Reference Model Maintenance Based On ERP System Implementations

Axel Winkelmann

PD Dr. Axel Winkelmann received his PhD and Post-Doctoral Qualification (Habilitation) at the University of Muenster, Germany. In 2009, he was appointed as an interim professor on Business Information Systems at the University of Koblenz-Landau. He is currently an interim professor at the institute of Information Systems at the University of Muenster and a researcher at the European Research Center for Information Systems (ERCIS). His research interests includes Business Information Systems, Business Transformations and Business Process Management with emphasize on business domains such in service sectors like retail, finance, and telecommunication. He has written numerous books and international conference and journal articles on these topics and is engaged with various research institutions and universities.

Conceptual reference models in the narrow sense are detailed descriptions of information processes in commercial domains such as retail, manufacturing, etc. They serve many usages such as business process reengineering, information systems development or business software selection. Hence, it is necessary that reference models offer the latest domain knowledge. Al-though there is much literature on the initial reference model creation, hardly anything has been said about the maintenance of reference models. This paper introduces a procedure model for updating reference models with regard to the implicit knowledge that is implemented in ERP software. The model uses the domain-specific know-how acquired by software development companies. Based on implemented ERP functionality, the procedure model derives domain practices for the maintenance of the reference model.

1. Introduction

Many companies have introduced Enterprise Resource Planning (ERP) systems in order to stay competitive and to improve and change their business strategies [21; 26; 2]. These are defined as information systems that support integrated core business processes on a single integrated database [28]. In general, ERP software is standard software, also named COTS (commercial off-the-shelf) software, which has been developed as general domain software for one or many different domains in consideration of "best practice." ERP systems offer a broad spectrum of functionality and various alternate supported processes. They offer customizing mechanisms for the specific need of each company [15; 31]. Hence, these standard systems are very flexible due to customizing possibilities and can be adapted to market needs of individual companies [8; 23]. Furthermore, individual companies are able to gain from the ERP inherent processes and structures by adopting these "best practices" [18; 31].

Implemented "best practices" originated from previous ERP projects as the ERP manufacturer has im-plemented domain know-how from various ERP do-main projects or working groups consisting of ERP ma-nufacturer and users from various companies. ERP ma-nufacturers also gain their knowledge from academic li-terature in terms of reference models and descriptions of industry and case scenarios [17; 29; 30]. In this context, reference models offer valuable domain knowledge that is used e.g. for the creation and further development of standard software. Models are established as interface definitions or frameworks (e.g. [3; 7]). Therefore, they standardize the outside view of company functions ("What"?). Particularly in the academic context, reference models are created as abstract descriptions of the inside view of processes ("How?"). In this article, the latter are reckoned as reference models in the narrow sense due to their higher level of domain and company details. Normally, reference model design and software development exist separately from each other, although the implementation of functions and business processes may be based on the model. Ideally, the reference model is fully representable in the ERP system. However, individual customer requirements go far beyond these standardized reference models that are generally applicable. A study among 27 ERP manufac-turers shows that approximately 50% of new features in standard EPR system products result directly from cus-tomer requirements [29]. In particular, so-called reference customers are at the forefront of this ERP maintenance and provide important and current requirements to the software houses.

Although reference models are very important for the development of ERP systems, their maintenance and updating seem to be very difficult. The problem does not occur with reference models in the larger sense, especially interface definition models. They are deve-loped by consortia of companies, because the members have an active interest in establishing intercompany communication standards. However, reference models in the narrow sense, especially domain-specific models, are excluded from these knowledge-gaining feedback cycles because companies do not have an active interest in providing their specific process implementations to the public. Having this in mind, the article strives for a procedure model that allows maintaining reference models according to the recent domain knowledge that is implemented in standard ERP systems. The article introduces a procedure model for the analysis of ERP inherent domain and process know-how and its alignment with and incorporation into domain reference models.

2. State of Research in Reference Modeling

Numerous authors have published works on the usage of reference models as tools for the development of organizations and applications (see e.g. [10, 11] for an overview). Assuming that domain knowledge can be explicated universally for a certain class of companies, reference models are a good starting point for the development of individual business processes [5]. The aspect of reuse in reference modeling especially aims at cost reduction in the context of individual modeling and implementation projects. On the one hand, this is achieved by time saving resulting from the reuse of a large base of models, and on the other hand by an implicit reduction of risk by applying reference model inherent business processes that have been proven and used many times already. As the reference model offers very detailed and generalized information from a domain, its initial development costs exceed the costs of developing an individual model [12]. Hence, a prerequisite for the realization of cost benefits is the actual multiple reuse of such a reference model. Reference models have to be adapted to each individual need in order to gain a competitive advantage from its usage. This differentiation between a universally valid reference model and the company-specific organizational model is generally directly reflected in the standard ERP systems: these systems offer relatively stable and generalized system cores, which are adapted in individual customer projects through parameterization and customization to the respective requirements [29]. For that, it is reasonable that ERP software vendors use reference models as a conceptual reference point for their ERP products. The models can serve as requirement definitions but they can also be an important part of the system documentation through their abstraction from details of implementation. For this case, three central requirements exist: First, reference models have to offer additional value. Second, they have to provide a fine-granular level of detail. Only with an adequate description of the respective application standard of the represented company class are reference models usable as conceptually functional blueprints. Third, reference models have to be maintained in order to describe the recent state of the art in process design, e.g. in goods and materials management. Old, out-dated reference models may display traditional processes and structures within the company class, but are not considered relevant e.g., technological innovations and changes. In the context of scientific discussions, researchers addressed the reference model design [1, 14, 27], and the variant management in the context of the adaptive reference modeling [9, 16]. In addition to the technical development of languages and procedure models for reference modeling, a number of reference models have been developed that respectively review domain knowledge (e.g. [6, 22]). However, the maintenance cycle for refe-rence models in the narrow sense has not been addressed so far.

3. Research Objectives

With regard to domain-oriented reference models, many models suffer from a lack of updates or are only updated in small proportions beyond the initial development project. As the cause of this development, two key problems could be identified: lack of methodic support of the incremental updating of reference models, as well as restricted access to current domain information. Process implementations with outstanding performance that are not to be standardized are considered a competitive advantage. Hence, a publication of that know-how is not in the interest of such organizations. Such process knowledge is usually barely acquirable for (academic) reference model developers unless they have an institutionalized knowledge exchange with domain experts (e.g., within a research project).

The aim of this paper is the development of a procedure model for domain-oriented reference model maintenance based on the analysis of ERP systems. It is primarily a design-oriented approach, which derives an artifact from an identified problem (Section 4). The artefact is distributed as a possible solution to the problem (see [13]). The case example in section 5 represents and explains an extract of the evaluation.

4. Procedure Model for ERP based Reference Model Maintenance

4.1. Conditions

Contrary to research for reference model initiations, domain information for reference model maintenance will not be gathered directly by application partners (or affiliated via BPM methods), but indirectly deduced from existing domain-oriented ERP software. The procedure model is based on the assumption that the functional information can be derived from the user inter-face. It draws upon methods that allow such a derivation for process models and data models in the opposite direction out of ERP systems (e.g. [4, 25]). Appropriate ERP systems offer market stability in terms of a high user base on the one hand but also flexibility in order to react to changes from their respective application companies on the other hand. The procedure model requires three conditions.

C1: There is an initial reference model, which allows for deriving structured problems and task descriptions.

C2: The ERP producers have an active interest in the evaluation of their systems by third parties.

C3: There are an appropriate number of systems available for the reference model maintenance.

Condition 1 implies that a reference model exists, which prestructures the investigated domain to a sufficient extent and sufficient granularity.

Condition 2 has to ensure that ERP vendors participate in the system evaluations because "living" systems (with sample data) have to be accessed. In research projects for reference model creation, information supply by application partners is normally specified through the project descriptions. However, for the maintenance of reference models, researchers have to live without any formal contracts or responsibilities of application partners. As information is gathered through the actual use of the system, a usability report can be created in addition to the reference model. This usability report offers additional value to the ERP manufacturer.

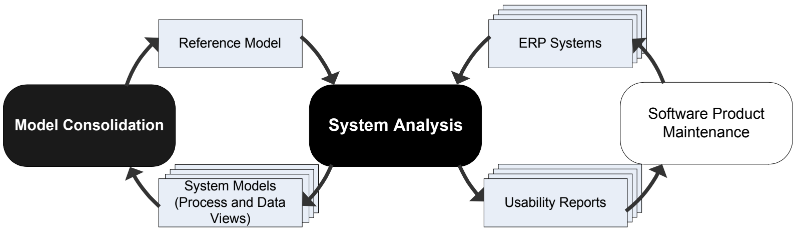

Figure 1: External View of the Procedure Model

Condition 3 is derived from the postulation that a reference model represents an abstraction from individual cases. Therefore, several systems are subject to investigation. The number depends on the research focus. For example, the procedure model offers the opportunity to revise only small parts of the reference model.

4.2. External View (Procedural Model)

At the top level, the procedure model is formed as a double loop (cf. Figure 1), which covers the reference model maintenance and the enhancement of the respective ERP systems.

It is important to notice that the procedure is not a closed, exclusive procedure for maintenance and further development. It is an addition on the sides of the model and software product maintenance. For example, important inputs for reference model maintenance can also be attributed to case studies or expert interviews [1]. On the side of the ERP system the procedure subordinates, e.g. the use of new technologies or the competitive analysis as a driver for further development. The initial reference model is the starting point of the procedure. The two core processes of the procedure are the system analysis and the model consolidation (black boxes in Figure 1). As the software product maintenance is a consequence of reference model maintenance and/or system analysis, it is not the focus of the article.

4.3. Internal View

In the context of ERP system analysis (cf. Figure 2), a set of case studies will be examined. The cases are typical scenarios of the domain and serve as a background skeleton for the analysis. Coarse granular tasks, as well as context parameters (data, preferences, etc.), have to be derived from the initial reference model. In that way, a uniform context for the analysis of each system is assured.

Modelers try to solve the cases with the help of the accessible ERP systems by searching for alternative solution paths within each system. Solution paths should be formally documented. The interaction patterns will be formalized as process models with suitable description languages (preferably with the process description language of the initial reference model such as event-driven process chains (EPC) or UML diagrams). Data requirement derivations based on data entry masks and attributes will be formalized with the help of data modeling languages such as Entity Relationship Models (ERM) or UML diagrams. It should be stressed in this context that the identified data structures are only extracted from the user interface - an analysis of actual data patterns of the underlying persistence layer cannot be conducted. As a disadvantage, the analysis based on user interface findings does not disclose the full potential of an ERP system. As an advantage, the superficial examination regards the actual system behavior whereas the underlying data structure does not. For

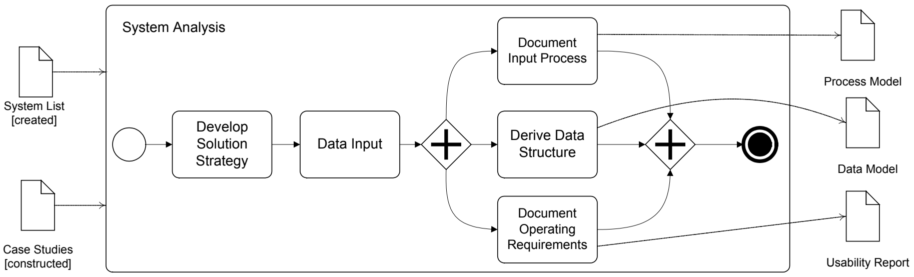

Figure 2: Refinement of Sub Processes of System Analysis

that, development decisions based on platform inherent reasons are ignored.

The derivation of the data and process structures from the input sequences and UI screens is due to heuristics. Control and user interface models are allocated to model structures. Table 1 shows a corresponding excerpt of heuristics for the derivation of process structures and data structures from ERP systems.

4.4. Model Consolidation

Within model consolidation, modelers have to decide, which model structures to add to the reference model. According to Pfeiffer and Gehlert [20], after the removal of language and structural conflicts, a semantic model comparison is accomplished, which discloses extensions or reductions in the reference model. This step usually has to be performed manually because an uncritical adoption of findings from the ERP systems may risk the reference model quality. Fast moving conceptual fashions may risk the quality of the reference model. Hence, for every new insight, modelers need to decide in a critical discourse whether or not a new change should be incorporated into the reference model. In this way, new model structures are integrated into the reference model and obsolete parts are eliminated. Due to the domain-specific characteristics of each reference model and its individual objectives, it is not possible to define universally valid semantic adoption criteria in advance. In critical discussions, model maintainers and domain experts have to decide on the adoption of new model elements based on the number of appearances in the ERP systems or the importance of the individual ERP system in which the domain knowledge appears.

5. Applying the Procedure Model

5.1. Reference Model Selection

Using the example of a reference model from retail, such an approach is explained subsequently. The retail-H was originally published in 1996 and slightly updated in 2004 [6]. It is a domain-oriented reference model for the construction of ERP systems in retail that offers very detailed functional and process models as well as data models. For instance, the retail-H subdivides the article construct into three specializations, Article Supply, Article Storage and Article Distribution.

Derivation of Process Structures		Derivation of Data Structures	
Model Structure	Operational Pattern	Model Structure	UI Pattern
Activity	Individual input mask Completed with save operation Input must be terminated …	Object Class / Entity type	Element has a dedicated input mask Element is existence independent Element has complex data type Element is referenced by other elements ...
Sequential Activities Flow	Successive input masks Existence dependencies …	Attributes / Object property	Element is specified by input Element has primitive data type Element is existence dependent Element is instantiated once …
Exclusive Branching	Certain input fields will be locked after selection …	Relationship types/ Association	Feature is specified with search field via selection field Feature is presented in a list

Table 1: Heuristic Derivation of the Model Structures (Excerpt)

In addition, different types of articles are to distinguish, depending on the property of purchase or sale. Besides Single Articles, which must be provided with appropriate attributes like article description, price, suitability for storage and so on, there exist Article Variants (e.g. two colors of same trousers), which refer to different articles, but same article description, etc.

Article Product Group serves as a collection group for multiple articles of one product group that are not sold on an individual basis. Articles, classified as Article Season are articles that are not sold regularly, but only bought and sold at a specific time. In these cases, additional logistical information has to be stored within the master data. Articles with a recursion relationship are allocated to Article Sets that are individual compilations of sales units consisting of diverse single articles. On the supply side, a recursion relationship is allocated to Article Lots, which means a collection of sales units to an obligatory total purchase amount. Article Displays consider articles that are combined for sale (e.g. in cardboard stand-ups), but are sold individually. Article Empty characterizes a multihierarchy bill of material (BOM), e.g. eleven 1-liter Coca-Cola bottles consisting of the actual product, eleven empty bottles and the packaging case. This differentiation is necessary for returning bottles, for example.

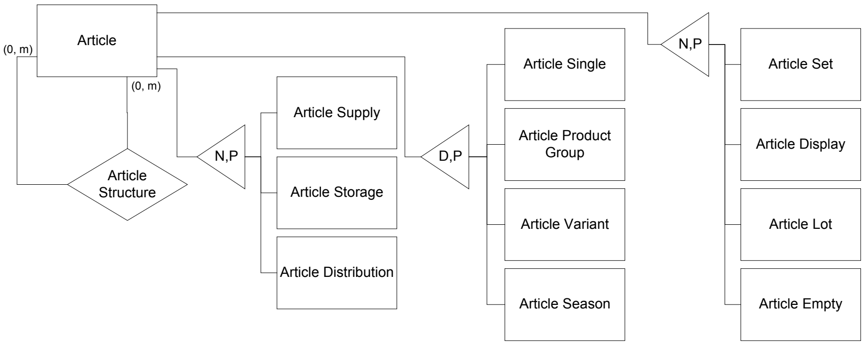

Figure 3: Reference Model Excerpt – Specialization of the Construct Article (cf. [6])

5.2. System Analysis Accomplishment

Two modellers collected and evaluated processes and data in a row of ERP systems independently on requirements derived from a case study. Exemplarily, the procedure will be

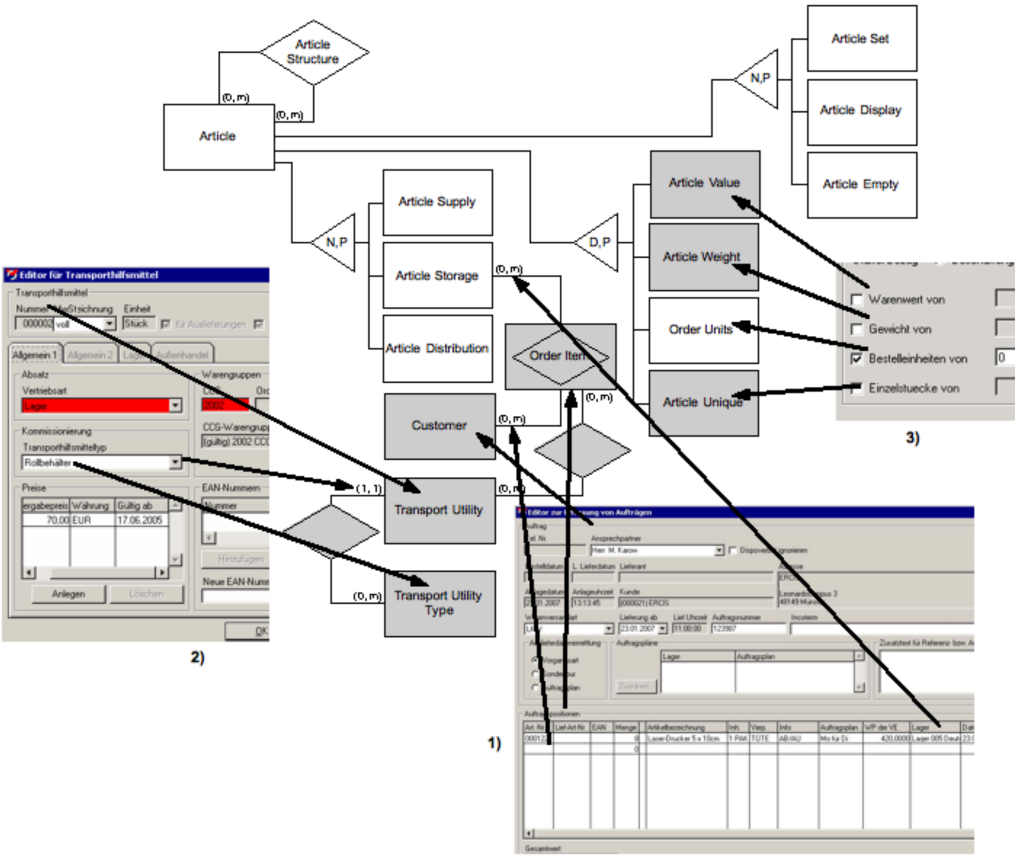

Figure 4: System Analysis of an ERP System (System A) with Specialization on Food Retailing (Excerpt)

demonstrated on two systems (A and B), which were analyzed in order to update the reference model. Figure 4 shows an excerpt of the system analyses for the derivation of the article data model from different masks of system A as an example of the procedure. Within the data model, new entity types are displayed in light grey and the original reference model constructs are displayed in white.

On the mask Order Item ("Auftragserfassung", cf. 1 in Figure 4) it is possible to allocate numerous articles to each inquiry. In parallel, several Order Items can be assigned to each Article Storage (note that each article instance can only be sold once). Each Order Item can be transported with one or many Transport Utilities. Transport Utility Types can be trucks, trains, planes, etc. (cf. "Editor für Transporthilfsmittel", 2 in Figure 4).

In addition to article units, that are called "Order Units" in the present system, it is also necessary to use Article Value, Article Weight and Article Unique (for actions or special orders). All three article types are derived from the analysis of the mask "article setting up" (cf. 3 in Figure 4). Each article type is not handled as an attribute but as its own entity type because specific additional information such as monetary units or certain grading may be necessary depending on every specific article.

5.3. Model Consolidation

The reference model will be enhanced based on the results of the system analyses. Therefore, all findings from the ERP systems (see both left models in Figure 5) will be consolidated and suitable extensions to the existing reference model will be incorporated into the maintained reference model (see right model in Figure 5).

In food retailing (ERP system A), there are different requirements for buying and selling. Weight articles such as cheese and value articles such as a 100 Euro collection of one Euro articles are traded. Unlike unit order articles, they are not counted with specific amounts but in accordance with their weight or value. They are substitutes for article groups without accurate inventory management. For example, weight articles are articles that are stored and sold with regard to their weight (e.g. wheels of cheese). Unfortunately, they suffer from shrinkage of weight by evaporation, which in turn requires intelligent mechanisms of deduction. Unique articles, which are sold only once (e.g. promotional articles), have to be treated differently because retailers do not want to enter much article master data (e.g. supplier data) and do not want to store the data for a long time in their ERP systems.

ERP system B is particularly suitable for the production, especially job production. As such, value and weight of articles are of great importance. Furthermore, succeeding articles are necessary in order to define sub-sequent articles once an article is not produced any-more. Also alternative articles have to be specified. These articles can be used for production alternatively to the original articles. Article sets can either be Selling Sets consisting of fixed product bundles or Price Sets consisting of fixed article quantities from a variety of articles.

These identified entity types can be used for the ex-tension of the retail-H reference model. It is possible either to add enhancements that are noted simultaneously in different ERP systems only (here: Article Weight, Article Value; dark grey entity types in the right model of Figure 4) or to add all enhancements that can be significantly contribute to the reference model according to expert opinions (Article Weight, Arti-

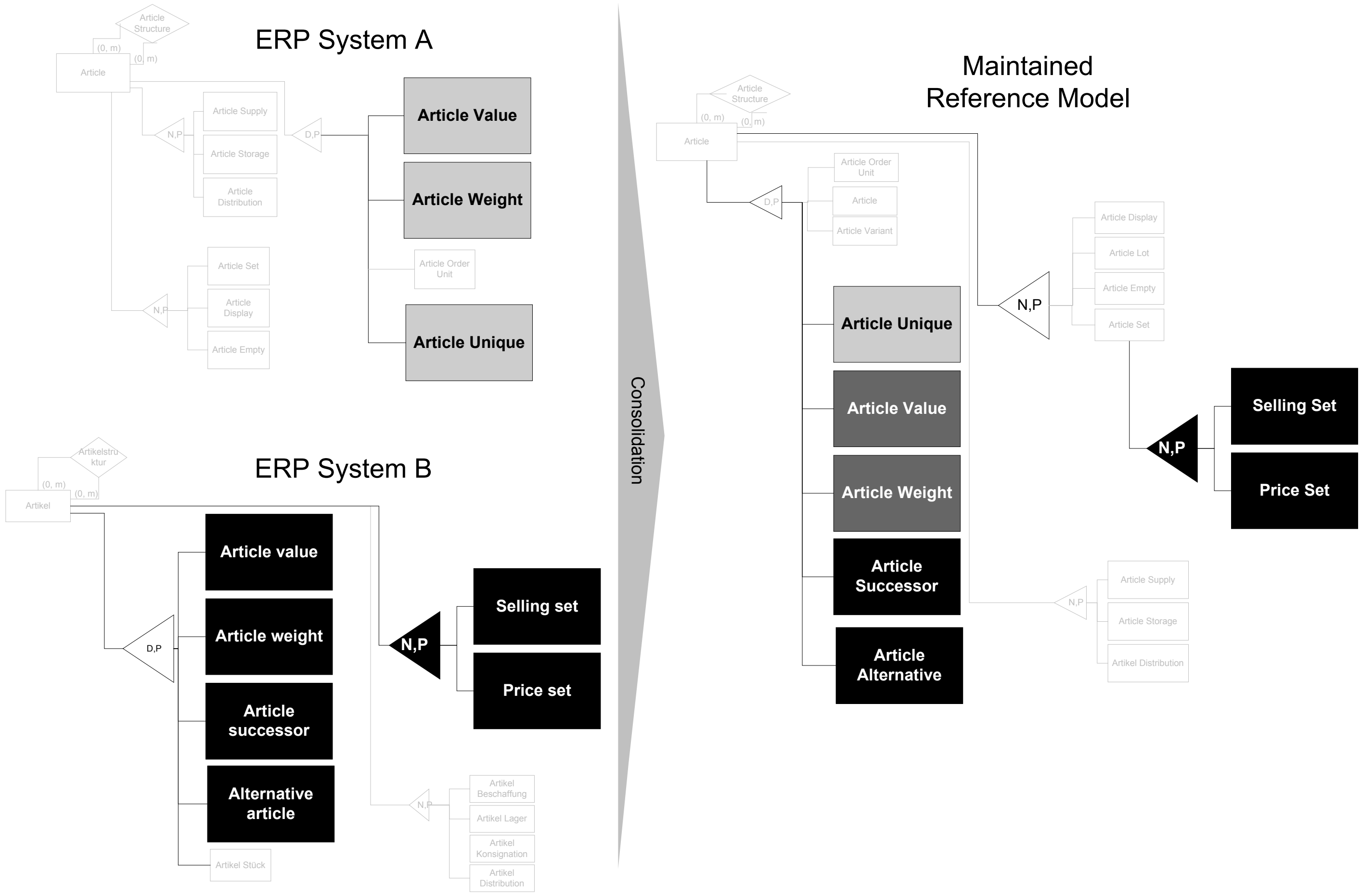

Figure 5: Model Consolidation (Excerpt)

cle Value, Article Unique, Article Successor, Article Alternative, Selling Set and Price Set). Although a model enhancement due to repeatedly observed model elements promises to result in more objective reference model enhancements, there is the potential danger that the reference model enhancements will only display "common practice." Innovative concepts in terms of "best practice" used by only one ERP system may not be incorpozated into the reference model that way.

6. Conclusion, Limitations and Outlook

The present article proposes a first procedure model for the maintenance of conceptual reference models. It tries to overcome the difficult access to domain information once a reference model has been initially created. The procedure model builds additional reference knowledge on ERP user interfaces only. An extension towards underlying concepts (e.g. object models) is conceivable but has not been tested so far. As an advantage, the procedure model allows ongoing maintenance of existing reference models without significant entry barriers, such as comprehensive project organization or contractual assurances or liabilities. The contribution is a complementation to the comprehensive methods of reference model design.

Although the procedure model is to some extend very pragmatic and easy to apply, there are also some limitations on its usage:

First, there has to be an existing reference model with a suitable granularity and a sufficient model size as a starting point for the maintenance. This may well be the case for food retail or production in general but may not be the case for specific domains like furniture retailing, although there are many ERP systems available for that domain.

Second, suitable ERP systems have to be identified in order to gain additional knowledge for the domain of the reference model. In addition, experienced modeling and ERP experts are necessary for the analyses of the ERP systems.

Third, the procedure model – as well as the reference model maintenance – may suffer from the subjective expert views on the underlying ERP systems and the elements that will be incorporated into the reference model. However, this is a general problem of reference modeling because it is hardly possible to achieve a true understanding when designing and maintaining a reference model. As a significant advantage, the acquired knowledge from the ERP systems has already run through various iterations and validations on the ERP market and therefore can be seen as widely accepted.

Fourth, the knowledge gain through ERP system analyses is mainly limited to "common practices" and may represent a low potential for the innovative development of systems based on maintained reference models. However, the maintained models are well suited for the purpose of software selection and documentation, as well as a structured, technical description of the core processes and information objects of a domain. They can potentially realize significant benefits.

The use of the procedure model outside of the ERP system domain is most likely possible but has not been tested yet and is subject to future work. Hence, further research includes the ongoing validation of the procedure model in different professional domains and with different reference models and IT systems. In addition, the integration of the presented approach into a comprehensive reference model design method and therefore the methodical coverage of the entire model life cycle is an important challenge for future research efforts.

References

[1] Ahlemann, F., Gastl, H. (2008). Process Model for an Empirically Grounded Reference Model Construction. In: P. Fettke and P. Loos, (eds.), Reference Modeling for Business Systems Analysis, Idea Group, 77-97.

[2] Arif, M., Kulonda, D. J., Proctor, M., Williams, K. (2004). Before you invest: An illustrated framework to compare conceptual designs for an enterprise information system. Information Knowledge Systems Management, 4(1), 119-135.

[3] ARTS (2009), Association for Retail Technologie. Available: http://www.nrf-arts.org/. Last accessed: 08-12-2009.

[4] Bajaj, A. (2005). Inferring the User Interface from an EER Data Schema. Proceedings of Americas Conference on Information Systems, Acapulco.

[5] Becker, J., Delfmann, P., Knackstedt, R. (2006). Adaptive Reference Modeling: Integrating Configurative and Generic Adaptation Techniques for Information Models. Proceedings of the Reference Modeling Conference (RefMod).

[6] Becker, J., Schütte, R. (2004). Retail Information Systems. Frankfurt/Main: Redline Wirtschaft, 2nd Edition (in German).

[7] Bolstorff, P., Rosenbaum, R. (2003). Supply Chain Excellence. Amacom, New York.

[8] Davenport, T. (2000). The Future of Enterprise System-Enabled Organizations. Information Systems Frontiers, 2(1), 163-180.

[9] Delfmann, P. (2006). Adaptive Reference Modeling. Logos, Berlin (in German).

[10] Fettke, P., Loos, P. (2004). Research in Reference Modeling. Wirtschaftsinformatik, 46(3), 331-340 (in German).

[11] Fettke, P., Loos, P. (2007). Reference Modeling for Business Systems Analysis. London: Idea Group.

[12] Frank, U. (2007). Evaluation of Reference Models. In: P. Fettke and P. Loos, (eds.), Reference Modelling for Business System Analysis, Idea Group.

[13] Hevner, A. R., March, T. S., Park, J., Sudha, R. (2004). Design Science in Information Systems Research. MIS Quarterly, 28(1), 75-105.

[14] Karow, M., Pfeiffer, D., Räckers, M. (2008). Empirical-Based Construction of Reference Models in Public Administrations. Proceedings of the MKWI, Munich.

[15] Klaus, H., Rosemann, M., Gable, G. G. (2000). What is ERP? Ìnformation Systems Frontier, 2(2), 141-162.

[16] Knackstedt, R., Janiesch, C., Rieke, T. (2006). Configuring Reference Models - An Integrated Approach for Transaction Processing and Decision Support. Proceedings of the 8th International Conference on Enterprise Information Systems (ICEIS), 135-143.

[17] Markus, M. L., Tanis, C. (2000), The enterprise systems experience: From adoption to success. In: R. W. Zmud, (ed.) The domains of IT research: Glimpsing the future through the past. Pinaflex Educational Resources, 173-207.

[18] Nah, F. F.-H., Zuckweiler, K. M., Lau, J. L.-S. (2003). ERP implementations: Chief Information Officers' Perceptions of Critical Success Factors. International Journal of Human-Computer Interaction, 16(1), 5-22.

[19] Ng, J. K. C., Ip, W. H., Lee, T. C. (1999). A Paradigm for ERP and BPR Integration. International Journal of Production Research, Vol. 37, 2093-2108.

[20] Pfeiffer, D., Gehlert, A. (2005). A framework for comparing conceptual models. Proceedings of the EMISA (Enterprise Modelling and Information Systems Architectures), 108-122.

[21] Robey, D., Ross, J., Boudreau, M. (2002). Learning to Implement Enterprise Systems: An Exploratory Study of the Dialectics of Change. Journal of Management Information Systems, 19(1), 17-46.

[22] Scheer, A.-W. (2002). Business Process Engineering. Reference Models for Industrial Enterprises. 2nd Edn., Berlin: Springer.

[23] Shanks, G., Seddon, P., Willcocks, L. (2003). ERP; The Quiet Revolution. In: G. Shanks, P. Seddon, and L. Willcocks, (eds.), Second-Wave Enterprise Resource Planning Systems: Implementing For Effectiveness, Cambridge: Cambridge University Press.

[24] Soh, C., Sia, S. K., Boh, W. F., Tang, M. (2003). Misalignments in ERP implementation: a dialectic perspective. International Journal of Human-Computer Interaction, 16(1), 81-100.

[25] Sousa, K., Mendon, H., Vanderdockt, J., Rogier, E., Vandermeulen, J. (2008). User interface derivation from business processes: a model-driven approach for organizational engineering. ACM.

[26] Strathman, J. K., Roth, A. V. (2002). Enterprise resource planning (ERP) competence constructs: two stage multi-item scale development and validation. Decision Sciences, Vol. 33, 601-628.

[27] vom Brocke, J., Thomas, O. (2007). Design Principles for Reference Modelling: Reusing Information Models by Means of Aggregation, Specialisation, Instantiation and Analogy. In: P. Fettke and P.

Loos, (eds.), Reference Modelling for Business Systems Analysis, London: Idea Group.

[28] Wei, H.-L., Wang, E. T. G., Ju, P.-H. (2005). Understanding misalignment and cascading change of ERP implementation: a stage view of process analysis. European Journal of Information Systems, Vol. 14, 324-334.

[29] Winkelmann, A., Knackstedt, R., Vering, O. (2007). Adoption and Development of ERP Systems. An Explorative Analysis. Münster. Available online: http://www.wi.uni-muenster.de/udoo/downloads/publications/1975.pdf. Last Accessed: 03-08-2009. In German.

[30] Winkelmann, A.; Leyh, C. (2010). Teaching ERP systems: A multi-perspective view on the ERP system market (accepted for publication). In: Journal of Information Systems Education.

[31] Winkelmann, A. (2010). Dynamic Reconfiguration of ERP Systems - Design of Information Systems and Information Models. Post-Doctoral Thesis, Münster 2010.

Prof. Dr. Axel Winkelmann
Julius-Maximilians-Universität Würzburg
Lehrstuhl für BWL und Wirtschaftsinformatik (II)
Stephanstr. 1
97070 Würzburg
axel.winkelmann@uni-wuerzburg.de