

Simulation of success stories into strategic corporate planning

PD Dr. Bernd Viehweger
Humboldt-Universität zu Berlin
Institut für Wirtschaftsinformatik
Spandauer Str. 1
10178 Berlin (Germany)
+49-30-20935716
bv@wiwi.hu-berlin.de

Prof. Dr. Angelika Kreitel
Fachhochschule Würzburg-Schweinfurt
Fachbereich Betriebswirtschaft
Münzstr. 12
97070 Würzburg (Germany)
+49-931-3511192
dr.wak@gmx.de

Dr. Johannes Palme
Institut für Terminologie und angewandte Wissensforschung (itaw)
Am Köllnischen Park 6/7
10179 Berlin (Germany)
+49-30-30862088
jop@itaw.hu-berlin.de

Keywords: Simulation, Management Information Systems, ERP-Systems, XML-Interface, System Dynamics Modeling, Strategy

Abstract:

Modern Management Information Systems make direct use of data from applications implemented in different functional areas of enterprises. Such applications could be ERP-Systems as well as legacy systems in functional areas such as a production planning application, an accounting system, a marketing and customer relationship system etc. These sources will be used from Management Information Systems as a consolidated basis for corporate planning and corporate reporting. Until today a simulation of strategic corporate forecast has been possible on a consolidated data basis.

By using a XML-Interface a direct access to different web-based data of functional systems in a client server environment will be created. This will open the way to use the System Dynamics Methodology to simulate a forecast for product development and their challenges at the market.

The main target of this simulation is to make transparent the efficiency of marketing measures and of research and development expenses.

Approach to the problem

In business management software packages like SAP SEM (Strategic Enterprise Management from the company SAP, Germany) use has been made of System Dynamics Modeling although until now only in a very few cases. Alongside knowledge of how to deal with System Dynamics Modeling (here: simulation software *Powersim* from the company Ventana), the acquisition of real data is also a problem. The software SAP SEM uses among other things a Business Warehouse (BW) which corresponds to a Datawarehouse in terms of how it is set up / structured to save the data. As well as SAP SEM other business management applications are conceivable which relate to the combination of e-commerce techniques und System Dynamics Modeling. Through this data exchange or access to data deserves particular attention.

System Dynamics Software like *Vensim* for example offer data export in different spreadsheet formats (.xls Excel format, .wk1 Lotus123 format) or data format (.dat Data format, tab Tab delimited). Until now the possibility of accessing existing relational data bases which could be made available for example also on web servers and of using the data which is continuously up-dated directly within the simulation software did not exist. Manufacturers of databases usually offer programming interfaces to gain access to SQL-databases from one program like for example ODBC (Open Database Connectivity). In the case of the programming language Java, access to relational databases is made possible through using JDBC (Java Database Connectivity) and CCI-API (Common Client Interface – Application Programming Interface).

Current business management data which are relevant to certain System Dynamics Modeling tasks for external parameters, for example for ERP-based tasks (Enterprise Resource Planning) could be located in different management information systems or databases. Preparing data in XML-Documents is not only perfectly conceivable but can also be very helpful. All of the parameters and constants in a System Dynamics Model can be concentrated into one source or seen more realistically, they can be distributed over several sources. In what follows, a procedure is introduced showing how, using Java of a developed module, current data from a database with XML-technology can be made available (incorporated into) to a System Dynamics Model (using *Vensim*): a DB-XML-SD-interface.¹

Acquisition of data

The simulation software *Vensim* saves models either in a propriety format or in a directly readable (legible) variant, a .mdl-file. In what follows, only the .mdl-files which are arranged / planned specially for *Vensim*-models are looked at. These .mdl-files consist mainly of two parts: the first part contains model equations and the second part carries the Sketch-block in which coded information for the model is located. The second part will be taken into account for further consideration. It will be taken on as it is, i.e unchanged. Contained in the equations in the first part are those which bring in constant

¹ In the poster session of the conference, the procedure will be demonstrated with the aid of a simple example, the complexity of which can be intensified as required and as much as participants want.

values as parameters in the model. At this point, getting hold of current values from the available data stock is considered to be useful (a good idea). It is necessary to isolate these equations – simple value allocations – from others by keeping the structure of the model and to acquire the values for these parameters from an XML-document or directly from databases. These values are then re-incorporated as equations into the model. To make sure of the validity of the exchange data, it is sensible to fix its structure in a formal specification. That can either be in a DTD or with a XML-scheme, whereby the latter is preferable because it provides more possibilities to print out. This formal specification can be used to check by machine the correctness of the data exchange. This check can either be in the system's test phase or during the exchange itself. The more different exchange partners are involved and the more complex the exchanged data the more important is the checking. The task therefore is to fix and think about the formal demands on the exchange data, from when and where it is useful to check the validity of the data with regard to the scheme.

Modeler's tasks

The System Dynamics Modeler can base his model as usual on the simulation software *Vensim*. Plausible values can first be inputted for the externally generated parameters. Finally, the model is saved as a .mdl-file. Afterwards the DB-XML-SD-interface software is started by the modeler which takes over the work of analysing the model, the acquisition of data and the putting into use of the data to be investigated. Finally, the software starts the program *Vensim* with the up-dated model and the user can now start the simulation run. The user can choose from several .mdl-files via a GUI (Graphical User Interface). Afterwards, the previously found external parameters are presented (tabled) together with the values found for them via a matching table. With this, it is irrelevant whether the DB-XML-SD-interface software acquired the values directly from an XML-document or in a roundabout way via / through a database. In the case of a value not being able to be found despite looking into the existing data sources available, the user is requested to enter the value him- / herself. Generally, the matching table offers the user again the possibility of accessing the data manually. After clearance of the data through the user, *Vensim* can then start the model.

Structure of the DB-XML-SD-interface

The software is fully portable written in Java (script) and consists of two core modules: a module A deals with the preparation of database query results in an XML-document. It can be used for database queries generally, so also in the context outlined here and beyond. Module B breaks down and analyses the available model in a .mdl-file, acquires data from the XML-Document or via Module A and puts the model correctly together again in the .mdl-file. These modules are controlled by the GUI-module.

Functionality of the modules

The modules are programmed in Java as classes. Module A is able to carry out database enquiries via JDBC. It can also accept – keeping to a certain structure - the enquiry in a

SQL-statement, as dissected parameters or in a XML-document. The results are available in a field or in an XML-document. Module B can take apart .mdl-files, identify external parameters and look for these parameters in an XML-document and / or in a database.

