Association for Information Systems
AIS Electronic Library (AISeL)

AMCIS 2008 Proceedings | Americas Conference on Information Systems (AMCIS)

2008

Combining System Dynamics and Multidimensional Modelling - A Metamodel Based Approach

Lars Burmester
Philipps-University Marburg, lars.byrmester@wiwi.uni-marburg.de

Matthias Goeken
Frankfurt School of Finance & Management, m.goeken@frankfurt-schools.de

Combining System Dynamics and Multidimensional Modelling – A Metamodel Based Approach

Lars Burmester
Philipps-University Marburg
Lars.Burmester@wiwi.uni-marburg.de

Matthias Goeken
Frankfurt School of Finance & Management
M.Goeken@frankfurt-school.de

ABSTRACT

Online analytical processing (OLAP) as a modern business intelligence (BI) concept provides support for representing vast amounts of data for supporting management's decisions. Though, there is no inherent support for the representation of causal structures which could provide a foundation for advanced analysis like what-if or scenario techniques. System Dynamics (SD) is an approach with a long tradition used for modelling and simulation of complex systems, which could provide a causal complement for OLAP. This paper aims at integrating OLAP and SD on a linguistic level. Therefore linguistic metamodels of the corresponding conceptual modelling languages are derived and related towards each other, creating a translational relationship between the languages.

Keywords

Language, Metamodelling, Conceptual Modelling, Multidimensional Modelling, System Dynamics.

INTRODUCTION

Data warehouse systems, storing data in support of management's decisions, and OLAP systems processing this data to multidimensional information, constitute core elements of state of the art BI solutions. With using the concepts of hierarchisation and multidimensionality, these systems take advantage of two mechanisms to provide their supportive functionality. Hierarchisation is a principle for the reduction of complexity of the real world (see Simon, 1996). Therefore, hierarchisation is widely applied in context of BI systems (esp. OLAP), allowing their users to handle the complexity caused by vast amounts of available data. A multidimensional approach, accompanied by the complexity adaption mechanism of hierarchisation, allows users to visualise a comprehensive picture of business objects, taking into account various perspectives on them. A side effect of organising informational elements into hierarchies is that information about causal relationships existing between these elements vanishes. Hence, the complementally use of a concept dedicated to modelling and simulating causal structures should be proposed to compensate this loss of information.

System Dynamics is an approach for modelling and simulation of complex and dynamic (socio-economical) systems, with a long tradition reaching back to the beginning of the 1960s (e.g. Forrester, 1964). Characteristic for SD is the emphasis on closed cause and effect chains between system elements which often lead to a counterintuitive behaviour of the system (Forrester, 1969, p. 107 ff.). Through simulation of the models, this counterintuitive behaviour can be revealed and analysed for possible decisions. Though, being widely used for business planning issues SD lacks proper integration into modern BI context. For Example, SD models are explicitly time variant, but they are seldomly related to data warehouse or OLAP concepts, although these are time variant as well (Inmon, 2005). Despite this similarity, little work is found relating these concepts to one another.

This paper aims at relating SD and multidimensional data modelling to one another on a conceptual level. To achieve this conceptual relation, the technique of linguistic metamodelling is applied within a design science related construction process. For reasons of brevity, we focuses on the conceptual properties of the SD modelling language, implementational and calculational aspects remain largely unconsidered.

In the following section, a brief overview of relevant literature is given. Further, the scientific position, construction process and construction technique of metamodelling are discussed. A description of the design process follows, consisting of a construction of the particular metamodels (SD, multidimensional data modelling) and the mapping between the languages. This is followed by an exemplary application and conclusions and suggestions for further research.

LITERATURE REVIEW

We believe that the combination of SD and OLAP can be fruitful (similar Golfarelli, Rizzi & Proli, 2006). It can help corporate planning processes to integrate simulation data by allowing the representation of the vast amount of data generated by SD models in OLAP Tools, which are frequently used in corporate planning applications. By this, SD as a sophisticated analysis technique can be integrated more easily into planning processes. This section gives a short overview of both areas.

According to (Burmester and Goeken, 2006) modelling for OLAP could be conducted from a conceptual and a logical perspective. So far, there is no accepted conceptual modelling language. Instead, the debate is dominated by a broad variety of models for multidimensional structures (for a comparison see e.g. Abello, Samos and Saltor, 2000; Trujillo, Palomar, Gómez and Song, 2001). Actually, there are missing in-depth analysis which relates syntax and semantics of the multidimensional languages and which reflect both with respect to the underlying phenomenon in the universe of discourse. Only few approaches exist, that are based on metamodels representing semantic and syntax in a formalised, unambiguous manner (Holten, 2003; Sapia, Blaschka, Höfling, Dinter, 1998; Goeken, 2006). Furthermore, the integration of simulation data is not supported directly. In the following, we make use of the metamodel presented in (Goeken, 2006), to represent the multidimensional structures.

Golfarelli et al. (see Golfarelli et al., 2006) propose the use of SD models for conducting what-if analysis and representing the results in OLAP context. A review of the existent SD literature reveals that little research is found regarding the integration of SD and multidimensional modelling. The problem has occasionally been addressed on a proprietary, vendor centred level (see e.g. Gonzalez, 2001). Regarding hierarchisation, some approaches could be found, introducing the concept to system dynamics modelling (e.g. Kim & Jun, 1995, Myrtveit, 2000, Eubanks & Yeager, 2001, Liehr, 2001). All approaches have in common that hierarchisation is achieved by reducing visual complexity through partitioning of SD diagrams. The calculation of aggregation functions along a hierarchy like it is used in OLAP is not addressed.

In order to benefit from the combination of both approaches, the lack of semi-formal languages has to be overcome, which should be addressed on a metalinguistic level (see the following section).

SCIENTIFIC POSITION

Design Science Approach and Design Process

This paper follows a design science approach in the broadest sense. In design science research, the work of (Hevner, March, Park & Ram, 2004) has become quite popular, to guide and evaluate scientific contributions. Hence, we use their proposed guidelines to asses our own approach. Here, established concepts and techniques are applied which ensures coherence to the knowledge base of IS research. Here, we use the technique of metamodelling and concepts from the SD and BI domain.

Our research addresses a problem considered as relevant ((***Guideline 2***); see Introduction and Literature Review). Regarding the guidelines, innovative artefacts are created (***Guideline 1***), which serve the purposes outlined so far: metamodels of the level/rate-language and multidimensional data models as well as a translational relationship between them. Concerning the evaluation of the artefact (***Guideline 3***), our mainly descriptive argumentation relies on the mentioned knowledge base. Additionally, the feasibility of implementation is hinted by presentation of a motivating example. We acknowledge that further evaluation is necessary and therefore propose some starting points (see Conclusions and Projected further Research). This paper contributes to design knowledge in two ways (***Guideline 4***): The proposed way of representing simulation results in a multidimensional manner is an artefact itself. Second, the approach could be drawn upon to create further artefacts (models, methods, implementations). In terms of research rigour (***Guideline 5***), we strive for a balanced representation of our findings, by formulating them in terms of semi-formal models. These can serve as starting points for further formalisation. At the same time, the metamodels are still comprehensible due to the selected metalanguage. Additionally, the application of well defined concepts and the form of representation also contributes to the internal consistency of the models. The design of the presented artefact follows a clear construction process (***Guideline 6***). The existent means of the knowledge base are utilised to reach the specified end, the artefact (see Figure 1 for an overview of the construction process). Finally, the appropriate communication of the research results is aspired (***Guideline 7***), by presenting the findings to different relevant communities.

Error! Objects cannot be created from editing field codes.

Figure 1. Design Process

Metamodelling as Design Technique

The analysis of the phenomenon language is subject of many scientific disciplines, like linguistics, psychology, sociology or philosophy (Rosenkranz & Holten, 2007). Here, we will focus mainly on aspects of modelling languages.

The syntactic aspect of a language is concerned with the set of symbols a language consists of and the rules/constraints for using the signs. It is further distinguished between the abstract syntax and the concrete syntax, the so called notation. The *abstract syntax* of a language defines the available language elements and relationships between them, as well as their meaning and generative rules. It defines rules for structuring the real world (or some part of it) which should be described, by specifying the building blocks the world consists of. The representation is matter to the *concrete syntax* which defines the assignment of abstract syntax elements and their relationships to representational objects (e.g. symbols).

Languages are usually used to represent real world objects. Using a modelling language results in a model of a selected part of the real world. If the represented object is a language, the description is performed in a metalanguage (a language to describe languages; Carnap, 1975) resulting in a model of the language, a so called metamodel. Therefore, “model of a modelling language” is another popular definition of “metamodel” (Karagiannis & Kühn, 2002). As it only defines the language and the signs for specifying a model, it does not refer to real world objects itself.

In order to metamodel the abstract syntax of SD-diagrams and multidimensional data models, we describe their syntactic elements. This is achieved by abstracting from notation and by naming and relating the building blocks the language offers. Here, a dialect of the E/R-approach (see Chen, 1976), the extended E/R-model (Scheer, 1998) will be used as metalanguage.

DESIGN

The SD Modelling Language

This section begins with an introduction to the language concepts of the level/rate-language (synonym: stock/flow-language). As mentioned above, only the conceptual aspects of the language should be considered, calculational and implementational aspects must stand back. All explanations refer to the type level of the language which leads to a structural description of a model. The following rationale refers to textual descriptions of the level/rate-language found in (Forrester, 1964; Forrester, 1972; Sterman, 2000; Roberts, 1981. An overview of the most common notation is found in Figure 2.

Node types

Levels are containers, representing state values of system elements. The value of a level changes over time, being the accumulated difference between inflows and outflows of content into, respectively out of the level. ***Rates*** control the flow between the levels of a system, representing the activity inside a system. The control of a flow is achieved via decision functions which determine the amount of flow depending on information about levels in the system. ***Auxiliary*** variables do not belong to the original concepts of the level/rate-language. From a calculational point of view, auxiliary variables are equation parts, unhinged from (comprehensive) rate equations. From a conceptual point of view, they are informational concepts, having an independent meaning. They influence the decision functions that control the rates and are themselves influenced by levels and/or other auxiliaries and constants (see below). In sum, they are derivative concepts, introduced for pragmatic reasons, for easing the communication and improving the clarity of the model. ***Sources and sinks*** represent the boundaries of a system model. Sources are the stocks from which a flow coming from outside the model originates. Respectively, sinks are the stocks taking flows which leave the model. ***Constants*** are state variables which do not or change so slowly that they could be assumed constant for the time scope of the model.

Edge types

Flows are the edges connecting levels, representing the inflow and outflow altering the level. Inflows are pointing at the level, adding content to the level, outflows are pointing away from the level, subtracting content from the level. ***Information links*** are immaterial and connect the inputs for the decision function of a rate. Information links may point to rates and auxiliary variables, but not to levels (may only be changed by flows, see above), constants (do not change, see above) and sources or sinks (beyond scope, see above). However, information links may point away from all element types (information take off), except sources or sinks (beyond scope).

Error! Objects cannot be created from editing field codes.

Figure 2. Concrete Syntax of the Level/Rate-Language

Metamodel

The following section presents the abstract syntax of the level/rate-language. The reading order of the resulting E/R-model is from node type to edge type. Additionally, the naming convention of the relationship types indicates the direction of the edges, e.g. 'Level precedes Flow' describes a flow edge, pointing away from a level-node. A diagram of the resulting metamodel is shown in Figure 3.

Levels are connected to flows pointing at or pointing away from the level. This relationship can be represented as a level succeeding or preceding a flow. Forrester states that „A level may have any number of inflows and outflows." (see Forrester, 1964). In the metamodel this results in cardinalities of (0, 1) on the level side and (0, m) on the flow side. Furthermore levels can only be changed by flows; in particular no causal link can point directly into a stock. However, it is possible that a causal link can point away from a stock (stock precedes causal link; see below).

Additionally ***rates*** define the flows between the levels of a system. Provided that node types can not directly connect to other node types, an edge type has to be the intermediate. Here, the flow type is the intermediate between a rate and a level. Because a flow is controlled by exactly one rate, the cardinalities are (1, 1) on both sides (rate and flow). Rates are determined by the levels of a system. Additionally, rates underlie influences of other, not yet specified, concepts (see below).

From a calculational point of view, ***auxiliaries*** are parts of the decision functions of a rate. They can be embedded (substituted) into the equations underlying the rates. From a conceptual point of view, auxiliaries have an independent meaning. They represent certain aspects of a rates decision function that, for reasons of clarity, should be presented separately from the rates. Auxiliary variables are related to levels, rates, constants and other auxiliaries. They connect to these other constructs solely via information links. Auxiliary variables are depending on levels, constants and other auxiliaries which means that an information link points from the related concept towards the auxiliary (auxiliary succeeds information link). The concepts that influence an auxiliary precede an information link. As stated above, auxiliaries are part of rates decision functions, directly or indirectly influencing the rate of flow. A direct influence would be modelled as an information link pointing towards the rate (rate succeeds information link), an indirect influence would be modelled as an information link pointing towards another auxiliary (auxiliary precedes information link succeeds auxiliary). The cardinalities of these relationships are depicted in Figure 3.

Constants influence rates directly or indirectly via auxiliary variables, connecting to them through information links. Constants themselves do not change which means no other concept influences (precedes) them. The cardinalities for this relationship would be (0, 1) on the constant side and (1, m) on the information link side.

Sources are stocks generating flows from outside the models boundaries. ***Sinks*** are stocks taking flows outside the models boundaries. These facts could be modelled as source preceding a flow, respectively a sink succeeding a flow. Since the sources and sinks are not differentiated regarding their contents, the cardinalities would be (0, 1) on source and sink side and one to many (1, m) on the flow side.

Figure 3 shows the abstract syntax of the level/rate-language. All components are on type level. The node-types and edge-types could be generalised into a more compact representation of the metamodel (see upper part of Figure 3).

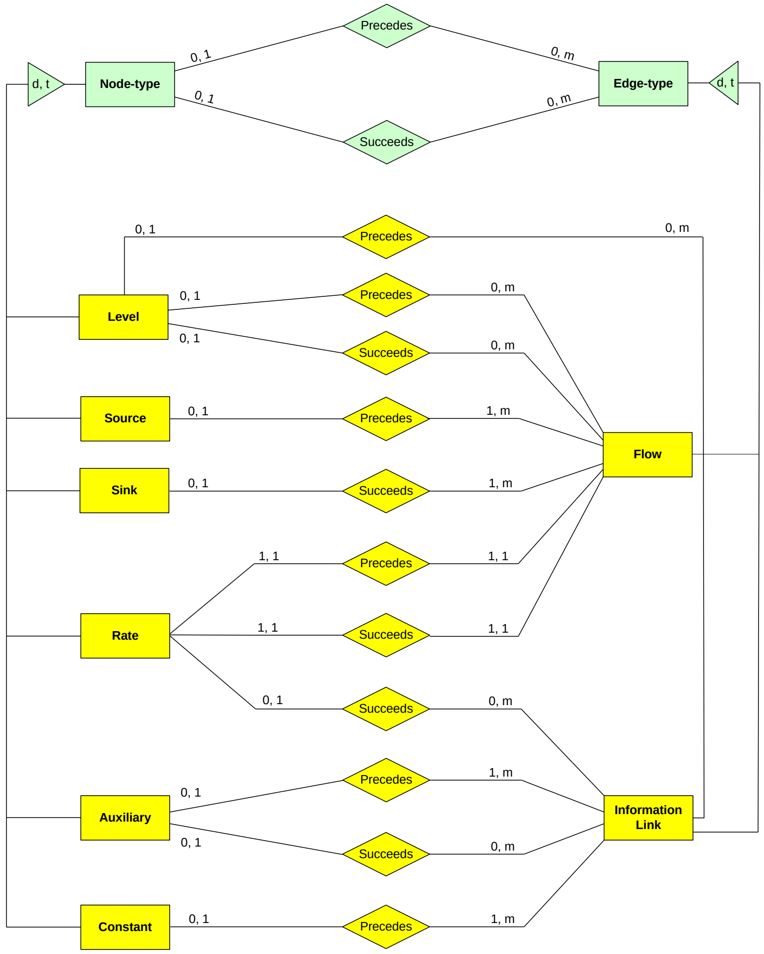

Figure 3. Abstract Syntax of the Level/Rate-Language

Modelling for OLAP

In order to support management decisions, OLAP Systems use at least two techniques to represent data, hierarchisation and multidimensionality.

Hierarchisation

A hierarchy results from creating sets of elements and, in a second step, from creating sets of sets (Goeken, 2006). The grouping of elements or sets should be guided by some type of set definition (e.g. loans with a variable interest rate could be grouped into “0-5%”, “6-10%” and “above 10%”). Due to the fact, that hierarchisation starts with granular elements, which are on the same level of the hierarchy, we state, that sets can also be located on a certain level. A repetitive application of the grouping leads to three possible types of hierarchical levels. First, a level could contain all elements. As it contains all available elements, this level is termed elementary level. Secondly, a level could contain only one set which all other elements are subordinated to. As this level is the initial point of navigation through the subordinated hierarchy it is called root level. At last, there can be intermediate levels, residing in-between (see Figure 4).

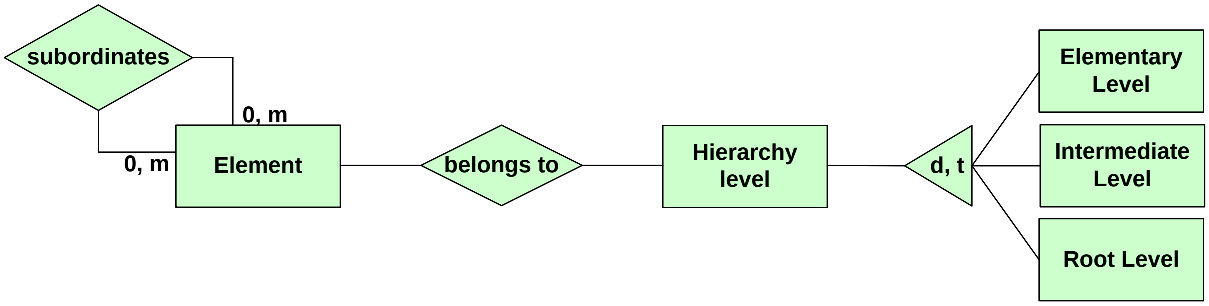

Figure 4. Hierarchical Levels

Multidimensionality

Multidimensional models consist of quantifying and qualifying data. The former, often referred to as measures, represent values of relevant objects of an application domain (e.g. turnover, sales etc.). Measures are qualified through dimensions, describing the selected viewpoints (e.g. time, region, and customer), leading to concrete information (e.g. sales for December 2007 (time) in Germany (region) with 'mega mart' (customer)). Combining quantifying and qualifying data, results in a cube, which represents both.

Dimensions consist of dimensional nodes (at least one (1)) which are regularly organised to hierarchies, following the mechanism of defining set memberships. The hierarchisation allows changing the level of detail a business object is represented, adapting view complexity to the actual information requirements. The multidimensional approach, accompanied by the described complexity adaption mechanism through hierarchisation, allows users to visualise a comprehensive picture of business objects. Figure 5 depicts the metamodel of multidimensional models. For a detailed derivation of the multidimensional metamodel see (Goeken, 2006).

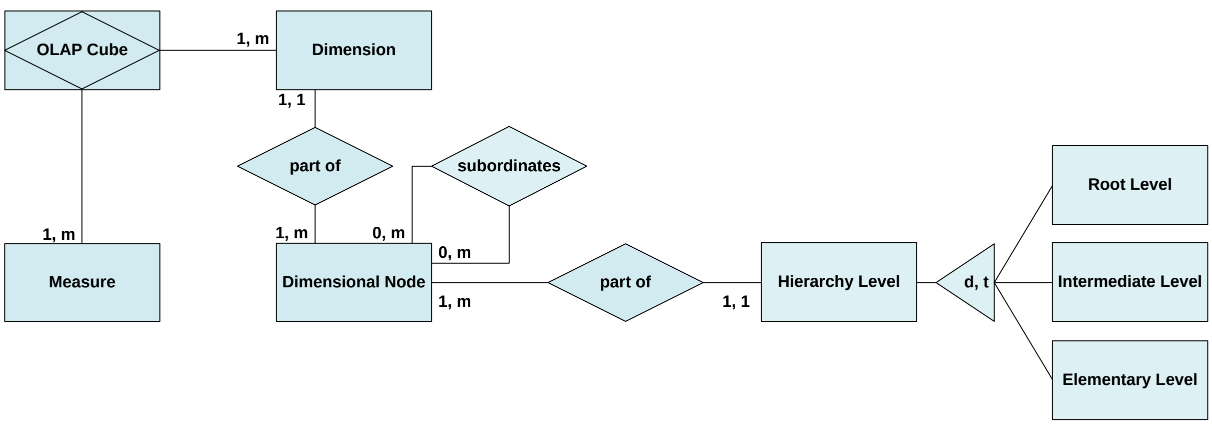

Figure 5. Multidimensional Metamodel

Metamodel Based Relation of SD and Multidimensional Modelling

For relating the approaches, the quantifying and qualifying aspects of SD models must be identified. However, this requires an extension of the scope of considerations from solely static aspects of model structure towards the dynamic results yielded by a simulation of the SD model. During the simulation of the model, the values of the variables are calculated, depending on their interrelationships while the simulation time advances. The result is a time series for each variable, representing the value of the variable at a certain point of time. A model could be simulated with different parameterisation, meaning that the value of constants and initial values of variables differ between two simulation runs. The result is another set of time series which could be compared to time series from previous simulation runs.

The sets of time series could already be regarded as multidimensional information about the SD model. The quantifying information within a model are the variables changing during the simulation. Their values are obviously qualified by a time dimension. Furthermore, the variables of a SD model depend on parameters defined at the beginning of a simulation. These parameters also qualify the values of the variables generated during simulation, with each parameter constituting a dimension. The variation of a model parameter between simulation runs leads to a set of dimensional nodes which could be organised into a dimensional hierarchy in the above described manner.

In terms of the metamodel, the above described could be formulated as follows. The node types can be specialised into parameters of the model (qualifying information) and variables (quantifying information). The parameters of a model are the constants and the initial values of the variables. The variables of the model are the levels, rates and auxiliary variables. Other node type concepts remain unconsidered because they cannot assume values. The metamodels and their correspondences are depicted in Figure 6. As stated above, parameters of the SD model correspond to dimensional nodes and variables of the SD model correspond to measures.

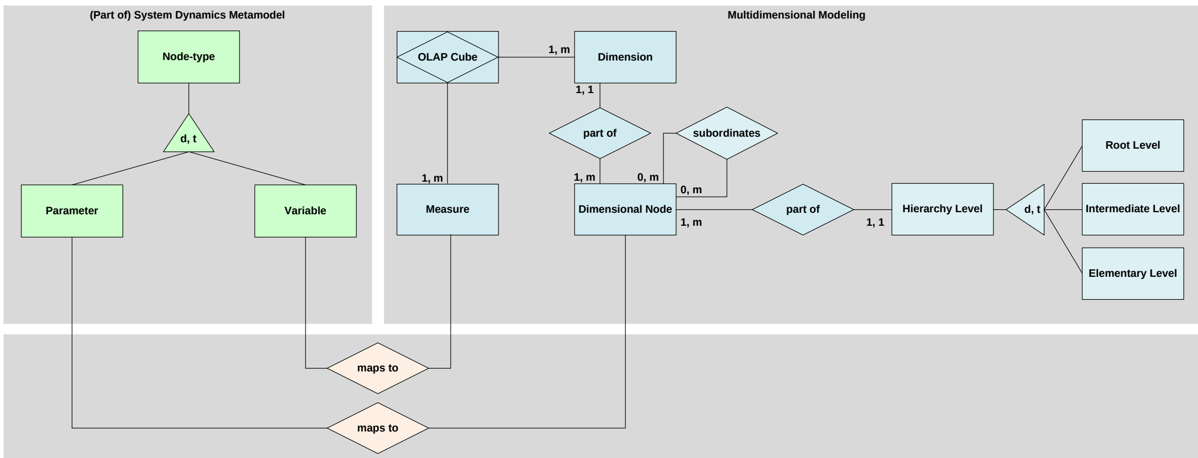

Figure 6. Combining of SD Metamodel and Multidimensional Metamodel

EXAMPLE

In the following, a part of a balanced scorecard (BSC) solution for faculty management should serve as an illustrating example. A SD model is utilised for representation and simulation of the balanced scorecard's underlying causal structure (see Kaplan & Norton, 1996, for an application example see Akkermans & van Oorshot, 2005). Technical implementations of balanced scorecards often rely on BI technologies such as data warehouses and OLAP-cubes. The choice of the application domain is due to the authors' experience of conceiving and developing BI solutions for university and faculty management. For reasons of brevity only a small part of the SD model is presented.

The assumed causal structure spans the four dimensions of a balanced scorecard (financial, resources, processes and customers). Financial resources of the faculty rely on two sources: the state funding, which is based on well defined parameters like the number of students or graduates and the tuition fees paid by students. The latter are assumed to influence the amount of students actually beginning their study. The funds are spent for three main expenses: personnel, tangible means (library, IT etc.) and facility management. Here, just the personnel expenditures should be considered, as they determine how many persons could be appointed to the teaching staff (professors, research assistants). It is assumed that the ratio of students and teaching staff (mentoring ratio) has a significant influence on the dropout rate of students (students leaving without degree). A high dropout rate will lower the amount of students studying and graduating at the faculty. Due to the fact that large parts of the funding relies on the population of students, a causal link to the financial perspective is established, which closes the assumed causal feedback loop. (See Figure 7 for a SD model of the described causal structure).

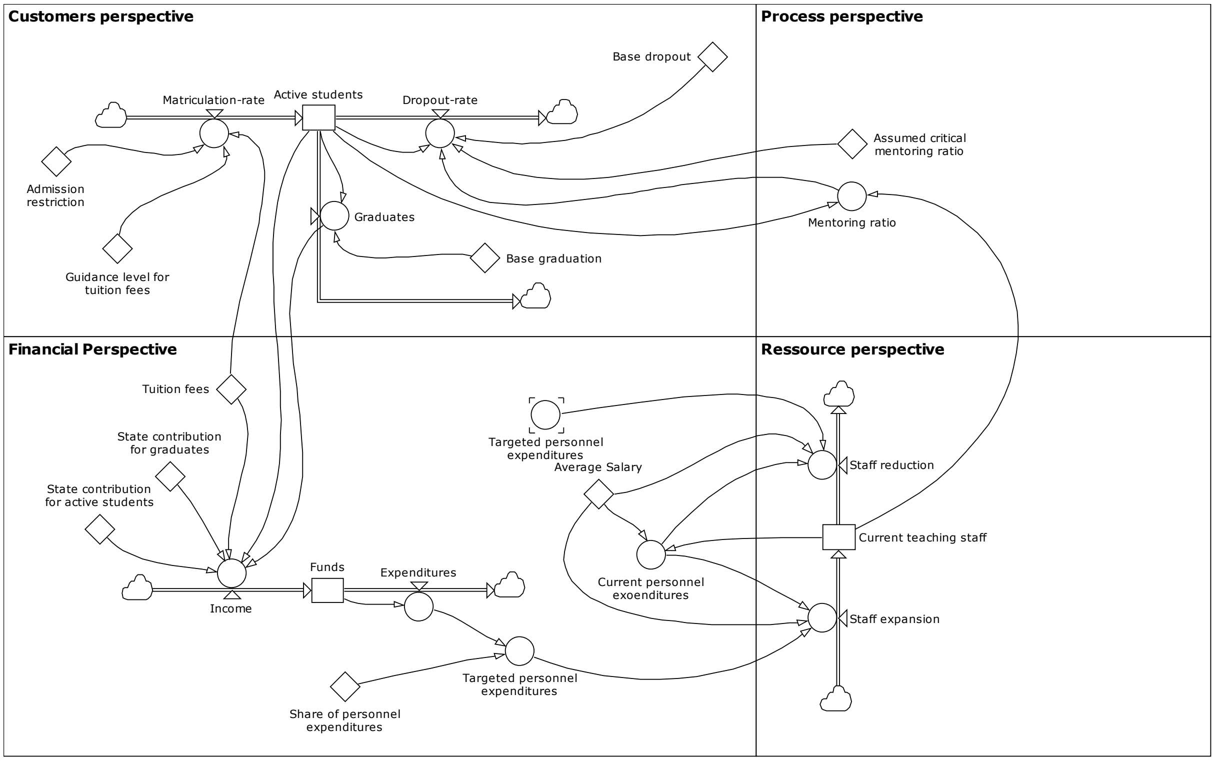

Figure 7. SD model of the BSCs assumed causal structure

Regarding the model, several parameters can be identified, some of them influenceable by faculty managers (*tuition fees*, *personnel expenditures*, *average staff salary*, *admission restrictions*). Others are regarded as constant or not influencable (e.g. *state contribution*) or represent model inherent causal assumptions (e.g. effect of *mentoring ratio* on *dropout rate*). In the following, we will concentrate on the influenceable parameters. Furthermore, a range of measures could be identified. In the customer perspective these would be the *matriculation rate*, the *number of active students*, the *dropout rate* and the *number of graduates*. Financial measures would be the *available funds* and *targeted* and *factual personnel expenditures*. The *current teaching staff* would be a measure for the resources of the faculty and the *mentoring ratio* could be seen as a quality aspect of the teaching process.

During a simulation of the model, a time series of each measure, depending on the parameters is generated. Each value of this time series can be qualified by the point of time and the values of the parameters. For example, the measure 'active students' could be qualified by the point of time, the tuition fees charged, the admittance of new students and the personnel expenditures (see Figure 8).

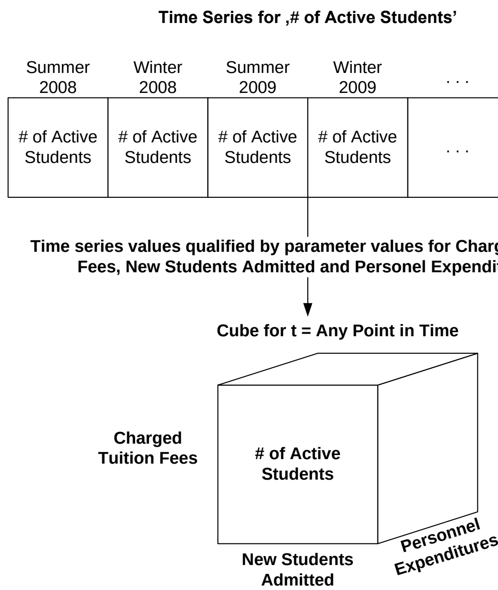

Figure 8. Multidimensional Representation of Model Variables

A repetition of the simulation with varying parameters yields a set of dimensional nodes for each parameter. For example, simulating the model with ten different tuition fees leads to ten dimensional nodes. Further simulations with other varied parameters, lead to even more dimensional nodes which qualify the measures. The number of value cells of the resulting cube could be determined by the Cartesian product of the values each parameter assumes. For example, letting the parameters tuition fees, admissions and personnel expenditures vary in 10 steps each and simulate the model half-yearly (winter & summer term) for 10 years would yield a hypercube with 20.000 cells (10^3 x 20). Figure 9 depicts an exemplary cube.

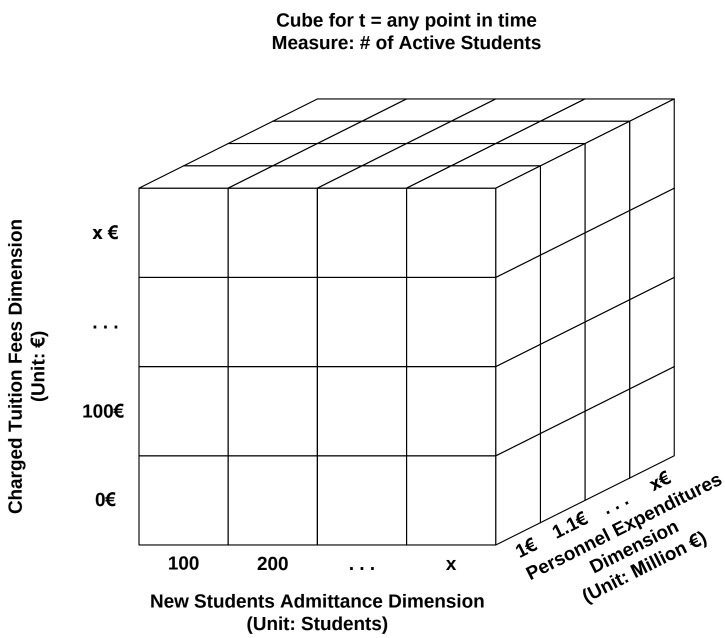

Figure 9. Hypercube Representing Repeated Simulation Runs

To handle this vast amount of generated data, the describing dimensional nodes could be organised into hierarchies.. For example, the parameter values for charged tuition fees could be organised into three groups (low, medium, high) relative to the state issued guidance level for tuition fees. This hierarchisation could be applied to all dimensions, leading to a full multidimensional description of the SD model. For a multidimensional schema of the example see Figure 10.

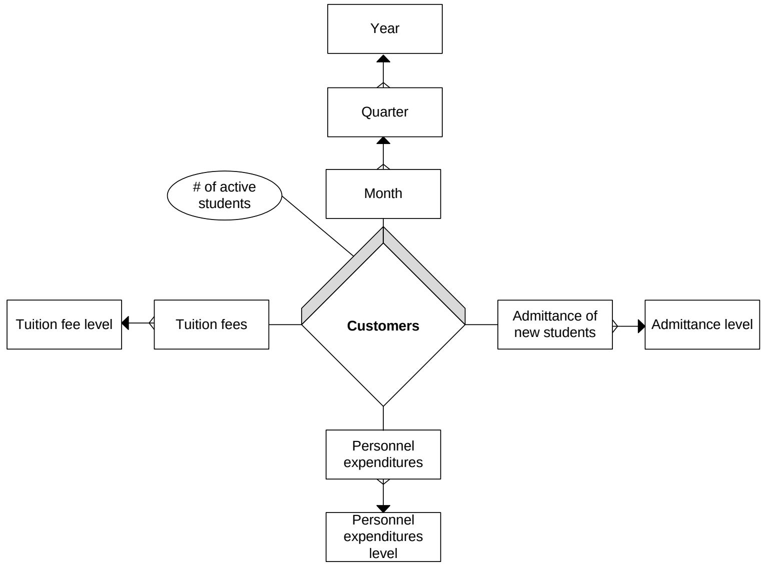

Figure 10. Multidimensional E/R-Model of the Example

CONCLUSIONS AND PROJECTED NEED OF FURTHER RESEARCH

In this paper we presented metamodels of the SD modelling language, multidimensional data models and showed a way of representing the simulation results in a multidimensional manner. The combination of information enriching multidimensionality and complexity reducing hierarchisation can be considered state of the art for the support of managerial work. The complementation of this approach with a possibility to simulate complex, dynamic and often counterintuitive system behaviour further augments management support. We propose three directions for further research. From the viewpoint of decision science it could be evaluated in how far the augmented information provided improves the quality of managerial decisions. From a practical point of view, the explanation of a modelling language is not sufficient for applicability of the introduced ideas. Further research should be pointed at introducing a way of working with the language definitions and be aimed towards an integrated methodology (see also Golfarelli et al., 2006 for a similar statement of research issues). From a linguistic-theoretical point of view, an ontological analysis of the modelling language and the representational benefits of its extension could be interesting (see Wand & Weber, 1993, Rosemann & Green, 2002). During this analysis, the ontological completeness (according to a reference ontology, e.g. Bunge-Wand-Weber or Chisholm) would be examined for the original language as well as for the result of an extension. Further, hints for combination of the level/rate-language with other modelling languages to reduce the representational deficiencies could be produced.

REFERENCES

1. Abello, A.; Samos, J.; Saltor, F. (2000). A Data Warehouse Multidimensional Data Models Classification. Technical Report LSI-2000-6, Universidad de Granada, 2000.
2. Akkermans, H., & van Oorshot, K. (2005). Relevance Assumed: A Case Study of Balanced Scorecard Development Using System Dynamics. *Journal of the System Dynamics Society, 56*, 931–941
3. Burmester, L.; Goeken, M. (2006). Method for User Oriented Modelling of Data Warehouse Systems. In: Manolopoulos, Y. et al. (ed.): Proceedings of ICEIS 2006, Paphos/Cyprus.
4. Carnap, R. (1975). *Introduction to Semantics and Formalization of Logic.* Cambridge, Mass.: Harvard Univ. Press.
5. Chen, P. (1976). The Entity-Relationship Model: Toward a unified View of Data. *ACM Transactions on Database Systems, 1* (Nr. 1), 9–36
6. Eubanks, K., & Yeager, L. (2001). An Introduction to Jitia: Simulation Software Designed for Developing Large System Models. In J. Hines, V. Diker, R. Langer & J. Rowe (Eds.), *Proceedings of the 19th System Dynamics Conference*. Atlanta: System Dynamics Society.
7. Forrester, J. (1964). *Industrial Dynamics* (4th edition). Cambridge, Mass.: MIT Press.
8. Forrester, J. (1969). *Urban Dynamics.* Cambridge, Mass.: M.I.T. Press.

9. Forrester, J. (1972). *Grundsätze einer Systemtheorie: (Principles of systems).* Wiesbaden: Gabler.
10. Goeken, M. (2006). Entwicklung von Data-Warehouse-Systemen. Wiesbaden 2006 [in german].
11. Goeken, M. (2006). *Entwicklung von Data-Warehouse-Systemen: Anforderungsmanagement, Modellierung, Implementierung.* Wiesbaden: Deutscher Universitätsverlag.
12. Golfarelli, M., Rizzi, S., & Proli, A. (2006). Designing What-if Analysis: Towards a Methodology. In I. Song & P. Vassiliadis (Eds.), *Proceedings of DOLAP 06* (pp. 51–58). New York: ACM.
13. Gonzalez, F. (2001). Integrating Dynamic Simulations With An OLAP System. In J. Hines, V. Diker, R. Langer & J. Rowe (Eds.), *Proceedings of the 19th System Dynamics Conference*. Atlanta: System Dynamics Society.
14. Hevner, A., March, S., Park, J., & Ram, S. (2004). Design Science In Information Systems Research. *Management Information Systems Quarterly, 28*(1), 75–105
15. Holten, R. (2003). Specification of management views in information warehouse projects. Information Systems, 28 (2003) 7, pp. 709-751.
16. Inmon, W. H. (2005). *Building the Data Warehouse.* Indianapolis, Ind.: Wiley.
17. Kaplan, R., & Norton, D. (1996). Using the Balanced Scorecard as a Strategic Management System. *Harvard Business Review, 74*(1), 74–85
18. Karagiannis, D., & Kühn, H. (2002). Metamodeling Platforms. In A. Min Tjoa & G. Quirchmayer (Eds.), *Lecture Notes in Computer Science: Vol. 2455. Proceedings of the Third International Conference EC-Web 2002* (pp. 451–464). Springer.
19. Kim, D., & Jun, J. (1995). Hierarchical Approach in System Dynamics Modeling. In T. Shimada & K. Saeed (Eds.), *Proceedings of the 13th International Conference of the System Dynamics Society* (pp. 94–103). Tokyo: System Dynamics Society.
20. Liehr, M. (2001). Towards a Platform - Strategy for System Dynamics Modeling: Using Generic Structures Hierarchically. In J. Hines, V. Diker, R. Langer & J. Rowe (Eds.), *Proceedings of the 19th System Dynamics Conference*. Atlanta: System Dynamics Society.
21. Myrtveit, M. (2000). Object-oriented Extensions to System Dynamics. In P. Davidsen, D. Ford & A. Mashayekhi (Eds.), *Sustainability in the Third Millennium. Proceedings of the 18th International Conference of the System Dynamics Society*. Bergen.
22. Roberts, E. B. (1981). System Dynamics - An Introduction. In E. B. Roberts (Ed.), *MIT Press/Wright-Allen series in system dynamics: . Managerial Applications of System Dynamics* (pp. 3–35). Cambridge, Mass.: MIT Press.
23. Rosemann, M., & Green, P. (2002). Developing a Meta Model for the Bunge-Wand-Weber Ontological Constructs. *Information Systems, 27*(2), 75–91
24. Rosenkranz, C., & Holten, R. (2007). Combining Cybernetics and Conceptual Modeling - The Concept of Variety in Organizational Engineering. In Y. Cho, R. Wainwright, H. Haddad & S. Shin (Eds.), *SAC '07: Proceedings of the 2007 ACM symposium on Applied computing* (pp. 1228–1233). New York: ACM Press.
25. Sapia, C.; Blaschka, M.; Höfling, G.; Dinter, B. (1998). Extending the E/R Model for the Multidimensional Paradigm. In: Proceedings of DWDM’98. LNCS, Heidelberg 1998, pp. 105-116.
26. Scheer, A. W. (1998). *Business Process Engineering: Reference Models for Industrial Enterprises.* Berlin, Heidelberg, New York: Springer.
27. Simon, H. A. (1996). *The Sciences of the Artificial* (3rd ed.). Cambridge, Mass.: MIT Press.
28. Sterman, J. (2000). *Business Dynamics: Systems thinking and modeling for a complex world.* Boston: Irwin/McGraw-Hill.
29. Trujillo, J., Palomar, M., Gómez, J., Song, I.-Y. (2001). Designing Data Warehouses with OO Conceptual Models. IEEE Computer 34, pp. 66-75
30. Wand, Y., & Weber, R. (1993). On the Ontological Expressiveness of Information Systems Analysis and Design Grammar. *Journal of Information Systems, 7*(3), 217–237