Association for Information Systems
AIS Electronic Library (AISeL)

ACIS 2006 Proceedings | Australasian (ACIS)

2006

Procedure Model for the Analysis and Design of Reporting Systems – A Case Study in Conceptual Modelling

Stefan Seidel
Queensland University of Technology, s.seidel@qut.edu.au

Ralf Knackstedt
European Research Center for Information Systems, ralf.knackstedt@ercis.de

Christian Janiesch
European Research Center for Information Systems, christian.janiesch@ercis.de

Procedure Model for the Analysis and Design of Reporting Systems – A Case Study in Conceptual Modelling

Stefan Seidel
Queensland University of Technology
s.seidel@qut.edu.au

Ralf Knackstedt
Christian Janiesch
European Research Center for Information Systems
ralf.knackstedt@ercis.de
christian.janiesch@ercis.de

Abstract

A structured conceptual design of reporting systems is a crucial task that has to precede implementation and monitoring. A basic challenge can be seen in information requirements engineering. On the one hand, users need certain information to successfully accomplish their tasks. On the other hand, it has to be avoided to supply them with too much and potentially irrelevant information. Information requirements engineering has to support the conceptual specification of information requirements. Thus, there is need for conceptual languages and appropriate procedure models. We propose a procedure model for the analysis and conceptual design of reporting systems as well as a conceptual modelling language that can be used for both as-is analysis and to-be modelling. It has been developed in an iterative approach based on multiple case studies. Due to the fact that different projects and different basic conditions require the use of different approaches of information requirements engineering, within the procedure model different approaches can be used. In this paper we introduce both procedure model and conceptual modelling language and present findings of a case study in which they were used.

Keywords

Case Study, Conceptual Modelling, Reporting, Data Warehousing, Information Requirements Engineering

INTRODUCTION

In nowadays business environments, besides standard reports that periodically supply users with information there is a need for exception reporting and ad-hoc reporting as well as for other analyses. Management has to react in quickly changing decision situations. Management Support Systems (MSS) and technologies such as Data Warehousing and Online Analytical Processing (OLAP) build up a sound foundation to reach these goals. However, the introduction of systems often fails due to deficient information requirements analysis. To overcome this, there have to be both modelling languages that allow a sound conceptual definition of information requirements and detailed procedure models that guide the use of these modelling languages. A sound communication between the IT users and IT developers is crucial, since system users are often not able to express and specify their information requirements. Furthermore, conceptual models serve as a starting point for later implementation.

This work makes a contribution to the Information Systems body of knowledge by introducing a detailed procedure model for the analysis and conceptual design of reporting systems as well as a conceptual modelling language for information requirements. Conceptual language and procedure model have been developed in an iterative process based on different case studies. We justify the development of a new approach as follows: Existing approaches (compare Related Work) do not meet all the requirements to a method for the conceptual design of reporting systems we have identified in several case studies and within a comprehensive literature review. The most important requirements are:

> (R1) Both as-is modelling and to-be modelling should be supported by a conceptual modelling language to foster communication. We propose a conceptual language that can be used both for analysis and conceptual design of reporting systems. This is reasoned as follows: First, by the continuous use of a conceptual modelling language communication between users and IT developers is fostered; communication is very important in the process of information requirements analysis. Second, the use of the same language of as-is analysis and to-be modelling enables the re-use of those parts of the as-is models that are still valid within to-be modelling.

(R2) In contrast to existing approaches that suggest to conduct conceptual modelling after the to-be state has been identified, we propose to construct the future system's conceptual model parallel to the identification of the to-be state using the conceptual modelling language. Thus, the modelling language has to be a constituent part of the procedure model. This is reasoned again by the important role that communication plays during requirements engineering as well as by the possibility to re-use results of the as-is analysis.

(R3) Different projects require the use of different approaches of information requirements analysis. This is due to differences in existing systems, time frames, budgets and customer requirements. For example, the management could request a mere transfer of the existing reporting system to a new technical platform. Consequently, the procedure model has to be capable of supporting different approaches. An overview about existing approaches to information requirements engineering can be found in the section about related work.

(R4) Existing procedure models tend to be on a very abstract level and often do not support the iterative nature of information requirements engineering (see section about related work). Consequently, our procedure model contains several iteration steps.

(R6) The conceptual language has to serve as a communication basis between IT users and IT developers. At the same time it must hold a degree of formality that allows the usage of the models as a starting point for system implementation. Besides, the conceptual language has to cover two basic aspects regarding reporting systems: It must be capable of being used for designing data warehouse structures as well as particular reports. We argue that a detailed description of reports respectively management views is necessary to depict particular information requirements and to serve as a starting point for implementation.

Evidence to support this we have collected in multiple case studies (Yin, 2003) of which one we present here. We illustrate how the method (consisting of procedure model and conceptual modelling language) has been used in a case study with a German retail company for luxury goods and what we have learned from this case study.

RELATED WORK

In literature there are many approaches to information requirements analysis, such as document analysis or interviewing (Carter, 1983; Watson and Frolick, 1993). Furthermore, there are more complex approaches, such as Business Systems Planning (Zachmann, 1982) or Critical Success Factors (Rockart, 1979). These approaches are often called *methods* of information requirements analysis. Since reporting systems are closely linked to data warehouses (DWH) which build the technological foundation, especially approaches to data warehouse development have to be considered (Golfarelli and Rizzi, 1998; Winter and Strauch, 2004). Winter and Strauch distinguish two basic approaches, demand-driven approaches and supply-driven approaches. They propose a method for demand-driven information requirements analysis in data warehousing projects. Based on case studies they derive an activity model that is used in a comprehensive methodology. List et al. identify three basic groups of data warehouse development methods: data-driven, goal-driven, and user-driven methods (List *et al.*, 2002). Data-driven methods do a priori not consider information requirements but focus on the analysis of the organisation's data model. Goal-driven approaches derive data warehouse structures based on an organisation's goals. User-driven approaches make use of techniques such as interviewing or observation. Becker and Dreiling suggest that information requirements can be derived from business processes as well as from objectives (Becker *et al.*, 2003; Becker *et al.*, 2004).

To meet the requirements to a modern reporting and due to the difference between projects and to varying basic conditions, we propose the use of a combination of different approaches. Basically, all possible approaches should be considered and it should be decided which approach or which approaches are actually to be used in a particular project: On the one hand, it is necessary to consider the as-is state to identify weaknesses and to build up a sound foundation for communication between developers and users. On the other hand, demand- or task-driven approaches have to be used to enable the construction of the to-be state. List et al. justify the need for task-oriented information requirements analysis by pointing out that only accounting for future information requirements allows for a longevity of systems (List *et al.*, 2002). Demand-driven approaches can usually be conducted easily and improve the acceptance of the future system by its users (List *et al.*, 2002).

In literature, a variety of procedure models for software engineering is proposed. Here, we focus on some well-known procedure models with special regard to procedure models for the development of data warehouse systems.

One of the best known procedure models is the so-called *phase model* (Royce, 1987). It consists of a sequence of different phases. The establishing of phases allows a reduction of complexity. The phase model is suited for the development of operative systems. Phase models for the development of data warehouses have been introduced for example by (Inmon, 1996; Kimball, 1996; Poe, 1996). However, regarding the development of management support systems these are quite deficient: As stated above, one of the main challenges is the identification of information requirements which is part of the analysis phase. A complete requirements analysis as required by the phase models is a very complex task. Therefore, detailed problem solving techniques are needed. Furthermore, a linear sequence of phases often cannot be obtained in reality (Inmon, 1991). Consequently, for the development of reporting systems a procedure model that supports an iterative approach is required.

Another approach is *prototyping*. A model of the final software is developed. The so-called *evolutionary prototyping* is described as a sustainable approach to developing management support systems by (Gabriel and Gluchowski, 1998). In this approach different phases are run through successively. Gradually, new and enhanced versions of the product with fewer errors are introduced and tested. The advantage of the approach is the involvement of the user into the development process. This is especially important in the context of reporting systems, since prototypes can serve as a communication basis.

Winter and Strauch introduce a procedure model for information requirements analysis in data warehousing (Winter and Strauch, 2004). They introduce different techniques and show their use in the context of a procedure model. They emphasise the importance of using semantic data models, especially to build up a communication basis. Conceptual Modelling is conducted after the to-be state has been identified. However, as argued above, we propose to construct the future system's conceptual model parallel to the identification of the to-be state. Furthermore, we propose to conduct an as-is analysis using the same modelling language used for the to-be modelling. Hence, the modelling language is a constituent part of the procedure model.

METHOD FOR CONCEPTUAL DESIGN OF REPORTING SYSTEMS

Procedure Model for the Analysis and Conceptual Design of Reporting Systems

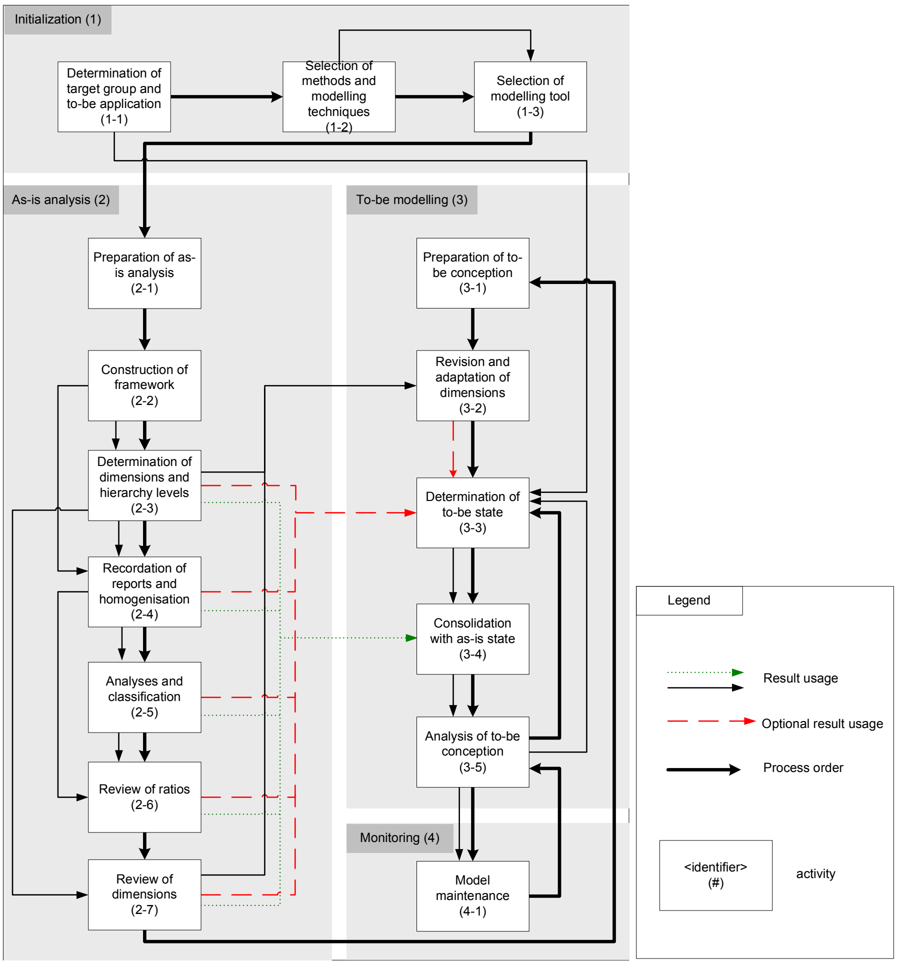

Figure 1: Procedure Model for the Analysis and Design of Reporting Systems

Figure 1 shows the procedure model as it resulted from the case studies and a comprehensive literature review we have conducted. In this paper, we do not aim to elaborate on the construction of the model but on its use in a case study. The procedure model comprises the phases of initialisation, as-is-analysis, to-be-modelling and monitoring. Phases are subdivided into activities. Within the initialisation phase, basic questions concerning existing systems and systems that are to be developed are answered. Within as-is-analysis, a more or less detailed assessment of the as-is state is conducted. Both master data such as product structures and meta-data about reports are assessed. This way an information model of the system's current state is created. This information model can be analysed and the results can be used within to-be modelling. In the latter, several methods of information requirements analysis are used to determine the reporting's to-be state. The results of the as-is-analysis can be used as a basis for to-be modelling. Thereafter, a consolidation of as-is-analysis and to-be modelling is conducted. Finally, subject to the monitoring phase is a continuous adaptation and maintenance of the models. The dotted lines in Figure 1 show that the use of as-is models for to-be modelling is optional. But in any case there has to be a consolidation of as-is state and to-be state since it has to be ensured that errors, week spots and inconsistencies of the as-is state are avoided. Activities have a number. These numbers depict the activities' order and serve as an orientation guide. As stated above, conceptual modelling is a constituent part of the procedure model. , we introduce such a conceptual modelling language that can be used both for as-is analysis and to-be modelling.

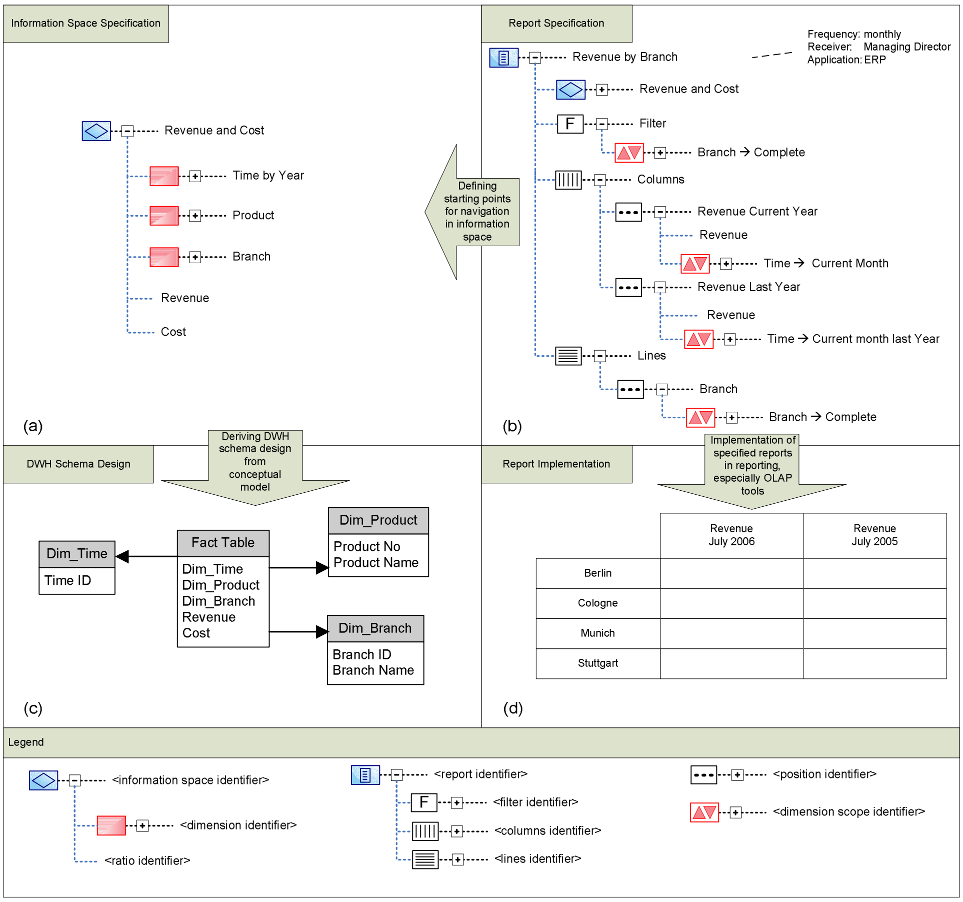

Figure 2: Conceptual modelling language in the context of the design of a reporting system

Conceptual Modelling Language

The introduction of a new modelling language is reasoned by the fact that existing approaches (Sapia *et al.*, 1998; Bulos, 1996; Golfarelli *et al.*, 1998) do not allow for the detailed conceptual design of both data warehouse structures and certain reports (R5). For a more detailed reasoning see (Becker *et al.*, 2006). The conceptual modelling language we propose here is called *MetaMIS for Reporting*. It is based on the conceptual modelling language MetaMIS (Holten, 2001). MetaMIS enables the conceptual modelling of management views. Models consist of tree structures. Basic constructs of MetaMIS for Reporting are *information spaces* and *reports*. For the representation of the language constructs cf. Figure 2. Information spaces are multidimensional views consisting of *dimensions* and *ratios* and thus constituting data warehouse structures. Dimensions consist of *dimension objects* such as certain products that can be ordered hierarchically (for example, *product* by *product* *group*). The combination of a (combined) reference object (consisting of dimension objects from different dimensions, for example, "product group A in branch XY for the year 2006") with a ratio results in a *fact*.

CASE STUDY: METHOD EVALUATION

We present a case study with a German retail company for luxury goods where both conceptual language and procedure model were evaluated. The company has more than 200 branches in Germany. In the course of the introduction of a new enterprise resource planning system, also a new management information system was introduced. Table 1 gives some facts about the case study.

Number of interviews	Interview Partners	Examined documents and systems
21	CIO, Controlling, Top Management, Managers Purchasing, Managers Sales, Managers Logistics, IT Department	Standard reports (daily, monthly), data warehouse for purchasing

Table 1: Facts on the case study

Initialisation

Determination of target group and to-be application (1-1)

At the outset, the to-be application is determined. This includes the identification of the future system's target group. Report receiver and their positions within the organisation are determined. Due to complexity reduction it can be helpful to focus on a subdivision of the organisation (Winter and Strauch, 2004). Furthermore, it has to be specified what areas the future system has to cover (e.g. sales, production). This can only be done in cooperation with the system's later users, the report receivers. It is determined how flexible the system has to be: is there only need for standard reports or and in what detail should there also be ad-hoc reports, exception reports and OLAP functionality. A mere migration of an existing system is a possible goal, for example. In our case study both OLAP functionality and standard reporting were required. Standard reports have a predefined structure and are generated on a regular (e.g. monthly sales reports for product groups) base whereas OLAP functionality allows multidimensional analyses at any time. Report receivers are the top management and managers in purchasing and sales. Furthermore, it was required that all information being part of the as-is state should be considered within the to-be state. These requirements were identified in interviews with the CIO, top management and controlling.

Selection of methods and modelling techniques (1-2)

Subject to this activity is to decide which methods of information requirements analysis are to be used. To what extend different phases are accentuated and which methods of information requirements analysis are used in the context of to-be modelling is due to the goals of a project and to other basic conditions such as project budget, project duration, and individual abilities of project members. In our case study, a detailed as-is analysis had to be conducted. This was reasoned by the fact that the existing reporting system had been successfully used and was quite complex. Consequently, no information that had been part of the existing system should be missing in the future system. It was decided that based on this analysis interviews with report receivers should be conducted to determine and model the to-be state. For analysis and to-be modelling the same modelling language should be used. The used language has been introduced in section 2 of this paper.

Selection of modelling tool (1-3)

The complexity of software development requires tool support for the methodical construction of results and for the conduction of a procedure model. Hence, there is a need for tools enabling the modelling process, monitoring of models and tools for definition, conduction, depiction, and monitoring of procedure models. As possible, the whole construction process should be supported by software tools. Under certain conditions this requires tool integration. In the case study the H2 Toolset (Delfmann *et al.*, 2006; Knackstedt *et al.*, 2006) was used. This application supports the conceptual modelling language introduced in section 2.

As-is-analysis

Preparation of as-is analysis (2-1)

Before the as-is analysis is conducted its level of detail has to be determined. Especially the estimation of how much of the as-is state will still be relevant for the future system has to be considered. In many companies there has been developed a comprehensive reporting based on heterogeneous systems. Some of the reasons that militate in favour of a comprehensive as-is analysis are the following (Becker *et al.*, 2006; Winter and Strauch, 2004):

- Identification of weaknesses, errors and inconsistencies of the as-is state help to avoid these within the to-be modelling. Furthermore, it can be assured that all relevant aspects are accounted for.
- The modelling of the as-is state can serve as a basis for to-be modelling. In the context of reporting systems it cannot be assumed that information categorically loose validity.
- Detailed modelling fosters the user's comprehension and thus the acceptance of the future system.

Against a detailed as-is analysis militate reasons such as creativity restraints and cost-benefit analysis. Hence, the usefulness of a detailed as-is analysis has to be determined based on forecasted call for action (Becker *et al.*, 2006; Rosemann, 2000). Furthermore, modelling conventions have to be set up to ensure the consistent use of terms. As mentioned earlier, in the case study it was required that all information being part of the as-is state should be regarded to in the future system. Consequently, it was decided to conduct a detailed as-is analysis.

Construction of framework (2-2)

This activity aims at systematising and decomposing the field of interest. Such a grouping gives indications to further action within the as-is analysis. The classification can be conducted with regard to different criteria such as applications that produce reports or addressees. This allows the persons involved in the project to integrate their working field into the overall project. Furthermore, a framework helps to introduce terms that are mandatory for all participants of the project. A framework for a company's reporting comprises all divisions that produce or receive reports, hence, all report receivers and all persons in charge. In the case study the existing reporting could be roughly divided into reports for purchasing of clocks, reports for purchasing jewellery, product manager reports and reports for top management. This grouping was then used for the identification of interview groups.

Determination of dimensions and hierarchy levels (2-3)

This activity aims at the recordation and conceptual modelling of dimensions used in reports. Existing documentations, report definitions and actual reports are a possible starting point. Likewise, a review of master data in operative systems is possible. If there already is an existing data warehouse its structure can give information about the composition of hierarchies. Figure 3 depicts three exemplary dimensions that could be identified in the retail case based on reports and expert interviews. It turned out that the models were an appropriate means that fostered communication. For example, interview partners from the controlling department could identify when any of the modelling done by the IT analysts did not depict the actual situation.

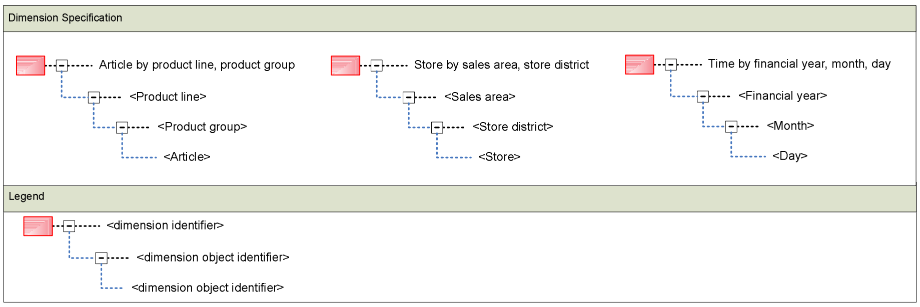

Figure 3: Exemplary dimensions identified within as-is analysis

Recordation of reports and homogenisation (identification of synonyms and homonyms) (2-4)

Reports are recorded and modelled conceptually. Parallel to this recordation, a glossary with synonymous terms for ratios is kept. The used terms are homogenised, i.e. unambiguous terms are chosen for semantically identical ratios. Meta-data about reports, such as report name, short description, ratios and dimensions, report receivers, person in charge, frequency and application that produces the report, are also part of the information model. Hence, they are a starting point for several analyses. Figure 4 depicts a report that has been recorded within the as-is analysis of the retail case and its conceptual representation.

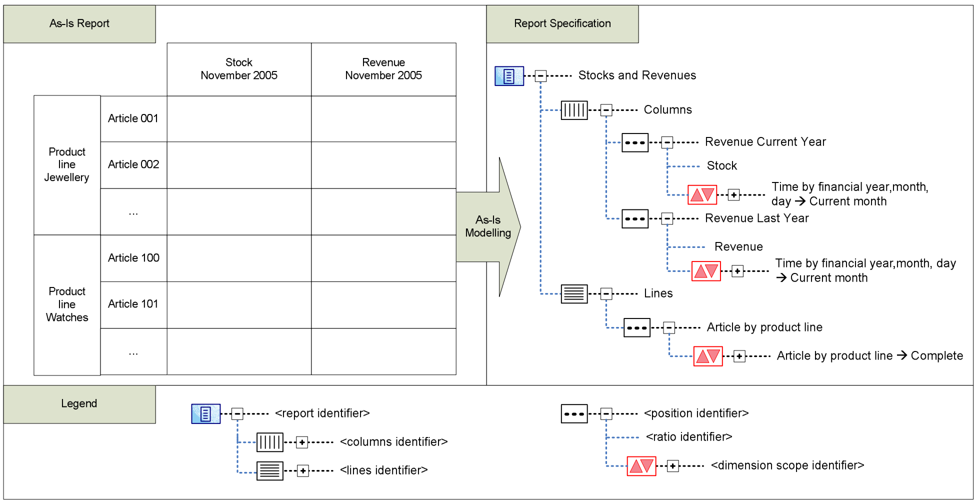

Figure 4: Report identified within as-is analysis

Analyses and classification (2-5)

The information model can be used for analyses and classification. The result is a so-called information map (Winter and Strauch, 2004). The information map can be used within the to-be modelling in several ways, e.g. as a communication basis or for the exposure of redundancies and deficiencies. Examples for questions that can be answered by analyses are:

- Who are the receivers of a particular ratio?
- Who are the receivers of particular reports?
- What reports contain the same ratios / dimensions?
- What applications produce what reports?

Analyses and classifications alone cannot be used to assess the reporting system's quality regarding content. An information requirements analysis has to be conducted to determine whether it meets the information requirements or not – and this is part of to-be modelling. Reports can be classified by the following attributes: receiver, person in charge, frequency, application, etc. Another possibility is the classification by reference objects. This way e.g. reports regarding certain products or branches can be grouped. Through this classification report groups are constructed. These groups can be used as a starting point for to-be modelling. In our retail case report groups such as stock and revenue lists or forecasts were identified. Furthermore, we could identify redundancies: many reports contained the same ratios and dimensions and thus the same information.

Review of ratios (2-6)

Formulas have to be reviewed with regard to their correctness. Usually, this requires a close cooperation with operating units. Under certain circumstances it can be necessary to examine the source systems to find calculation expressions. This is a costly task. Hence, it has to be decided from case to case if a detailed analysis has to be conducted. Errors are documented in a list. In our case study especially synonymous terms for ratios were found, e. g. "Revenues" and "Sales" were used to express the same issue. It turned out that in some cases the same ratio was calculated in a different way which lead to inconsistencies in the current reporting system.

Review of dimensions (2-7)

Dimensions have to depict the actual situation within the organisation. This can be examined within a discussion with report receivers and the persons in charge or through an examination of source systems. Wrongly depicted issues are documented in a list. In the retail case, for example, some branches within the according dimension could be identified that did not exist anymore. Furthermore, the discussion of the dimensions based on the conceptual models led to a sound mutual understanding of the organisational structure by both IT analysts and report receivers.

To-be modelling

Preparation of To-be modelling (3-1)

Analogical to as-is analysis, at the beginning of to-be modelling the level of detail is to be determined. To-be models are the foundation for a *reporting inventory* which has to be maintained continuously. On the one hand, more detailed to-be modelling leads to more expansive maintenance of the reporting inventory. On the other hand, a high level of detail allows a detailed specification of information requirements and fosters the implementation of the system. Furthermore, modelling conventions have to be defined. This is not necessary if the same modelling language is used both in as-is analysis and in to-be modelling and if modelling conventions have already been defined. In the retail case it was decided that a comprising to-be-model had to be constructed that both depicted data warehouse structures and specific reports.

Revision and adaptation of dimensions (3-2)

This activity aims to revise and adapt the analysed dimensions. This is to eliminate inconsistencies and dimension objects that are not used. This activity does not aim to determine the to-be state but at eliminating businesslike errors from as-is models. In our example, several branches were part of the according dimension but did not exist anymore and thus had to be eliminated.

Determination of to-be state (3-3)

To determine the to-be state different methods of information requirements analysis can be used (demand-driven, goal driven, etc.). If and to what extent as-is models are used to determine the to-be state has to be decided individually. In our example the *branch* dimension was enhanced as required in activity (2-7). Independent of the concrete approaches to be used, we propose to conduct the following construction tasks:

- Conceptual design of information spaces
- Identification of ratios and conceptual design of ratio systems
- Conceptual design of user adequate views onto information spaces, i. e. of reports

Within the conceptual design it has to be avoided that new synonyms and homonyms emerge. This can be ensured by maintenance of the glossary. If new elements are needed, it has to be checked initially if these are already part of the to-be model.

In the retail case the determination of the to-be state was conducted through interviews with report receivers based on as-is models. Based on the as-is analysis (especially the identified report groups) and interviews different information spaces were identified that built up the basis for the definition of reports. Figure 5 shows a central stock and revenue information space and a conceptual model of a report. The report shows for every store stocks and revenues for every product. This is expressed by the hierarchical order of the positions “Branch” and “Product”.

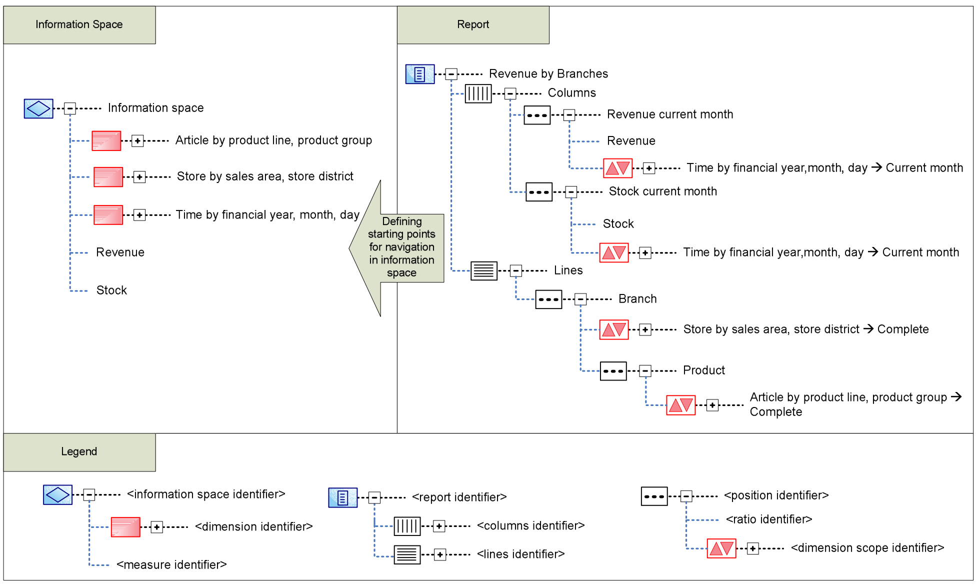

Figure 5: To-be information space and report

Consolidation with as-is state (3-4)

Through the consolidation with the as-is state it is ensured that relevant information, which had already been considered within the as-is state, are not forgotten within the to-be modelling. The consolidation is conducted based on the as-is state. Weaknesses of the as-is state, especially mistakes, inconsistencies and redundancies can be avoided since they have already been identified and recorded within as-is analysis. However, there is the risk that non-relevant information is transferred to to-be modelling which leads to a new information overflow. In our case study, the result of the consolidation was that all relevant information had been regarded to in the to-be modelling.

Analysis of to-be modelling (3-5)

The analysis of the to-be modelling serves as a quality inspection. The conceptual model is compared to the goals that have originally been defined. If the goals have not been accomplished the conceptual model has to be revised in an iterative process. Under certain circumstances an approach based on prototyping can be applied. Interviews with report receivers and controlling led to the conclusion that the original goals were met by the conceptual model.

Monitoring

Model maintenance (4-1)

The development of a reporting system is not completed once it is implemented. Information requirements are constantly changing. Consequently, there is a need for a continuous adaptation of the reporting. Hence, the conceptual model should not be the starting point for a singular implementation. It should rather be adapted continuously and serve as a topical reporting inventory. Many reasons that lead to a reporting that does not meet information requirements can be avoided through such a continuous documentation.

CONCLUSION AND OUTLOOK

The case study has been conducted in compliance with the procedure model we introduced in this paper. We have shown that a detailed guideline for the process of information requirements engineering incorporating several iteration steps and supporting a conceptual modelling language can largely support the development of reporting systems. The use of the conceptual modelling language has fostered communication both in as-is analysis and to-be modelling. Controlling, report receivers and IT analysts used the models intensively for discussion purposes. Through analyses, redundancies, weaknesses and mistakes could be identified. Reports were classified and report groups were generated. These report groups served as a starting point for to-be modelling. Parts of the as-is models were re-used for to-be modelling and thus modelling cost could be reduced. The established report inventory can be adapted continuously and serve as a topical reporting inventory. This inventory can be used for regular analyses and thus for a continuous monitoring to avoid redundancies, for example. Regarding our research some limitations have to be pointed out. We have no opportunity to compare the project outcomes with results that would have been generated using a different method in the same setting. Consequently, we do not claim general validity and applicability. The method (procedure model and conceptual modelling language) as presented here is a result of an iterative, consensus-oriented construction process. Within our case studies we also do not aim to develop “the best” solution but try to achieve a high-consensus between the different stakeholders. Procedure model and conceptual modelling language will be evaluated in further projects. Furthermore, we will take a closer look on domain specific requirements to the conceptual language. We also aim to develop (domain specific) reference models for different industries to further reduce modelling cost.

REFERENCES

Becker, J., Brelage, C., Dreiling, A. and Ribbert, M. (2004) "Business Process-driven Information Requirements Engineering" in *Proceedings of the 2004 Information Resource Management Association International Conference*, New Orleans.

Becker, J., Dreiling, A. and Ribbert, M. (2003) "Using Management Objectives to specify Management Information Systems - A Contribution to MIS Success" in *Proceedings of the Wirtschaftsinformatik 2003*, 313-323.

Becker, J., Janiesch, C., Pfeiffer, D. and Seidel, S. (2006) "Evolutionary Method Engineering - Towards a Method for the Analysis and Conception of Management Information Systems" in *Proceedings of the 12th Americas Conference on Information Systems*, Acapulco.

Bulos, D. (1996) A New Dimension. OLAP Database Design, *Database Programming & Design,* 9, 33-37.

Carter, M.P. (1983) Determining information needs, *Management Decision,* 21, 45-51.

Delfmann, P., Janiesch, C., Knackstedt, R., Rieke, T. and Seidel, S. (2006) "Towards Tool Support for Configurative Reference Modelling- Experiences from a Meta Modeling Teaching Case" in *Proceedings of the 2nd Workshop on Meta-Modelling and Ontologies (WoMM 2006)*, Karlsruhe, Germany.

Gabriel, R. and Gluchowski, P. (1998) Grafische Notationen für die semantische Modellierung multidimensionaler Datenstrukturen in Management Support Systemen, *Wirtschaftsinformatik,* 40, 493-502.

Golfarelli, M., Maio, D. and Rizzi, S. (1998) The Dimensional Fact Model - A Conceptual Model for Data Warehouse, *International Journal of Cooperative Information Systems,* 7, 215-246.

Golfarelli, M. and Rizzi, S. (1998) "A Methodological Framework for Data Warehouse Design" in *Proceedings of the ACM First International Workshop on Data Warehousing and OLAP (DOLAP 98)*, Washington D.C., 3-9.

Holten, R. (2001) *The MetaMIS Approach for the Specification of Management Views on Business Processes,* Arbeitsberichte des Instituts für Wirtschaftsinformatik, Westfälische Wilhelms-Universität Münster, Institut für Wirtschaftsinformatik, Münster.

Inmon, W.H. (1991) For Managers Only – A Tale of Two Cycles: Iterative development in the information warehouse environment, *Database Programming & Design,* 12, 73-75.

Inmon, W.H. (1996) *Building the Data Warehouse*, New York et al.

Kimball, R. (1996) *Data Warehouse Toolkit - Practical Techniques for Building Dimensional Data Warehouses*, New York.

Knackstedt, R., Seidel, S. and Janiesch, C. (2006) "Konfigurative Referenzmodellierung zur Fachkonzeption von Data-Warehouse-Systemen mit dem H2-Toolset" in *Proceedings of the DW2006 - Integration, Informationslogistik und Architektur*, Friedrichshaven, Germany.

List, B., Bruckner, R.M., Machaczek, K. and Schiefer, J. (2002) "A Comparison of Data Warehouse Development Methodologies - Case Study of the Process Warehouse" in *Proceedings of the 13th International Conference on Database and Expert Systems Applications (DEXA 2002)*, Aix-en-Provence, 203-215.

Poe, V. (1996) *Building a Data Warehouse for Decision Support*, New Jersey.

Rockart, J.F. (1979) Chief executives define their own data needs, *Harvard Business Review*, 81-93.

Rosemann, M. (2000) Using Reference Models in the ERP Lifecycle, *Australian Accounting Review,* 10, 19-30.

Royce, W.W. (1987) "Managing the Development of Large Software Systems" in R. H. Thayer (ed.) *Tutorial: Software Engineering Project Management*, Los Alamitos, 118-127.

Sapia, C., Blaschka, M., Höfling, G. and Dinter, B. (1998) "Extending the E/R Model for the Multidimensional Paradigm" in *Proceedings of the International Workshop on Data Warehouse and Data Mining*, Singapur, 105-116.

Watson, H.J. and Frolick, M.N. (1993) Determining Information Requirements for an EIS, *MIS Quarterly,* 17, 255-269.

Winter, R. and Strauch, B. (2004) "Information Requirements Engineering for Data Warehouse Systems" in H. Haddad (ed.) *Proceedings of the 2004 ACM Symposion on Applied Computing*, Nicosia, 1359-1365.

Yin, R.K. (2003) *Case Study Research - Design and Methods,* Sage, Thousand Oaks, CA et al.

Zachmann, J.A. (1982) Business Systems Planning and Business Information Control Study: A comparison, *IBM Systems Journal*, 31-53.

COPYRIGHT