Association for Information Systems
AIS Electronic Library (AISeL)

AMCIS 2007 Proceedings

Americas Conference on Information Systems (AMCIS)

December 2007

Dynamic Alignment of ERP Systems and their Documentations - An Approach for Documentation Quality Improvement

Ralf Knackstedt

Axel Winkelmann
European Research Center for Information Systems (ERCIS)

Joerg Becker

Dynamic Alignment of ERP Systems and their Documentations - An Approach for Documentation Quality Improvement

Ralf Knackstedt, ERCIS, ralf.knackstedt@ercis.uni-muenster.de

Axel Winkelmann, ERCIS, axel.winkelmann@ercis.uni-muenster.de

Jörg Becker, ERCIS, becker@ercis.uni-muenster.de

In this article, we introduce an approach for a better alignment of ERP functionality and software documentation in order to support the increasing need for standard software customization. Software companies integrate more and more additional functionality into their systems. These are customized for individual customers, because they are not relevant for all users. Hence, software manufacturers are challenged to adapt their system and documentation to individual implementation variants without much effort. The article discusses the problems of documentation adaptation based on theoretical evidence. It provides a consistent and redundant-free, tool based solution for the management of documentation variants – via a Design Science approach.

Keywords: ERP, Variant Management, Documentation, Configuration, Modeling

Alignment of functionality and documentation in ERP systems

Manufacturers of ERP-systems can apply different business strategies. Many companies develop a specialized system in the first place. Development is easier to manage, because functionality is narrowed to users with very similar requirements. As a disadvantage, the user basis is very limited in this stadium of software development. For further development, software companies try to broaden the ERP-functionality in order to reach different customer segments. Table 1 shows the functional broadening of selected German ERP systems from their market entry to recent product versions. Nevertheless, not all customers need all functionalities. Therefore, the standard system is customized to each customer individually. For example, functionality is deactivated or eliminated. The provisioning of relevant functionality and the adoption of customers' needs is crucial for the acceptance of an ERP system.

Table 1. Examples for the development of ERP products

System-ID	Market entry	# of customers	Historically supported types of enterprises	Added types of enterprises
1	1997, based on MS-Navision	300	Industry independent (limited functionality)	Engineering, wholesale, gardening, mail order business, retail, breweries, beverage retail
2	1994	45	Industry independent (limited functionality)	Specialized solutions for textile, food, tools, chemistry
3	1999	450	Technical wholesale	Car spare parts retail
4	1985	900	Food, consumption goods, meat, dairy products	Chemistry, paint, logistics, pharma, cosmetics
5	1998	140	Wholesale, production	Retail
6	2000	550	Wholesale, retail	Industry independent
7	2000, based on MS-Navision	200	Co-operatives	Wholesale, retail
8	2001	200	Production	Technical retail
9	2000	1700	Production	Wholesale
10	1999, based on SAP	54	Wholesale, handcraft	Retail import
11	1992	400	Wholesale	Retail
12	1996	500	Wholesale	Shoes, food, technical retail
13	1991	400	Electronic retails	Wholesale for building materials
14	2006	8	Tyres	Fashion, car spare parts retail
15	2003	13,000 worldwide, 900 in Germany	Production	Wholesale, technical retail
16	2000	410	Direct mail business	E-commerce business
17	1996	150-200	Energy, health insurance, social insurance, banking, other insurances	Public administration, tourism, retail

In May and June 2006, we asked 96 ERP-software manufacturers (about ½ of all ERP manufacturers in Germany) to participate in a qualitative telephone survey on the software quality of their ERP products. 32 companies responded to our inquiry, which led to a telephone survey of 28 ERP-system producers (appr. 30% of all manufacturers) within the limited period. On average, the probands worked 10 years for their companies. In general, probands came from the top management or software development level and were able to answer technical as well as economical questions about their software. Most participating companies are small or medium-sized, which reflects the German ERP-market with 200-250 mainly medium-sized standard ERP-systems. All analyzed ERP-systems are classified as standard solutions, because they have at least 3-5 installations running. On average, every company has 350 customer installations, ranging from 5 to 1,700 installations. Only one out of five systems is older than 10 years. An increasing amount of older systems and architectures has already been replaced by modern systems and object-oriented architectures.

73% of the responding manufacturers do not offer any form of individual documentation tailored to the individual needs. In contrary, only 27% offer documentation adaptation. Some manufacturers argue that they do not want to customize their documentation for marketing reasons. The customers are able to read about functionality that they might want to use but have not licensed and installed so far. Although this argumentation is understandable from a coarse granular point of view it is not understandable from a fine granular view. For example, users that read about CRM functionality will easily understand that they do not have the CRM module at hand due to licensing or customization reasons. Customers that read about the functionality of a certain process or screen form will not necessarily understand why a described button or functionality cannot be used or is not visible on their GUI.

The adaptation of software raises the question of how it is possible to keep the documentation in synchronization with the functionality adaptation. Hence, documentation should only describe functionality that is available to the individual user. This means, an approach for a dynamic configuration of ERP system documentation is necessary. The paper develops a special concept for the dynamic alignment of ERP software functionality and documentation. This concept is based on sophisticated approaches of configurative reference modeling. Section 2 relates our approach to the state of the art. Section 3 presents our problem solution. A software prototype proves and evaluates the practicability and helpfulness of the developed concept. Section 4 concludes the article by analyzing our approach in the context of Design Science guidelines.

Related work

Despite complaints that suggest “Nobody Reads Documentations” (Retting (1991)), many studies have proven that most of the users of software systems use a print or online documentation (Smart et al. (2001)). Studies, which investigate information design decisions in computer software documentations, address diverse dimensions of document design with regard to the usability of software system documentation, such as the effect of typography, organization, language and choice

of medium. Studies of printed and online documentation usage have for example revealed preferences for different media according to the tasks and experience of the users (Smart et al. (2001), Schriver (1997), (Ansons (1995)). To assess the usability of software documentation the dimensions navigation, presentation, learnabilty and task support can be used. While the terminology sometimes varies, analogous dimensions to these are proposed in diverse approaches (e.g. Gillan and Bias (2001)).

Table 2. Literature review for documentation quality impacts

	Purchasing	
1	The quality of software documentation influences the **quality of software purchase decisions** (Bernroider, Koch 1999).	Alignment deficiencies lead to a wrong perception of software functionality. This may result in wrong software purchase decisions.
	Training and Support	
2	Training documentation influences **user satisfaction** (Shaw, Delone, Niedermann 2002).	Training documentation should conform with users' real business processes in order to increase the individual learning success (see also Scott 2005).
3	Informal self support among colleagues (also known as "Hey, Joe!") is a very expensive way of user support (**support costs**) (Scheithe 2004).	If users do not accept and use the documentation because of alignment deficiencies, the usage of self-support increases. Hence, increased support costs will occur.
	Usage	
4	Documentation deficiencies lead to direct **failure costs** (e.g. handling mistakes) and indirect **failure avoidance costs** such as costs for increased information seeking (Guillemette 1987).	Documentation deficits can especially be alignment deficiencies. Therefore, we conclude that alignment deficits lead to failure costs and failure avoidance costs.
5	Documentation quality influences the **usability** (Agarwal, Venkast 2002; Nielsen 1994; (Keirnan, Anschub, Rosenbaum 2002; Mayhew 1992).	Efforts to assure an alignment are seen as a contribution to documentation quality which influences the usability as stated in the literature.
6	Documentation quality influences the **efficiency of system usage** (Rautenberg, Sova 1983).	Alignment deficiencies reduce documentation quality and therefore affect the efficiency of system usage negatively.
7	Software documentation directly influences user satisfaction and **perceived system benefits** (Rupietta 1987; Ives, Olson, Beroudi 1983).	Under any circumstance, software documentation should be complete, correct and up-to-date. All aspects correlate with the alignment. Hence, an alignment is a contribution to an increased perceived system benefit.
	Maintenance	
8	Documentation deficiencies increase the uncertainty with regard to system functionality. The system is perceived to be more complex, which causes higher **maintenance costs** (Kajko-Mattsson 2005).	In particular, alignment deficiencies are a reason for uncertainty and perceived complexity. Hence, maintenance costs increase.

Scott (2005) shows in a study that users considered support of their tasks much more important than presentation, navigation and learnability. Availability of the reference documentation, step-by-step guides to carrying out the task, illustrations of screens from the ERP software, enough explanation and complete and up-to-date information are significant aspects of task support. Scott (2005) therefore demands that documentations should focus on organization-specific business processes. This demand can be only met if the alignment of system and system documentation is given. Nevertheless, currently no empirical studies exist which explore the effects of misalignment and usability of system documentations in the different phases of the life cycle of an ERP software implementation. Only general studies about the effects of bad documentation quality and their impact on diverse success factors of ERP software projects exists (cf. left column in table 1). The general findings are transferred to the special problem of software and documentation alignment. To give a structured overview we ordered the effects of misalignment by the phases of ERP software implementation projects (cf. right column in table 2).

This statements can be consolidated into a framework. It can be seen as a first starting point for overcoming the research gap in the area of misalignment effects (cf. fig 1). The framework illustrates the consequences of deficiencies between functionality and documentation with regard to software purchasing, software training and support, software usage and software maintenance.

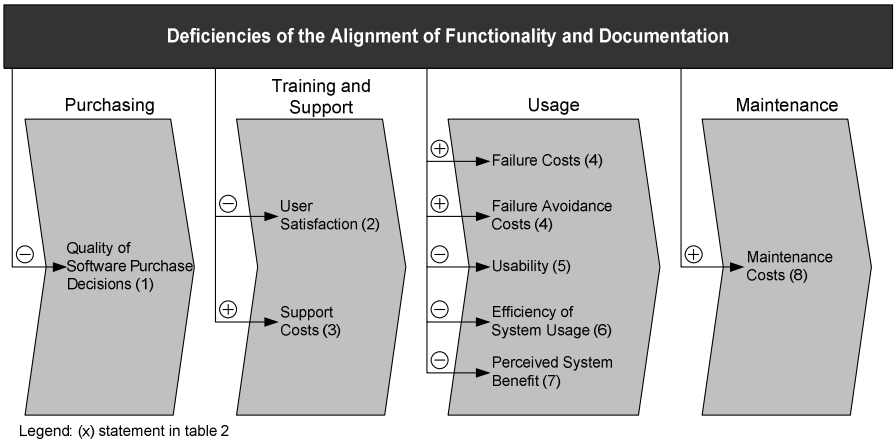

Figure 1. Consequences of alignment deficiencies between ERP variant and system documentation

Although the relevance concerning the alignment of system and system documentation has not been examined exhaustively, the problem has already been pursued in the past in the context of the development of tools to support software development:

- The literate programming approach by Knuth (2001) is based on the idea of providing only a single document encompassing both the implementation and documentation of a system in the context of programming.
- In contrast, the approach of the Desert System Reiss (1996) pursues the idea of having multiple different system documents. In order to secure their mutual consistency, a concept is suggested that is based on a variety of tools, which are integrated through broadcast messages.
- The Software Documentation Support (SODOS) System by Horowitz and Williamson (1986) administrates the documentation in a uniform document graph model and handles its elements and relations in a relational database.
- The Document Integration Facility (DIF) by Garg and Scacchi (1990) stores all textual information in files and administrates the relationships between documentation items in a relational database. It supports traceability through a keyword-based search and navigation mechanism.
- Nguyen and Munson (2003)'s special motivation for the development of the prototype Software Concordance (SC) was not only to ensure the consistency between documents of a single version of a software project, but also to represent its evolution. SC uses tree-based document representation of all software documents, including XML compatible source code, hyperlinks and embedded multimedia elements of documentation. Single parts of the documentation can be automatically analyzed (e.g. compilation of source code), without hindering interoperability.

The goal of the existing approaches is to ensure the alignment of system and system documentation, despite a continuous advancement of information systems, so that e.g. in the context of bug-fixing not only the source code is adapted, but also the documentation. Our motivation of generating and delivering customized variants for specific software customers has not been addressed by the discussed approaches so far. The distinctiveness of our approach lies in a parameter-based demarcation of system variants and the generation of system-variant-specific documentations in dependence of selected parameter values. Furthermore our approach supports a model-based navigation through the documentation, as well as the generation of a printable text file. For the advancement of our current concept we propose to adopt concepts from the other different approaches, in order to integrate the adaptable documentation more strongly with the source code.

Approach for the dynamic configuration of ERP system documentation

Method

Configurative reference modeling

For the development of a problem solution for the adaptation of documentation in sync with system customizing we transfer the concept of configurative reference modeling to this area. Under the name "configurative reference modeling", a variant management approach for information systems has been introduced. It helps customizing holistic models to model variants for specific needs via projections (e.g. Delfmann et al. (2006); Knackstedt and Klose (2005)). Configurative reference

modeling wants to reduce the general, holistic model to customized models for only a specific need. For example, a general model of retail companies should be reduced to an individual model for a specific retail company on a business or operational level and not on a technical level. Terms such as "retail business" and "warehousing business" (retailers accomplish all functions of procurement, storing and distributing) or "third-party delivery" (retailers do not accomplish logistic functions) help to eliminate unnecessary process and data elements. For example, with "single-level commissioning" and "Everyday Low Prices (EDLP) strategy" all process and data elements will be deleted that belong to aspects of wholesaling, multi-level commissioning or direct delivering. The (reduced) model that has been defined through the configuration process is a good basis for the further documentation of individual company characteristics.

Configurative reference modeling uses configuration parameters and configuration parameter values to formulate rules that help to reduce the holistic model to individual model variants. These are annotated to model elements. Configuration parameters with various configuration parameter values describe the application context. An example of a configuration parameter can be the retailer's type of business. Possible values are warehousing or third-party delivery. Configuration parameter values can be combined logically into configuration terms. These terms can in turn be used for true or false analyzes. For example, the configuration term "business type (warehousing) + trade level (wholesaling)" is true for all wholesalers with their own warehouses. For retailers with centralized clearing and third-party delivery the value will be false. In contrary, the configuration term "business type (warehousing) | trade level (wholesaling)" will deliver a true value for companies with warehousing or wholesaling, for example, retailers with warehouse or wholesalers with only a centralized clearing.

Projection rules are defined by connecting configuration terms and model elements. Per default, all model elements of the holistic model are part of a context specific model variant. Once a configuration term is annotated to a model element, the configuration parameter value has to be logically analyzed within the application context. If the analysis returns true, the model element will also be part of the variant. If the analysis returns false, the model element will have to be suppressed for the model variant. Most likely, model elements that are connected to this model element will have to be removed, as well. Appropriate rules have to be defined model type specific. To use the concept of configurative reference modeling for the alignment of documentation and functionality, various adaptations have to be taken.

Configuration parameters

Relevant configuration parameters for the alignment of documentation and functionality can be classified by two dimensions (see figure 2):

Stability: The dimension stability distinguishes configuration parameters by the frequency with that configuration parameter values change. Normally, within configurative reference modeling only configuration parameter values with high stability are used. For example, consumer retailers who own stores will not modify their business to centralized wholesaling within days or months – or maybe even years. On the contrary, operative processes with regard to specific measures are changing frequently. For example, validation rules for orders or travel expense claims change with the ordered or refunded amount. If the amount is very small the staffer will be able to handle the process on his own. If the amount is above a certain limit, a second employee will have to countercheck the transaction. With regard to the alignment of functionality (and therefore processes) and documentation, changes in functionality should also lead to changes in documentation. In this case, specific measure instantiation intervals are relevant configuration parameters. They can be used to define the interval for a specific process instantiation (and therefore documentation instantiation).

Individuality: Normally, configuration parameter values such as "retailer", "wholesaler", "warehousing" or „third-party delivery"refer to the entire company. Hence, configuration parameter values are relevant to all members of a company. This is also valid for the above mentioned order and travel expense claim processes. They are valid for all members of a company, especially for the employees in the accounting department. Therefore, the configuration parameter values are of low individuality. In contrary, configuration parameter values that apply to personal preferences such as color or GUI layout variations are not valid for all members of the company. They have a greater individuality. Therefore, individuality is the second important dimension for the characterization of relevant configuration parameters. Configuration parameters that describe the user behavior or user experience with the system are of high individuality. Within the alignment problem they can be used to adapt documentation elements to the recent interests of a user. For example, a detailed help documentation can be reduced if it is known that the user has understood the functionality. Such configuration parameters are of high individuality but low stability, because they are bound to the learning process of the individual user. Their values will change according to the behavior and experience of the user.

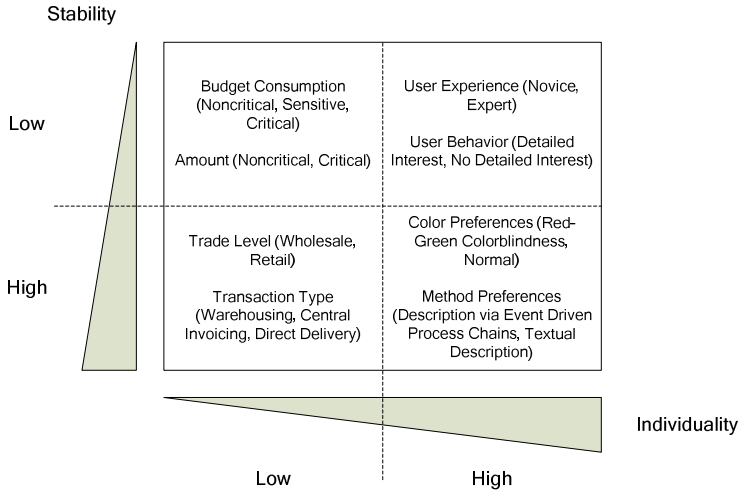

Figure 2. Classification of configuration parameters (examples)

Configuration parameter value settings for functionality and documentation alignment

In the area of configurative reference models it is proposed to query parameter values with regard to the model user. Subsequently, the parameter values lead to the adaptation of the model system. For the alignment problem a dynamic value setting is needed that can vary with regard to stability and individuality. Configuration parameter values with high stability and individuality can be defined once the system is installed. Parameter values with high individuality have to be defined by the user himself or automatically. It is reasonable to automatically identify value settings with low stability within the ERP-system. For example, budget consumption can be identified automatically. Furthermore, the system can react to decisions that are done within the process. If an employee decides to enter article master data in a more detailed way than normally, the specific value parameter can lead to extended, modified help documentation instead of the "normal" help documentation.

Configuration objects

Objects of the configuration can be all elements of ERP system documentation. Common documentation objects are very different in their structure, e.g. process models or textual user handbooks. The changes within the documentation have to be synchronized to the changes of the ERP functionality. Therefore, the configuration is not limited to documentation objects but can be extended to dynamically modifiable user interfaces. Configuration terms of different configuration objects need to be adjusted in order to achieve configured system and documentation versions that are consistent with each other. On the one hand, the complexity is reduced for system users because of dynamically adapted systems and documentations. On the other hand, the complexity for the creation of configurable systems and documentations increases. Nevertheless, the additional effort for documentation and system building is done once, while numerous customers profit from better usability and consistency. It can be expected that the additional effort for advanced system building will work out in the long run.

Procedure model

From a process based view, the approach changes the way of maintaining and provisioning documentation for ERP-systems. It changes the way ERP-systems and their documentation have to be designed and used. A procedure model (figure 3) can be summarized as follows:

1) First of all, configuration parameters such as "business type" and parameter values such as "warehousing" or "direct delivery" have to be defined in order to customize the documentation later on.

2) The creation of model and handbook items is done within the holistic model system. In the past, such model systems had to be constructed. The new approach requires not only documentation elements of a static system but all elements of all possible dynamic variants.

3) Configuration terms are assigned to individual items of the model system. The defined rules are the basis for generating document variants. The assigned configuration terms are the rule base of the dynamic configurable system and its documen-

tation. The analysis of the rules with regard to current parameter value settings allows the configuration of the system and its documentation.

4) The setting of the configuration parameter values is a new task for the use of dynamic documentation configuration. The setting can be done in various ways. Stability and individuality of configuration parameters have to be considered. Due to the dynamic of our approach this task has to be repeated depending on the stability of the configuration parameters.

5) For the generation of a documentation variant, all parameter values, which are relevant for the variant, will be checked. Configuration rules will be analyzed and the holistic documentation will be adapted to the specific variant needs. This task has to be repeated every time a configuration parameter value changes.

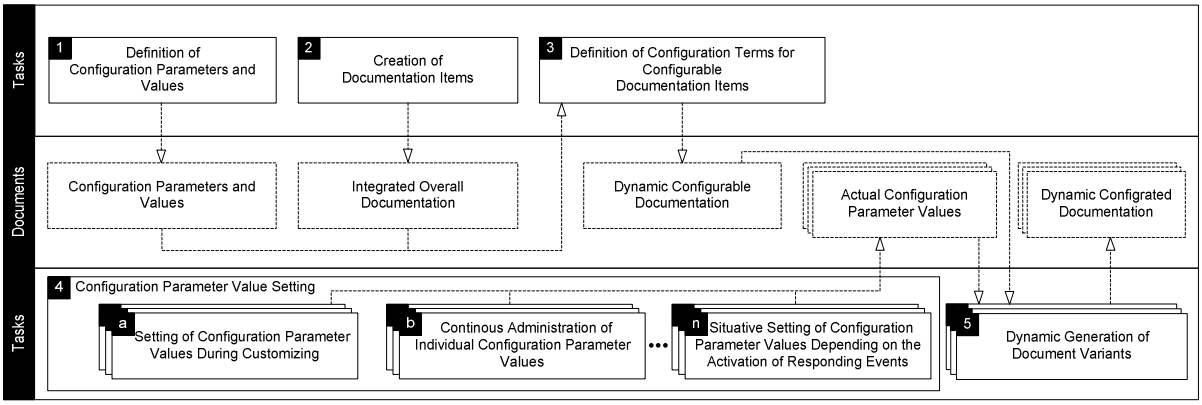

Figure 3. Procedure model

Example

An example of the applicability of such an approach for the management of variants in retail ERP-systems is given in figure 4. Configuration objects are an ERP GUI, an event-driven process chain (EPC) documentation and a user documentation. The user documentation consists of individual documentation elements that are structured hierarchically. Within the example, three configuration parameters are used that have different individuality and stability. The article structure will be defined during customization. If the company uses any bill of material (BOM) structures, the specific bill of material has to be defined before individual article data can be entered.

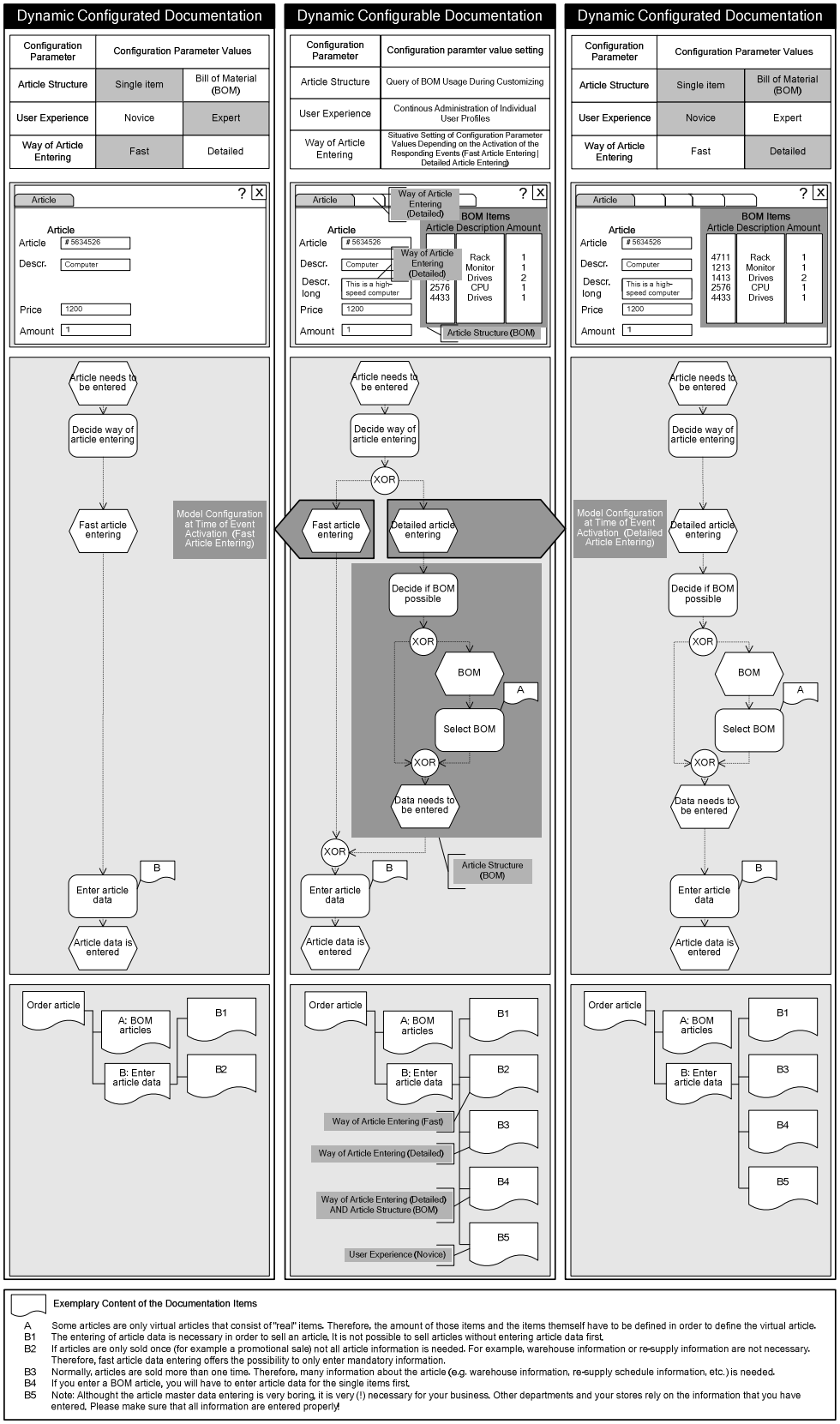

Figure 4. ERP example of a rule based configuration of a process model and a users' guide

If a BOM is not needed, these functionalities will not be necessary. Corresponding to the functionality specific BOM GUI fields will not be displayed and handbook items that describe the functionality will be eliminated from the help documentation. The configuration parameter "article structure" is very stable, because in most cases the company will only decide during implementation if they need this functionality or not. An employee's decision as to he will enter very detailed article master data or not is less stable and might differ from time to time. If an article is only used once, the employee most likely will only enter all mandatory data but no additional data. The way of master data entering cannot be decided during customization but depends on individual needs. Once the employee decides whether he needs detailed or fast article entering, the process model can be adopted automatically without any specific configuration terms. The XOR conjunction of the succeeding process paths only allows one of the two paths. Nevertheless, specific adaptations have to be done on the GUI and text documentation. Therefore, configuration terms are necessary. The example clearly shows the dynamic of the proposed configuration because the configuration depends on a specific event. As a third example for configuration parameters the individual user experience is used. Specific explanations in the online handbook will only be displayed if the user is an inexperienced user.

Data structures that are underlying the example in figure 4 are formalized as an Entity Relationship Model (ERM) in figure 5. It mainly describes the identification of model elements that need to be configured. Objects with configuration terms are entity-types as well as relationship-types. The entity-types fields, process chain element, process chain, function, handbook and handbook item are defined in order to use these elements for configuration terms. Therefore, it is possible to eliminate these types for model variants.

A model element that is used at various places, e.g. a specific handbook item that is used in various handbook chapters, would be eliminated everywhere if the requirements match the conditions. Hence, it is reasonable to define configuration terms for relationship types such as "Resource Function Associations", "Handbook Item-Function-Relation", "Handbook Handbook Item-Association" in order to eliminate specific model elements selectively.

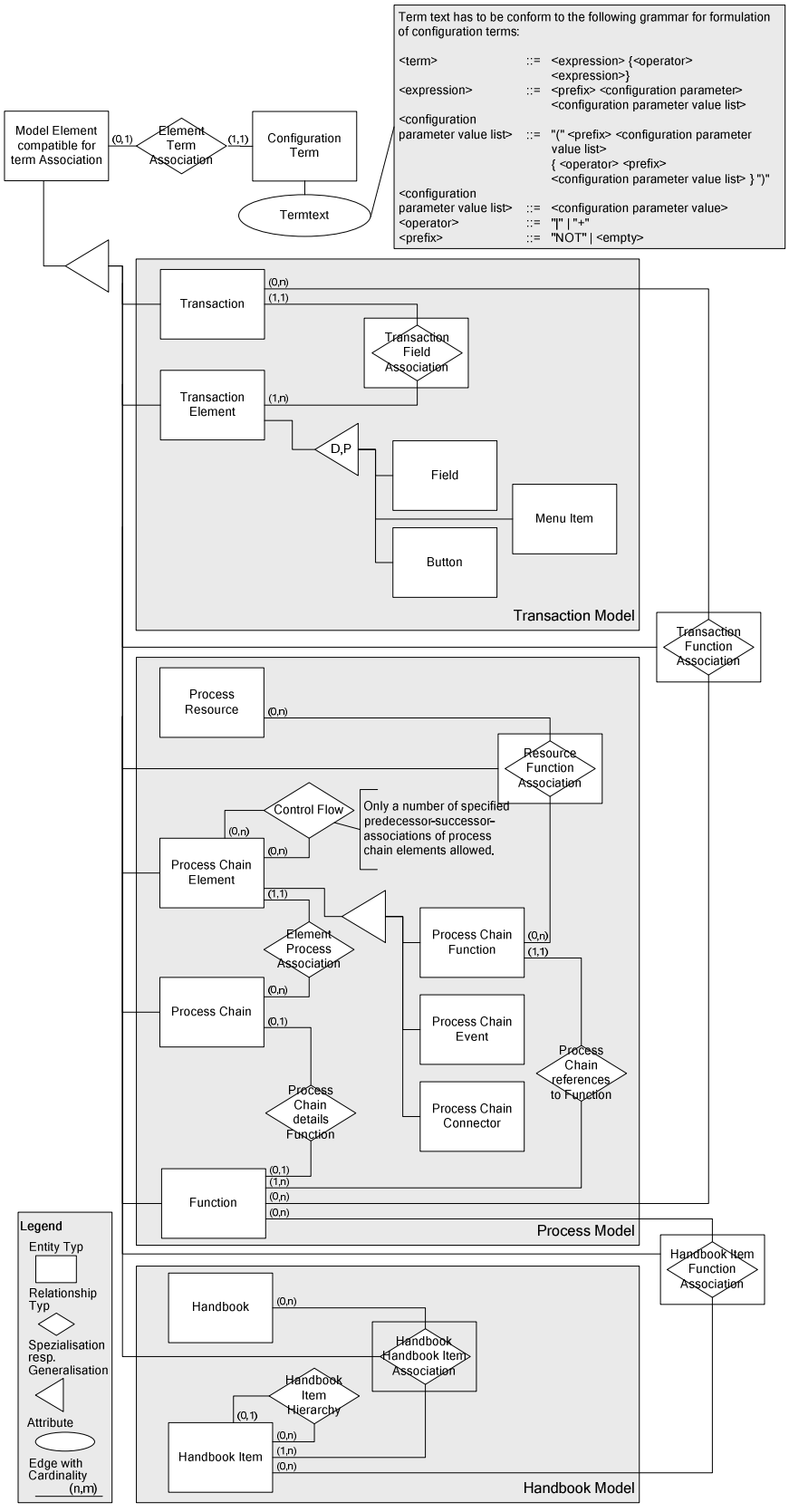

Figure 5. Data model for dynamic configuration of documentation

Tool support

For the evaluation of our approach, the H2-Toolset has been expanded in order to enable the creation, modeling and configuration of documentation items (Delfmann et al. (2006)). H2 is a meta-modeling tool that allows the construction of model languages within the tool. Models are represented in a hierarchical tree structure that can be expanded or shortened. This helps for the concise presentation of models.

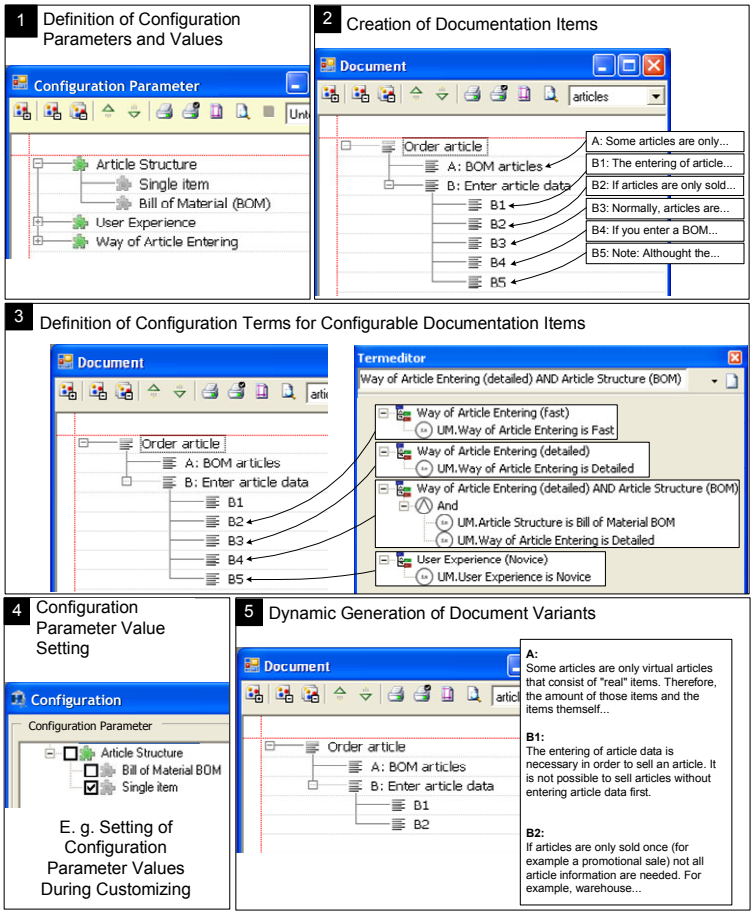

Figure 6. Tool support for the configuration process

Figure 6 gives an example of a handbook configuration, which is adapted to the above described example in figure 4. The whole documentation management can be done within the tool. Generated handbook variants can either be represented as a hierarchical model with its various handbook items or it can be represented as successive text. Both display formats only show the document variant that has been declared with regard to the functional ERP variant.

Until now, we accomplished expert interviews with two senior employees of important German ERP systems in order to evaluate our approach. We presented our solution in a face to face communication and asked for their opinion on the procedure model and concept. Both mentioned redundancies as one of their biggest problems of documentation. The advantage of our approach was fully understood and acknowledged. In further research, we will expand the evaluation of our approach by case studies and additional expert interviews.

Concluding Remarks

Within this article, an approach for the alignment of functionality and documentation adaptation has been developed. The approach addresses the alignment problem between customized functionality and provided documentation. For many ERP manufacturers documentation inconsistencies are a major concern. The approach helps to overcome these problems.

The article is focused on the development of a problem solving artifact and the research approach is a Design Science approach (Hevner et al. (2004)). Therefore, we finally classify our research in Hevner et al.'s Design Science guidelines in table 2.

Table 2: Orientation of the research project with regard to Hevner et al. (2004)'s Design Science guidelines

Guideline	Orientation of Research Project
1. Design as an artefact	The artefact of this article is a concept for software documentation management. It ensures the consistency of functional customization and its documentation.
2. Problem relevance	The relevance of the concept is proven by literature reviews. In further work it should be expanded to questionnaires with producers and users of software systems.
3. Design evaluation	The constructed approach is evaluated via a software prototype that enables the management of consistent software documentation of customized implementation projects. The helpfulness and relevance of such a prototype itself should be evaluated by further case studies and expert interviews.
4. Research contribution	Our research contributes to the discussion on software quality of ERP-systems. The concept that has been developed transfers the concept of configurative reference modeling to a synchronization approach for the management of system and documentation variants. It delivers a valuable contribution to the diffusion and utilization of these approaches. Our research is focused on ERP-systems. The analysis of the portability of our approach to other segments and software applications is an obvious and sensible enhancement of this article.
5. Research rigor	The progressive concepts of configurative reference modeling were transferred to our research problem. A framework that is based on literature review (cf. fig. 2) shows the theoretical evidence of our approach.
6. Design as a search process	Our research is a search process in two ways. First, the problem solution evaluation based on the tool prototype leads to a minor change of the concept. Second, the approach is limited to ERP-systems . Therefore, results need to be transferred to other software systems and markets.
7. Communication of research	A procedure model and a data model explicates the conceptional structures of the concept in order to formalize the approach. An example helps to illustrate the concept. A screenshot-based explanation of the prototype helps to understand the technical realization.

References

Agarwal, R., and Venkatesh, V., "Assessing a Frim´s Web Presence: A Heuristic Evaluation Procedure for Measurement of Usibility", Information Systems Research, 13 (2002) 2, S. 168-186.

Anson, P. H.: A comparative analysis of paper versus online documentation search strategies accessed by novice and expert adult computer users to achieve learning and performance goals. Doctoral dissertation, University of Idaho, Moscow, 2005.

Bernoider, E. and Koch, S., "Empirische Untersuchungen der Entscheidungsfindung bei der Auswahl betriebswirtschaftlicher Standardsoftware in österreichischen Unternehmen", Diskussionspapiere zum Tätigkeitsfeld Informationsverarbeitung und Informationswissenschaft, Nr. 20. Wien 1999.

Delfmann, P., Janiesch, C., Knackstedt, R., Rieke, T., and Seidel, S., "Towards Tool Support for Configurative Reference Modeling - Experiences from a Meta Modeling Teaching Case. In: Proceedings of the 2nd Workshop on Meta-Modelling and Ontologies (WoMM 2006)", Lecture Notes in Informatics. Karlsruhe, Germany. 2006, pp. 61-83.

Garg, P. K. and Scacchi, W., "A hypertext system to manage software life-cycle documents", IEEE Software, (7:3), May 1990, pp. 90-98.

Gillan, D. J. and Bias, R. G., ""Usability Science", Foundations", International Journal of Human-Computer Interaction, (13:4), pp. 351-372.

Guillemette, R. A., "Predicting readability of data processing written materials", SIGMIS Database, 18 (1987) 4, S. 40-47.

Hackos, J. T., Managing your documentation projects, 1994, New York.

Hevner, A. R., March, S. T., and Park, J. et al., "Design Science in Information Systems Research", MIS Quarterly (28:1), 2004, pp. 75–105.

Horowitz, E. and Williamson, R., "SODOS: a software documentation support environment – ists use", IEEE Transactions on Software Engineering, (SE-12:11), November 1986, pp. 1076-1087.

Ives, B., Olson, M. H., and Baroudi, J.J. "The measurement of user information system success", SIGMIS Database, 36 (2205) 2, S. 8-27.

Kajko-Mattsson, M.: A Survery of Documentation Practice within Corrective Maintenance. In: Empirical Software Engineering, 10 (2005) 1, S. 31-55.

Keirnan, T., Anschuetz, L., and Rosenbaum, S.: Combining usability research with documentation development for conference on Computer documentation. Toronto 2002, S. 84.80.

Knackstedt, R. and Klose, K., "Configurative Reference Model-Based Development of Data Warehouse Systems", Khosrow-Pour, M. (Ed.): "Managing Modern Organizations With Information Technology", Information Resources Management Association International Conference, San Diego, California, USA, May 15-18, 2005, pp. 32-39.

Knuth, D., Literate Programming, Volume 27 of Center for the Study of Language and Information – Lecture Notes, CSLI Publications, January 2001.

Mayhew, D. J., Principles and guidelines in software user interface design, Englewood Cliffs, NJ: Prentice-Hall, 1992.

Nguyen, T. N. and Munson, E. V., "The Software Concordance: A new Software Document Management Environment", SIGDOC'03, October 12-15, 2003, San Francisco, California, USA, pp. 198- 205.

Nielsen, J., Usability engineering, San Francisco: Morgan Kaufmann, 1993.

Nielsen, J., Enhancing the Explanatory Power of Usability Heuristics. CHI94. pp. 152-158, 1994.

Reiss, S. P., "Simplifying data integration: The design of the Desert software development environment", Proceedings of the 18th International Conference on Software Engineering, pp. 398-407, 1996.

Rautenberg, K.-U.; Sova, O., Dokumentation computergestüzter Informationssysteme: ein praxisorientierter Leitfaden für die Gesatltung, Erstellung und Wartung einer DV- Dokumentation. München 1983.

Retting, M., Nobody reads documentation. In: Communications of the ACM, (34:7), 1991, pp. 19-24.

Ruüietta, W.: Benutzerdokumentation für Softwareprodukte. Mannheim 1987.

Scheithe, P., "Vom "Hey Joe-Prinzip" zum professionellen IT-Service", Information Management & Consulting, 19 (2004), 1, S. 67-77.

Schriver, K. A., Dynamics in document design: Creating texts for readers. New York: John Wiley, 1997.

Scott, J. E., Post-Implementation Usability of ERP Training Manuals: The User's Perspective. Information Systems Management, Spring 2005, pp. 67-77.

Shaw, N., DeLone, W. H., and Niedermann, F.: Sources of dissatisfaction in end-user support: an empirical study. In: SIGMIS Database, 33 (2002) 2, S. 41-56.

Smart, K. L.; Whiting, M. E., and DeTienne, K. B., "Assessing the Need for Printed and Online Documentation: A Study of Customer Preference and Use", The Journal of Business Communication, (38:3), July 2001, pp. 285-314.