
3. For each attribute in the organizational model but not in the decisional model, the designer has to decide whether its primary role is that of a dimension or a measure. In both cases, it is included in the fact schema and labeled as "supplied".
4. Among the dimensions and measures on the decisional rationale diagrams, those for which no mapping on the operational schema could be find are nevertheless included in the fact schema and labeled as "demanded".
5. All the attributes in the operational schema that were not mapped and were not reached by navigation of foreign keys starting from a dimension are not included in the fact schema.

The basic fact schemata generated here may be considerably simpler and smaller than those generated in [8]. Besides, while in [8] the analyst is asked for identifying facts, dimensions, and measures directly on the operational schema, here such identification is driven by the diagrams developed during requirement analysis. We also note that the names used for measures in decisional diagrams can give the designer precious suggestions regarding which aggregation operators to use: for instance, from Figure 5 we may presume that measure amount is to be aggregated through both the sum and the average operators.

The preliminary fact schema obtained for fact TRANSACTION in the bank example is reported in Figure 6. Consistently with the Dimensional Fact Model, the fact is represented as a box containing the measures; dimensions are circles connected to the fact; hierarchies are represented as trees rooted in dimensions. Measure withdrawal cost is labeled as "demanded" since it appears as a measure associated to goal analyze external withdrawals but not as an available attribute on the organizational diagram. Dimension motivation is labeled as "supplied" since it is present in organizational diagrams but has not been indicated by decision makers as a dimension in decisional diagrams.

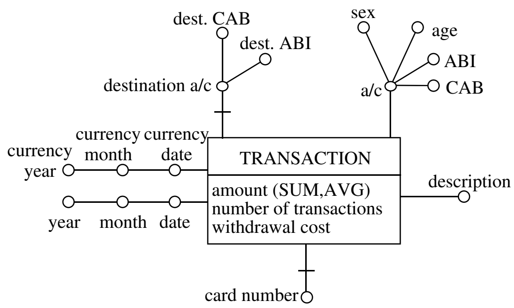

Figure 7: Fact schema for fact TRANSACTION after refinement in a mixed framework

4.1.3 Refinement

This phase is aimed at rearranging the fact schemata in order to better fit them to the users' needs. The basic operations that can be carried out to this purpose are: dropping a node a and all the subtree rooted in a from a hierarchy; dropping a node from a hierarchy while preserving the subtree rooted in a; adding (deleting) a many-to-one association to (from) a hierarchy, which results in changing the parent of a node [8].

Note that, thanks to the labeling of dimensions carried out during hierarchy construction, the decision makers and the analyst are enabled to distinguish, on fact schemata, what is needed and available (unlabeled dimensions/measures), what is needed but unavailable (dimensions/measures labeled as "demanded"), what is available but not needed (dimensions/measures labeled as "supplied"). While the second category may drive the designers in enriching the source database or in considering new data sources, the second may stimulate the decision makers to undertake new directions of analysis.

The final fact schema for fact TRANSACTION is shown in Figure 7. We assumed that users are not interested in the client granularity, that measure withdrawal cost is computed during ETL, that dimensions destination a/c and motivation are considered to be, respectively, relevant and not relevant for analysis.

4.2 Demand-Driven Design Framework

Within a demand-driven framework, in the absence of *a priori* knowledge of the source schema, the building of hierarchies cannot be automated; the main assurance of a satisfactory result is the skill and experience of the designer, and her ability to interact with the domain experts.

The starting point is a set of preliminary fact schemata obtained by associating each fact from decisional rationale diagrams with the corresponding dimensions and measures. In the bank case, from the rationale diagram in Figure 5 we immediately derive the preliminary fact schema in Figure 8. The main issues to be faced afterwords, by strictly interacting with business users, can be summarized as follows:

1. Detect functional dependencies between dimensions and represent them in the form of hierarchy (e.g., $date \rightarrow month \rightarrow year$).

2. Recognize optional dimensions (for instance card number, that is present only for some kinds of

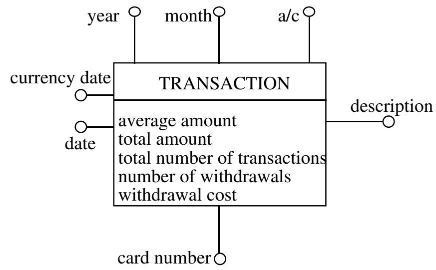

Figure 8: Preliminary fact schema for fact TRANSACTION in a demand-driven framework

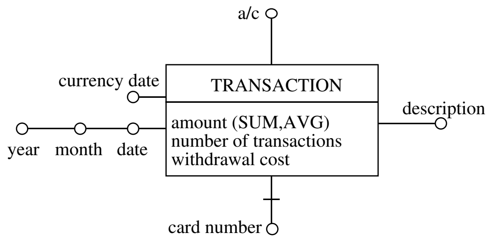

Figure 9: Fact schema for fact TRANSACTION after refinement in a demand-driven framework

transactions).

3. Unify measures that only differ for the aggregation operator (e.g., average amount and total amount).

4. In case of dimensions or measures related to specific subsets of events, either unify them or split the fact (e.g., number of transactions and number of withdrawals are unified since withdrawals are a specific type of transaction).

The fact schema obtained by applying the criteria above is represented in Figure 9.

5 Conclusion

In this paper we have proposed a goal-oriented methodology for requirement analysis in DWs, which can be used within both a demand-driven and a mixed supply/demand-driven design framework. The advantage of our methodology over the existing ones is to ensure that early requirements are properly taken into account—which ensures a "good" design—and, at the same time, that the resulting DW schemata are tightly rooted to the operational database—which makes the design of ETL simpler.

The methodology was applied to the BI-BANK case study, a project developed in collaboration with a company based in Trentino. The experience with the company was extremely useful for refining and validating our approach. We received a positive feedback about the methodology and, in particular, about the importance of deriving the requirements directly from the analysis of the stakeholders and decision makers goals. The case study also supported us in investigating the scalability of our approach. With regard to this, we verified that associating an actor diagram with several rational diagrams, one for each actor, has a crucial role in dealing with complex application domains. In fact, detailed requirement

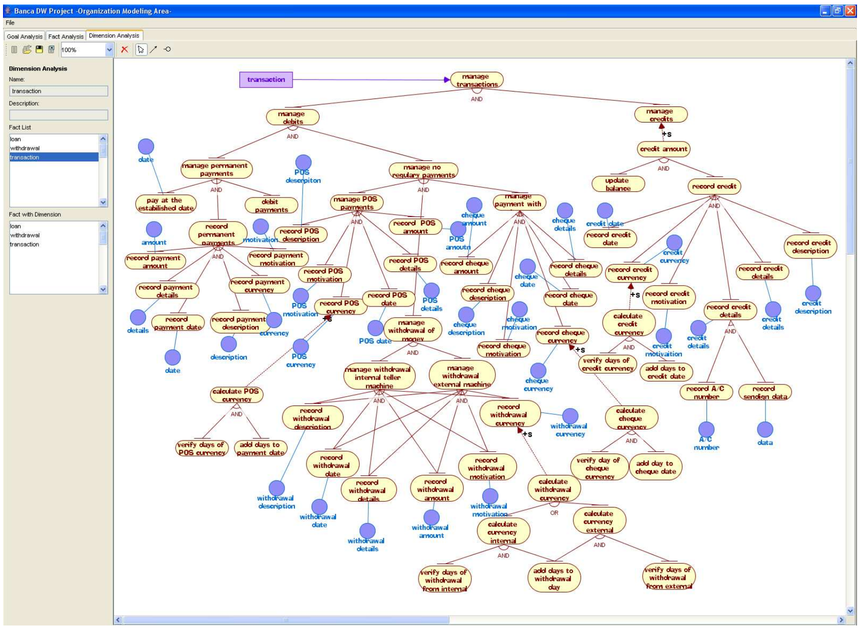

Figure 10: A snapshot from the prototype CASE tool.

analysis is carried out on rationale diagrams, whose complexity depends on how articulated the decisional tasks of a single actor are (which is not directly related to how large the application domain is). On the other hand, the actor diagram—on which the size of the application domain, in terms of number of stakeholders, directly impacts—is drawn at a high level of abstraction, thus its complexity never becomes overwhelming.

Our methodology is fully supported by a CASE tool[1] (Figure 10), which has been developed in java and supports the analyst in the following activities:

- *Organizational modeling.* The analyst is supported with graphical tools to draw actor and rationale diagrams, and performs goal analysis, fact analysis, and attribute analysis.
- *Decisional modeling.* The analyst can draw rationale diagrams for decision makers and extend them with facts, dimensions, and measures, possibly by referring to diagrams produced during organizational modeling.
- *Conceptual design.* Fact schemata are automatically generated from the diagrams elaborated during decisional modeling. The analyst can refine such schemata to generate the final ones.

References

[1] A. Bonifati, F. Cattaneo, S. Ceri, A. Fuggetta, and S. Paraboschi. Designing data marts for data warehouses. *ACM Trans. Softw. Eng. Methodol.*, 10(4):452–483, 2001.

[1]See the Tropos web site (http://www.troposproject.org) for more details.

[2] P. Bresciani, P. Giorgini, F. Giunchiglia, J. Mylopoulos, and A. Perini. Tropos: An agent-oriented software development methodology. *Journal of Autonomous Agents and Multi-Agent Systems*, 8(3):203–236, 2004.

[3] R. Bruckner, B. List, and J. Schiefer. Developing requirements for data warehouse systems with use cases. In *Proc. 7th Americas Conf. on Information Systems*, pages 329–335, 2001.

[4] D. Bulos. Designing OLAP with ADAPT. Technical report, Atos Origin, 1999.

[5] J. Castro, M. Kolp, and J. Mylopoulos. Towards requirements-driven information systems engineering: The Tropos project. *Information Systems*, 27(6):365–389, 2002.

[6] A. Dardenne, A. van Lamsweerde, and S. Fickas. Goal-directed requirements acquisition. *Science of Computer Programming*, 20(1–2):3–50, 1993.

[7] P. Giorgini, E. Nicchiarelli, J. Mylopoulos, and R.Sebastiani. Formal reasoning techniques for goal models. *Journal of Data Semantics*, 1, 2003. Springer.

[8] M. Golfarelli, D. Maio, and S. Rizzi. The dimensional fact model: A conceptual model for data warehouses. *Intl' Journal of Cooperative Information Systems*, 7(2-3):215–247, 1998.

[9] M. Golfarelli and S. Rizzi. A methodological framework for data warehouse design. In *Proc. DOLAP*, pages 3–9, 1998.

[10] M. Golfarelli, S. Rizzi, and E. Saltarelli. WAND: A CASE tool for workload-based design of a data mart. In *Proc. SEBD*, pages 422–426, Portoferraio, Italy, 2002.

[11] B. Hüsemann, J. Lechtenbörger, and G. Vossen. Conceptual data warehouse design. In *Proc. 2nd DMDW*, pages 3–9, Stockholm, Sweden, 2000.

[12] W. H. Inmon. *Building the Data Warehouse*. QED Press/John Wiley, 1992.

[13] R. Kimball, L. Reeves, M. Ross, and W. Thornthwaite. *The Data Warehouse Lifecycle Toolkit*. John Wiley & Sons, 1998.

[14] J. Lechtenbörger. Data warehouse schema design. Technical Report 79, DISDBIS Akademische Verlagsgesellschaft Aka GmbH,, 2001.

[15] B. List, R. Bruckner, K. Machaczek, and J. Schiefer. A comparison of data warehouse development methodologies: Case study of the process warehouse. In *Proc. DEXA*, pages 203–215, 2002.

[16] S. Luján-Mora and J. Trujillo. A comprehensive method for data warehouse design. In *Proc. DMDW*, 2003.

[17] S. Luján-Mora, J. Trujillo, and I. Y. Song. The Gold Model Case Tool: An environment for designing OLAP applications. In *Proc. ICEIS*, pages 699–707, 2002.

[18] D. Moody and M. Kortink. From enterprise models to dimensional models: A methodology for data warehouse and data mart design. In *Proc. 2nd DMDW*, Stockholm, Sweden, 2000.

[19] N. Prakash and A. Gosain. Requirements driven data warehouse development. In *CAiSE Short Paper Proc.*, 2003.

[20] R. Sebastiani, P. Giorgini, and J. Mylopoulos. Simple and minimum-cost satisfiability for goal models. In *Proc. CAiSE'04*, pages 20–33, 2004.

[21] R. Winter and B. Strauch. A method for demand-driven information requirements analysis in data warehousing projects. In *Proc. HICSS*, pages 1359–1365, Hawaii, 2003.

[22] E. Yu. *Modelling Strategic Relationships for Process Reengineering*. PhD thesis, University of Toronto, Department of Computer Science, 1995.