Designing What-if Analysis: Towards a Methodology*

Matteo Golfarelli
University of Bologna - Italy
golfare@csr.unibo.it

Stefano Rizzi
University of Bologna - Italy
srizzi@deis.unibo.it

Andrea Proli
University of Bologna - Italy
aproli@deis.unibo.it

ABSTRACT

In order to be able to evaluate beforehand the impact of a strategical or tactical move, decision makers need reliable previsional systems. What-if analysis satisfies this need by enabling users to simulate and inspect the behavior of a complex system under some given hypotheses, called scenarios. Though a few commercial tools are capable of performing forecasting and what-if analysis, and some papers describe relevant applications in different fields, no attempt has been made so far to comprehensively address methodological and modeling issues in this field. This paper is a preliminary work in the direction of devising a structured approach to designing what-if applications in the BI context. Its goal is to summarize the main lessons we have learnt by facing real what-if projects, and to discuss the related research issues. We also provide a methodological framework for design and discuss its application to a case study.

Categories and Subject Descriptors

H.4.2 [**Information Systems Applications**]: Types of Systems—*Decision support*

General Terms

Design

Keywords

What-if analysis

1. INTRODUCTION

An increasing number of enterprises feel the need for obtaining relevant information about their future business, aimed at planning optimal strategies to reach their goals. In particular, in order to be able to evaluate beforehand the impact of a strategical or tactical move, decision makers need reliable previsional systems. Data warehouses (DWs), that indeed have been playing a lead role within business intelligence (BI) platforms in supporting the decision process over the last decade, are aimed to support detailed analysis of past data, thus they are not capable of giving anticipations of future trends. That's where what-if analysis comes into play.

In a nutshell, *what-if analysis* can be described as a data-intensive simulation whose goal is to inspect the behavior of a complex system (i.e., the enterprise business or a part of it) under some given hypotheses (called *scenarios*). More pragmatically, what-if analysis measures how changes in a set of independent variables impact on a set of dependent variables with reference to a given simulation model [20]; such model is a simplified representation of the business, tuned according to the historical enterprise data. A simple example of what-if query in the marketing domain is: *How would my profits change if I run a* 3×2 *promotion for one week on some products on sale?*

What-if analysis should not be confused with *sensitivity analysis*, aimed at evaluating how sensitive is the behavior of the system to a small change of one or more parameters. Besides, there is an important difference between what-if analysis and simple *forecasting*, widely used especially in the banking and insurance fields. In fact, while forecasting is normally carried out by extrapolating trends out of the historical series stored in information systems, what-if analysis requires to simulate complex phenomena whose effects cannot be simply determined as a projection of past data, which in turn requires to build a simulation model capable of reproducing – with satisfactory approximation – the real behavior of the business. For the same reason, the design of what-if applications is also more complex than that of conventional DWs, which only relies on a static model of business.

Surprisingly, though a few commercial tools are already capable of performing forecasting and what-if analysis, and some papers describe relevant applications in different fields, no attempt has been made so far outside the simulation community to comprehensively address methodological and modeling issues in this field. On the other hand, facing a what-if project without the support of a methodology and of a modeling formalism is very time-consuming, and does not adequately protect the designer and his customers against the risk of failure.

This paper follows from the experience we made on some real what-if projects, and is a preliminary work in the direction of devising a structured approach to designing what-if

*Partially funded by Fondazione Cassa di Risparmio di Cesena and by Onit Group, Cesena, Italy.

DOLAP'06, November 10, 2006, Arlington, Virginia, USA.

applications for BI. Its goal is to summarize the main lessons we have learnt, and to bring the what-if problem to the attention of the BI community in order to pave the way for future research. The remainder of the paper is structured as follows. Section 2 discusses the related literature and summarizes the main features of the commercial tools for what-if analysis. Section 3 presents the beliefs we came to and the related research issues. Section 4 proposes a sketch of the methodology we attained. Section 5 describes a case study and gives some indication about how its main challenges were faced within our methodological framework. Finally, Section 6 draws the conclusions.

2. RELATED LITERATURE AND TOOLS

There are a number of papers related to what-if analysis in the literature. In several cases, they just describe its applications in different fields such as e-commerce [4], hazard analysis [3], spatial databases [14, 16], index selection for relational databases [5]. Other papers, such as [10, 12, 13], are focused on the design of simulation experiments and the validation of simulation models. In [2], the authors survey a set of alternative approaches to forecasting, and give useful guidelines for selecting the best ones according to the availability and reliability of knowledge. In [15] the authors explore the relationships between what-if analysis and multidimensional modeling; though some useful indications are given, no design methodology is proposed.

A separate mention is in order for *system dynamics* [9, 7, 22]. System dynamics is an approach to modeling the behavior of nonlinear systems, in which cause-effect relationships between (aggregate and quantifiable) abstract events are captured as *dependencies* among numerical variables; in general, such dependencies could give rise to retroactive interaction cycles, i.e., feedback loops. From a mathematical standpoint, systems of differential equations are the proper tool for modeling such systems. In the general case, however, a solution cannot always be found analytically, and the dependencies among variables make it very difficult to predict the behavior of the system by adopting the classical, reductionst approach to problem solving; thus, numerical techniques are often used instead. A system dynamics model consists of a set of variables linked together, classified as *stock* and *flow* variables; flow variables represent the rate at which the level of cumulation in stock variables changes. By running simulations on such a model, the user can understand how the system will evolve over time as a consequence of a hypothetical action she takes; she can also observe, at each time step, the values assumed by the model variables and (possibly) modify them.

From what said above, it appears that system dynamics is a good candidate technique to cope with what-if applications in which the current state of any part of the system could influence its own future state through a closed chain of dependency links. On the other hand, though a huge literature about system dynamics has been written over the last four decades, most design-related papers are focused on the validation of system dynamics models (e.g., [21]) and only a few offer valid guidelines for their construction (e.g., [18]).

Due to their strategic importance, forecasting and what-if analysis have also raised a keen interest by vendors. A tool for what-if analysis should at least have the following features:

- Natively support a core set of techniques for expressing and building simulation models, plus a language for further extending the modeling capabilities.
- Support decision makers in formulating hypothetical scenarios on the model.
- Support interactive update of data.
- Allow decision makers to hierarchically aggregate and disaggregate predictions and see the impact of modifications at every level.
- Support statistical techniques for evaluating how reliable and accurate the predictions are.

Though no dedicated what-if platforms are commercially available, some data warehousing or forecasting tools have been extended with what-if features. In the following subsections we overview some of these tools; for space reasons, we will only mention other tools, such as Hyperion Essbase and SymphonyRPM, that present similar characteristics.

2.1 Applix TM1

The Applix TM1 Platform [1] is basically a read-write MOLAP server: data are stored in multidimensional arrays and analyzed through Excel or web clients. Business managers can change some values and recalculate cubes on-the-fly, so they are enabled to immediately view how changes propagate throughout the model. This real-time what-if analysis is made possible by the proprietary memory-based approach adopted by TM1, that allows quick manipulation of vast data sets in main memory, while avoiding to precalculate consolidations as commonly done in other MOLAP tools.

2.2 Powersim Studio

Powersim Studio[1] is one of several tools for system dynamics, and is aimed at simulating discrete dynamic models expressed by systems of differential equations. Powersim is capable of performing statistical analyses on the behavior of such models by repeatedly executing them and evaluating the final states of the system, provided some probabilistic assumptions (specified by the designer) on the value distribution for input variables. Based on well-known statistical techniques, such as the Montecarlo and Latin Hypercube methods, Powersim provides specific functionalities for sensitivity analysis and risk assessment tasks.

Also, Powersim can be seamlessly integrated with the SAP solution for Business Planning and Simulation (see Subsection 2.4): this allows, for instance, to feed a Powersim model with input coming directly from the enterprise DW, and perform what-if analysis over multidimensional data.

2.3 QlikView

QlikView Enterprise[2] is a tool proposed as an alternative to traditional DW-based systems for BI. It is capable of efficiently storing a large amount of data in main memory by means of a non-relational associative structure called *data cloud*, directly fed by operational data sources. QlikView integrates the functions of an environment for developing analysis applications with those of an OLAP interface for accessing and navigating data.

[1] www.powersim.com
[2] www.qliktech.com

Despite the interesting capabilities of analysis offered, which allow users to compose complex queries by interacting with an intuitive representation of data, QlikView does not provide sophisticated support to what-if analysis. Unless external scripts are used to implement complex forecasting models, the only built-in primitive for defining hypothetical scenarios is the computation of variables.

2.4 SAP BPS

SAP Strategic Enterprise Management – Business Planning and Simulation [8] enables the user to make assumptions on the enterprise state or future behavior, as well as to analyze the effects of such assumptions. The working data are modeled as cubes whose measures represent economic accounts, balance items, and so on.

The standard type of analysis supported requires the designer to define a set of rules capable of driving the disaggregation of aggregated measures down to the finest granularity. In this way, the user can first express hypothetical scenarios as a function of macroscopic quantities, and then analyze their impact on the most detailed aspects of the enterprise. Different criteria may be chosen to determine how measures will be disaggregated: for instance, the trivial uniform distribution may be adopted, or an ad hoc driver for proportional disaggregation may be specified, or such driver may be extrapolated from historical data.

2.5 SAS Forecast Server

SAS Forecast Server [17] enables the automatic diagnostics and the statistical forecasting of very large sets of time series. It relies on a wide set of forecasting models that are automatically tested and optimized over the data in order to find out the one that fits at best. Another interesting feature concerns the capability of taking the hierarchical nature of data into account by reconciling the forecasted data at aggregation levels that are different from the one used for forecasting. The gap between forecasting the data represented in time series and simulating a real business model is filled by the Base SAS software, a programming language that provides a rich library of pre-written, ready-to-use integrated procedures aimed at handling many common task including data manipulation and management, information storage and retrieval, statistical analysis, and report writing.

3. LESSONS LEARNT AND OPEN ISSUES

In this section we summarize the main beliefs we came to following our experience on what-if projects, and we outline some related research issues.

3.1 Data Model

Though in principle the outcome of a what-if simulation could be anything, from a single Boolean value to a whole database, we argue that, in the context of BI, the multidimensional model should be taken as the reference. In fact: (i) it is widely recognized to be the most suitable model for supporting information analysis; (ii) it is inherently capable of representing historical trends; (iii) it natively supports fruition of information at different abstraction levels; and (iv) what-if analysis is typically made on top of a DW system, where data are multidimensional. Consistently with this assumption, in the following we will assume that the result of a what-if simulation is a multidimensional cube, which we will call *prediction*.

Decision makers are used to navigating multidimensional data within OLAP sessions, that consist in the sequential application of simple and intuitive OLAP operators, each transforming a cube into another one. Consequently, it is natural for them to ask for extending this paradigm for information fruition also to what-if analysis. This would allow users to mix together navigation of historical data and simulation of future data into a single session of analysis. For instance, one could interactively try different scenarios and compare the predictions, or use the outcome of a simulation as the basis for another simulation. Remarkably, in the same direction, an approach has recently been proposed for integrating OLAP with data mining [6].

This raises an interesting research issue. In fact, OLAP should be extended with a set of new, well-formed operators specifically devised for what-if analysis. An example of such operator could be apportion, which disaggregates a quantitative information down a hierarchy according to some given criterion (*driver*); for instance, a transportation cost forecasted by branch and month could be apportioned by product type proportionally to the quantity shipped for each product type. In addition, efficient techniques for supporting the execution of such operators should be investigated.

3.2 Simulation Model

A what-if application is centered on a *simulation model*, that describes one or more alternative ways to construct a prediction. Each alternative corresponds to a *class of scenarios* required by the users. A class of scenarios declares which ones, among the variables appearing in the simulation model, the user has to value in order to make the model executable. For instance, the class of scenarios for the promotion example in Section 1 includes the type of promotion, its length, and the product category it is applied to.

3.2.1 Expressing vs. Building

To avoid confusion, it is worth to distinguish between the techniques used to *express* the model and those used to *build* it. A simulation model is often expressed by means of equations (as in system dynamics), but it may also be expressed in terms of a set of production rules or through a correlation matrix. A model expressed by equations may then be built for instance by applying some regression technique to the time series describing the past events; conversely, a model expressed by rules may be built by applying some data mining algorithm to the business data, or by directly capturing the relevant rules during an interview with a domain expert. In general, the techniques for building simulation models can be classified into *statistical* and *judgmental* [2]:

- Statistical techniques, such as regression and data mining, derive a model for the system from the behavior it exhibited during a given time period. Their main limitation is that they do not capture the causes of a phenomenon but just its effects; thus, when used on a complex system, they may fail if the past data available are not sufficient to comprehensively describe the system behavior.

- Judgmental techniques, such as conjoint analysis and role playing, are aimed at analyzing and formalizing the cause-effect relationships between those components of the system that rule its overall dynamics. The models of system behavior yielded by judgmental tech-

niques may be more general and accurate than those provided by statistical techniques, but for complex systems it is typically very difficult to obtain them with the required accuracy.

In several cases, the two types of techniques are combined as suggested in [2] to maximize the model reliability.

The first research issue here is to give an effective classification of the different expressivity levels required by different kinds of what-if applications, and to relate it to the techniques to be used for achieving such expressivity. Besides, it would be interesting to study how different techniques can be usefully coupled to further increase their expressivity.

From the design point of view, another crucial issue is to find an adequate formalism to conceptually express the simulation model, so that it can be discussed and agreed upon with the users. Unfortunately, no suggestion to this end is given in the literature, and commercial tools do not offer any general modeling support. On the other hand, developers of what-if applications complain for the lack of a semi-formal language to facilitate the transition from the requirements informally expressed by users to their implementation on the chosen platform. A suitable formalism should cover and integrate static, functional, and dynamic aspects. Major emphasis will typically be given to functional aspects, that describe how data are transformed and derived during simulation. Dynamic modeling may be required to describe application domains where time has a critical role in determining the cause-effect relationships between the variables involved in simulations. As to static aspects, as argued in Subsection 3.1, the reference is the multidimensional model, used to describe both the source historical data and the prediction. Though UML could be used in principle, since it is potentially capable of covering all three aspects, we believe that an ad hoc, specific formalism should be devised instead.

3.2.2 Variables and Dependencies

The simulation model defines the nature of *dependencies* among variables, i.e., how to compute the value of a dependent variable, provided that all of the variables it depends on have been valued. Typically, in the BI context, numerical variables are multidimensional and are linked to measures of the input (or the prediction) cube. For instance, a dependency could be enforced between sold quantity by month, branch and customer of product A, and sold quantity by month, branch and customer of product B, so that the overall amount of sold items does not exceed a given threshold: this could be meant to reproduce the behavior of a real setting, where selling more of a newer product could negatively influence the sales of older ones (*cannibalization*).

Dependencies among variables can be classified into two categories: *constraint* dependencies and *temporal* dependencies. Constraint dependencies are enforced at every time instant, and define the legal states of the simulated system: a straightforward example of constraint dependencies is given by formulae that define derived measures, like *amount* = *quantity* × *unit price*. On the other hand, temporal dependencies state how the value of variable v at time instant t influences the value of variables $v_1, v_2, \ldots, v_n$ at time instant $t + k$: for example, selling more of a product in February due to a special sales promotion could have an impact on the amount of sold items for the same product in March (supposed that the sales promotion no longer holds).

A relevant research issue concerning dependencies is how to keep them consistent with each another. In fact, temporal dependencies should not bring the system into a state which violates constraint dependencies, and constraint dependencies should not be conflicting with one another (or, a policy for solving conflicts should be stated). This is not trivial, since dependencies could link variables with different granularities, and a single variable could be involved in more than one dependency. Thus, some effective technique should be devised in order to efficiently detect (and possibly solve) dependency conflicts.

3.2.3 Simulation Granularity

A crucial design issue for developing a reliable simulation model is to address the trade-off between precision and complexity. A very precise and fine-grained model could give rise to high simulation costs, while a lightweight simulation engine could be too simplicistic to be reliable. A careful choice is required to meet both requirements at an acceptable degree: eventually, this process could unravel the need for new information requirements.

Two main research issues arise at this point. Firstly, we argue that what the designer actually needs, in order to determine the optimal resolution to express the simulation model, is a way to estimate the loss of precision that is introduced when modeling low-level phenomena with higher-level dependencies. Though ad hoc, statistical techniques may be applied when a particular formalism and/or methodology is chosen to express and build the simulation model, we believe that an investigation is worth aimed at establishing a general framework for evaluating the simulation error.

A second relevant problem arises from the fact that modeling the behavior of a complex system may require to adopt multiple perspectives in order to properly capture the rules, entities, and interactions that shape its temporal evolution. Indeed, different parts of the business processes and events could be better modelled at different granularities: as long as the domains of such models are mutually disjoint, integrating them simply amounts to aggregating or disaggregating a representation of the system in order to translate between different levels of granularity; however, in case the *same* phenomenon is modeled at more than one abstraction level, how to maintain the consistency between multiple, concurrent simulation models becomes a key issue [19].

4. A METHODOLOGICAL SKETCH

As summarized in Figure 1, the methodology we adopted for the case study relies upon the seven phases sketched in the following:

1. *Goal analysis.* This phase is aimed at determining which business phenomena are to be simulated, and how they will be characterized. More precisely, the goals of analysis are expressed by (i) identifying the set of business variables the user wants to monitor and their granularity; and (ii) defining the relevant classes of scenarios in terms of business variables the user wants to control and other additional parameters – such as the temporal width of the simulation window.

2. *Business modeling.* A draft model of the application domain is built, to the extent suggested by the requirements expressed during phase ♯1. In general, three submodels will be included: (i) one to statically repre-

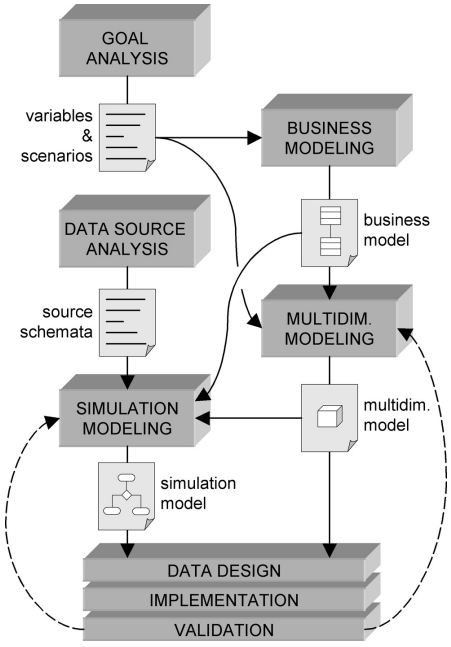

Figure 1: Methodological sketch for what-if design

sent the main entities involved in the business phenomenon and their associations; (ii) one to express how business variables are functionally derived on each other; and (iii) one to describe the dynamic interactions between the entities involved. Overall, this phase should help the designer to understand the business phenomenon as well as give her some preliminary indications about which aspects can be either neglected or simplified for simulation. A set of standard UML diagrams can be used here, e.g., a class diagram for (i), an activity diagram for (ii), and a sequence or state diagram for (iii).

3. *Data source analysis.* The relevant data sources are carefully analyzed, in order to understand what information is available to drive the simulation and how it is structured. Specific attention should be devoted to evaluate the quality of each data source, which significantly impacts on the actual applicability of the simulation model to be built.

4. *Multidimensional modeling.* The multidimensional schema describing the prediction is built, taking into account the static part of the business model produced at phase ♯2 and respecting the requirements expressed at phase ♯1. In particular, the requirement concerning granularity is crucial for defining the dimensions of the prediction cube, which in turn will determine the maximum detail for analyzing the prediction. Any formalism for conceptual/logical modeling of multidimensional databases can be effectively adopted in this phase.

5. *Simulation modeling.* This is the core phase of design. Its aim is to build, based on the business model, the functional/dynamic model allowing the prediction to be constructed, for each given scenario, from the source data available. The most crucial issue the designer has to face during this phase is the achievement of a good compromise between the level of precision of the simulation model and its complexity.

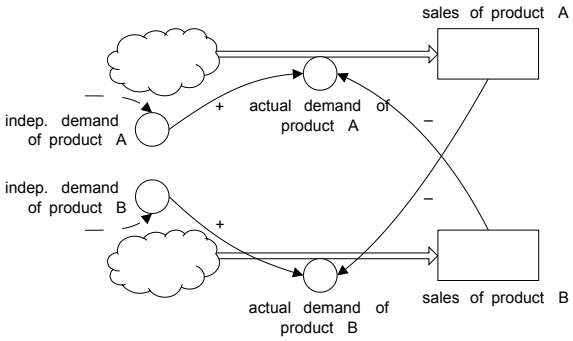

Figure 2: A simple dynamic model of cannibalization

6. *Data design and implementation.* The multidimensional schema (phase ♯4) and the simulation model (phase ♯5) are implemented on the chosen platform, to create a prototype for testing.

7. *Validation.* During this last phase the designer evaluates, together with the users, how faithful the simulation model is to the real business model. The simplest approach to validation consists in running the simulation on a past period and comparing the prediction obtained with the actual values recorded. If the approximation introduced by the simulation model is considered to be unacceptable, phases ♯4 and ♯5 should be iterated to produce a new prototype.

5. A CASE STUDY

Orogel S.p.A. is a large Italian company in the area of deep-frozen food. It has a number of branches scattered on the national territory, each typically entrusted with selling and/or distribution of products. Its information system includes a DW composed of a number of data marts, one of which dedicated to commercial analysis. In the remainder of this section we briefly discuss, phase by phase, the main issues emerged by applying the methodological framework proposed in Section 4 to the Orogel case study.

1. *Goal analysis.* The managers of Orogel are willing to carry out an in-depth analysis on the *profitability* of branches. More precisely, they wish to know if, and to what extent, it is convenient for a given branch to invest on either selling or distribution, with particular regard to the possibility of taking new customers and/or new products. Thus, the two classes of hypothetical scenarios chosen for prototyping are: (i) analyze profitability during next n months in case one or more new products were taken/dropped by a branch; and (ii) analyze profitability during next n months in case one or more new customers were taken/dropped by a branch. Decision makers ask for analyzing profitability at different levels of detail; the finest granularity required is essentially characterized by month, product, customer, and branch.

2. *Business modeling.* The static model of the domain was formalized as a UML class diagram, and is not reported here for brevity. From the functional point of view, we just put together a glossary explaining how business variables are derived, for instance:

Profitability is defined as the difference between revenues and costs.
The *revenue* over a given time period t and for a given product/customer/branch is calculated as the sum of the gross amounts of the invoices issued during t for that product/customer/branch.
The *gross amount* of an invoice is defined as quantity times unit price.
The costs are distinguished into *variable costs* and *fixed costs*: the first are proportional to the quantity sold (e.g., the transportation cost), the second do not depend on it, at least approximatively (e.g., the cost for renting the buildings used to store food does not depend on the financial turnover, at least till the turnover becomes so big to require a new building to be rented).

From the dynamic point of view, we analyzed the phenomena that characterize the sale domain. One of the most influential is the so-called *cannibalization*, defined as the process by which a new product gains sales by diverting sales from existing products, which may deeply impact the overall profitability [23]. Although cannibalization is well-known in the simulation and economic literature, its effective modeling and measurement are still open problems. A simplified dynamic model of cannibalization is shown in Figure 2: consistently with the graphical formalism commonly used in system dynamics, clouds represent infinite sources of cumulable amounts of matter (sold items, in our case), rectangles represent stock variables, circles represent either flow variables (in case they are attached to a double line, connecting the source and the target of the flow) or auxiliary variables, and single lines trace dependencies among variables (see also Section 2). Dependencies are labelled by a '+' or '−' mark, according to the kind of contribution (positive or negative) through which the independent variable affects the value of the dependent one. Instead of providing a complete model for cannibalization, Figure 2 actually shows a template for capturing such phenomenon, which is in general very complex, potentially involving more than just two products, and where the quantification of the influences among variables could depend on many domain-specific or temporal factors.

3. *Data source analysis.* The commercial data mart of Orogel is centered on a SALE cube whose conceptual schema is sketched in Figure 3 according to the Dimensional Fact Model [11]. To give an idea of the cube size, we report that the cardinalities of the main dimensions branch, product, and customer are, respectively, about 20, 6000, and 32 000; 5 years are stored, each yielding about 1 700 000 sale events. The average quality of data is good, since the DW is fed by a reliable ETL system. Remarkably, the business modeling phase revealed that some details in the accounting methods have changed over the last few years, which required an ad hoc ETL procedure to be set up in order to actualize the historical costs thus enabling a consistent forecast.

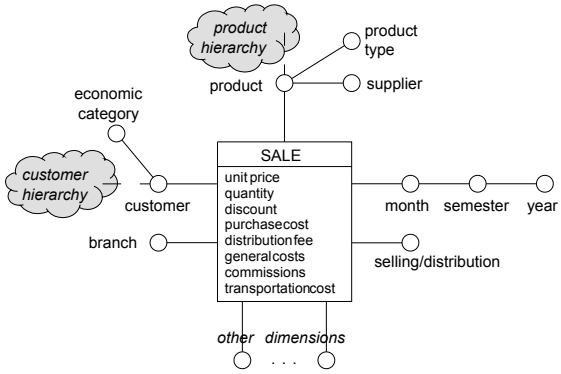

Figure 3: Simplified conceptual schema for the SALE cube

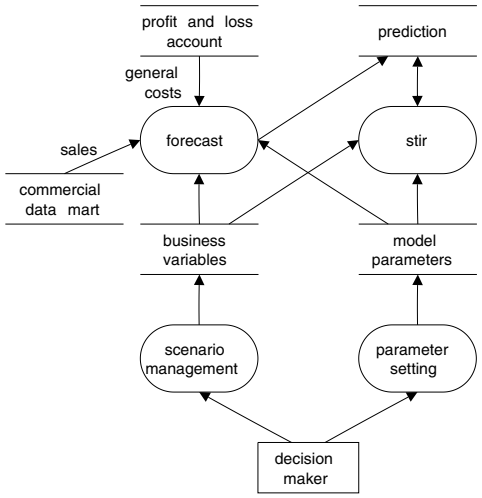

Figure 4: Functional view of the simulation model

4. *Multidimensional modeling.* The requirements expressed by decision makers led us to replicate the multidimensional schema of the source SALE cube, with the addition of a scenario dimension to allow for multiple, alternative scenarios to be generated and compared.

5. *Simulation modeling.* The main issue here was to achieve a good compromise between reliability and complexity. To this end, in constructing the simulation model we adopted a two-step approach that consists in first forecasting past data, then "stirring" the forecasted data according to the events (new product and/or new customer) expressed by the scenarios. With reference to Figure 4, that shows a global functional view of this process through a Data Flow Diagram, we remark that (i) besides the commercial data mart, also the profit and loss account is used as a source for forecasting the general costs; and (ii) a conceptual separation is enforced between business variables (such as the type of the new product) and model parameters (such as the length of the past period taken as a reference for regression), both valued by the decision maker. Considering the good quality of data sources and the complexity of the underlying

Table 1: Forecast granularities for measures

	branch	sell./distr.	supplier	econ. cat.	prod. type	cost elem.
unit price				×		
quantity	×	×		×		
discount	×	×		×		
purchase cost			×		×	
distr. fee	×		×			
general costs	×					×
commissions	×	×				
transp. cost						

phenomena, we mainly adopted statistical techniques for both the forecasting and the stirring steps. In particular, linear regression is employed to forecast unit prices, quantities, costs, percentage discounts and sales commissions starting from a past period taken as a reference. At this stage, the decision maker may provide a bias to express her belief of a peculiar trend for the future. Based on the decision makers' experience, and aimed at avoiding irrelevant statistical fluctuations while capturing significant trends, we adopted different granularities for forecasting the different measures of the prediction cube (see Table 1). Note that an exact computation of discounts and sales commissions would require a complex procedure that operates on single invoices, which cannot be applied at a monthly granularity. In order to model cannibalization we adopted a simple solution based on a matrix that expresses the correlation between the quantities sold for each couple of products of the same type. The correlation matrix is built by judgmental techniques, also taking into account the similarity between the products, that depends on a set of shared characteristics such as the package size and the price segment. Remarkably, this solution also enables the representation of the positive correlations arising when a new product makes one or more other products more attractive for consumers since it completes a product range. Finally, as concerns stirring, the effects of adding a new customer (product) of a given economic category (product type) is simulated by reproducing the sales events related to a representative customer (product) of the same category (type) in the same branch.

6. *Data design and implementation.* Oracle 9i is the platform chosen for hosting the predictions and as a repository for business variables and model parameters. Business Objects is used for OLAP analysis of predictions. The prototype for simulation has been totally implemented in C♯. A screenshot of the GUI used to input business variables is reported in Figure 5; in particular, the form used to formulate hypotheses about the trend of the unit price over the next months is shown. As concerns its engineering, we are currently evaluating to adopt SAS Forecast Server, that seems to fit all the requirements of this case study.

7. *Validation.* We run a first round of validation by using 2003 and 2004 data to plainly forecast the profitability for 2005. A comparison with the actual data for 2005 yielded an average error of about 18% on the total profitability of the single branches, which decision makers judged to be very promising. The error on the total profitability for 2005 is significantly lower (about 7%) due to a compensation effect. Currently, the system is being tested by the decision makers to verify the soundness of the approximations made.

6. CONCLUSIONS

What-if analysis has been promoted by software vendors during the last decade and has been perceived by users in the field of BI as one of the most common applications on-top of DW systems; nevertheless, the number of mature projects is surprisingly low. Several factors contribute to this:

1. *Immature technology.* Despite the marketing claims, until now only few tools offered what-if capabilities, and usually they were limited to a specific application.

2. *Complexity of design.* Designing what-if applications requires to understand, simplify, and model business-related phenomena, which may become very difficult in complex enterprises. Besides, the effort for proving the reliability of the simulation model often does not counterbalance the costs for defining and implementing it. This sometimes discouraged decision makers from undertaking what-if projects.

3. *Lack of a design methodology.* The problem raised above is worsened by the adoption of naive approaches that make projects more expansive and expose them to higher risk of failure.

The new generation of analytic tools are now compensating for item ♯1. In some cases, the hurdle to developing the application from scratch can be overcome by relying on pre-configured models (e.g., SAP-SEM is based on the business models captured by its ERP), thus reducing the impact of ♯2. As to ♯3, a wider spread of what-if analysis within the BI context necessarily requires the development of a well-structured design methodology, possibly supported by an ad-hoc formal model capable of properly emphasizing the representation of the crucial information. Indeed, the adoption of software engineering techniques will significantly reduce the cost and time for delivering standard solutions to recurrent problems.

This paper is a first step towards providing a design methodology, and is based on the lessons learnt during our experience on real projects. Our future efforts in this direction will be aimed at (i) refining and further testing the methodology; (ii) devising a formal model for representing simulations; (iii) extending OLAP with a set of operators for what-if analysis.

Acknowledgment

We would like to warmly thank the administrative staff of Orogel for providing the requirements, and Lorenzo Baldacci, Vladimiro Buda, and Luca Girotti for their precious support to the prototype implementation.

7. REFERENCES

[1] Applix Inc. The TM1 platform: Taking business performance management to a new dimension. http://www.applix.com, 2005.

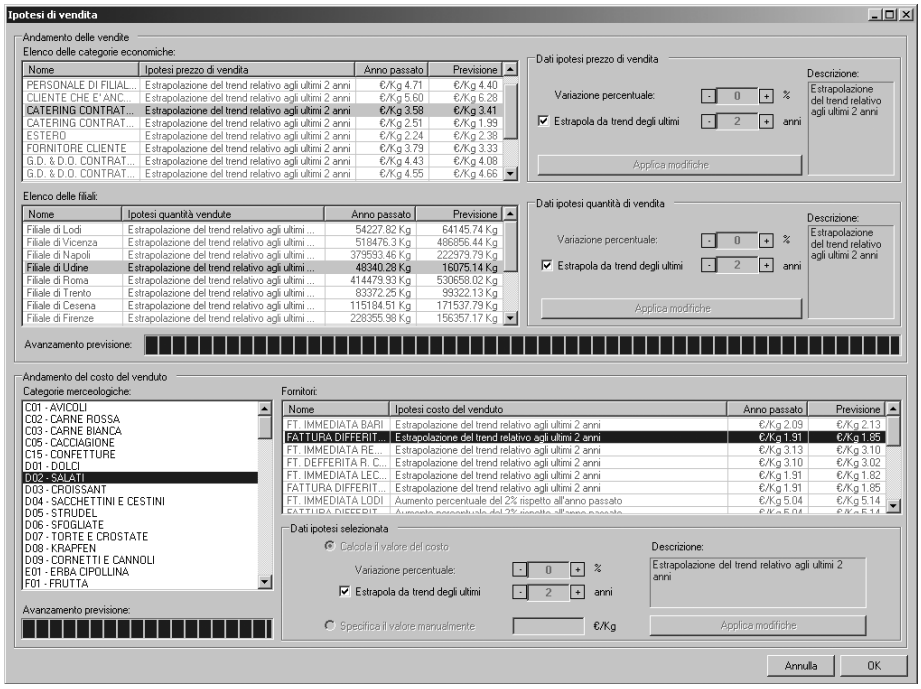

Figure 5: Screenshot of the GUI for scenario management

[2] S. Armstrong and R. Brodie. Forecasting for marketing. In G. Hooley and M. Hussey, editors, *Quantitative methods in marketing*, pages 92–119. Int. Thompson Business Press, 1999.

[3] P. Baybutt. Major hazards analysis – an improved process hazard analysis method. *Process Safety Progress*, 22(1):21–26, 2003.

[4] H. K. Bhargava, R. Krishnan, and R. Müller. Electronic commerce in decision technologies: A business cycle analysis. *Int. Jour. of Electronic Commerce*, 1(4):109–127, 1997.

[5] S. Chaudhuri and V. Narasayya. Autoadmin what-if index analysis utility. *SIGMOD Rec.*, 27(2):367–378, 1998.

[6] B. Chen, L. Chen, Y. Lin, and R. Ramakrishnan. Prediction cubes. In *Int. Conf. on Very Large Data Bases*, pages 982–993, 2005.

[7] R. G. Coyle. *System Dynamics Modelling: A Practical Approach*. Chapman & Hall, London, 1996.

[8] R. Fischer. *Business Planning With SAP SEM*. SAP Press, 2005.

[9] J. W. Forrester. *Industrial Dynamics*. MIT Press, Cambridge, MA, 1961.

[10] C. Fossett, D. Harrison, and H. Weintrob. An assessment procedure for simulation models: a case study. *Operations Research*, 39(5):710–723, 1991.

[11] M. Golfarelli, D. Maio, and S. Rizzi. The Dimensional Fact Model: A conceptual model for data warehouses. *Int. Jour. of Cooperative Inf. Syst.*, 7(2-3):215–247, 1998.

[12] J. Kleijnen. Sensitivity analysis and optimization in simulation models: design of experiments and case studies. In *Proc. Winter Simulation Conf.*, 1995.

[13] J. Kleijnen, S. Sanchez, T. Lucas, and T. Cioppa. State-of-the-art review: A user's guide to the brave new world of designing simulation experiments. *INFORMS Jour. on Computing*, 17(3):263–289, 2005.

[14] R. Klosterman. The What if? collaborative support system. *Environment and Planning, B: Planning and Design*, 26:393–408, 1999.

[15] N.-S. Koutsoukis, G. Mitra, and C. Lucas. Adapting on-line analytical processing for decision modelling: the interaction of information and decision technologies. *Decis. Support Syst.*, 26(1):1–30, 1999.

[16] I. Lee and M. Gahegan. What-if analysis for point data sets using generalised Voronoi diagrams. In *Proc. Int. Conf. on GeoComputation*, Greenwich, UK, 2000.

[17] M. Leonard. Large-scale automatic forecasting with inputs and calendar events. In *Proc. Workshop on Temporal Data Mining*, San Francisco, US, 2001.

[18] L. F. Luna-Reyes and D. L. Andersen. Collecting and analyzing qualitative data for system dynamics: methods and models. *System Dynamics Review*, 19(4):271–296, 2004.

[19] A. Natrajan. *Consistency Maintenance in Concurrent Representations*. PhD thesis, Dep. Computer Science, Univ. of Virginia, 2000.

[20] A. Philippakis. Structured what if analysis in DSS models. In *Proc. HICSS*, pages 366–370, 1988.

[21] H. Qudrat-Ullah. Structural validation of system dynamics and agent-based simulation models. In *Proc. European Conf. on Modelling and Simulation*, 2005.

[22] E. B. Roberts. *Managerial Applications of System Dynamics*. Pegasus Communications, 1999.

[23] S. R. Srinivasan, S. Ramakrishnan, and S. E. Grasman. Incorporating cannibalization models into demand forecasting. *Marketing Intelligence & Planning*, 23(5):470–485, 2005.