

Reference Models for Standard Software—Scientific Myth Instead of Practical Reality?



Reinhard Schütte

1 Motivation

Information models are an important object of research in information systems (see, among others Loos & Scheer, 1995; Becker, Rosemann, & Schütte, 1995; Rosemann & Schütte, 1999; Frank, 1999, p. 695; Davies, Green, Rosemann, & Gallo, 2004, p. 165; Wand & Weber, 2002; Becker & Schütte, 2004, p. 74ff.; Fettke & Loos, 2004, p. 550; Persson & Stirna, 2002). In this article, because the discussion about the model definition in literature cannot be conducted here in the required brevity,¹ information models should be understood in short as a representation of something for something (Becker & Schütte, 2004, p. 65).

¹In 1997, the author introduced the construction-oriented model concept into information systems, cf. Schütte (1998), which has been discussed in many publications over the last 20 years, cf. among others Wolf (2001), vom Brocke (2003), Wysusek (2004), Thomas (2006). From the discussion, it has emerged that the boundary between “model” and “no model” is narrow, which in particular also speaks against any image postulate. The image-oriented definitions represent a demand on an artifact before something can be called a model. In this context, image-oriented definitions necessarily become entangled in the problems of verifiability or uncertainty, cf. also Zelewski (1995, pp. 24–25). For reasons of the general validity of the following explanations, no conceptual restriction with the associated reduced extensionality of the term model is undertaken. This also means that an information model is not as narrowly understood as Jörg Becker published in 2003 in the sense of a construction-oriented scientific theory position: “Against this background, an information model can be defined as a representation of a fact. Furthermore, an information model is always to be understood in the sense of current speech and will usually last as a brand and will not be transient. [...] It is demanded that each modeling step and the insertion of each modeling component can be determined by reasons. As a representation of a fact, an information model consists of a set of elementary model statements that are linked to complex model statements. The truth of these statements can be verified by interpersonal verification” (Becker, Holten, Knackstedt, & Niehaves, 2003, p. 21).

R. Schütte (✉)
University of Duisburg-Essen, Essen, Germany
e-mail: reinhard.schuette@icb.uni-due.de

The purposes associated with information modelling have been the same since the Cologne Integration Model of Grochla (1974). These are to analyze and design information systems (Fettke, 2009). Information modelling is therefore not highly but outstandingly important for information systems (Fettke, 2009; Frank, 1999). This is also implicitly documented in the fact that the role of information models is emphasized in the standard literature on requirements engineering and software engineering, up to the efforts of Model Driven Architectures (cf. Pohl, 2010; Partsch, 2010; Wagner, Andres, & Lauer, 2007; Millet, 2013). The importance of information models in research and the significance for practice propagated by researchers was empirically investigated in the literature of Davies, Green, Rosemann and Gallo (2004), Davies, Green, Rosemann, Indulska and Gallo (2006), Sarshar, Loos and Weber (2006), Fettke (2009), among others.² The published empirical studies were usually carried out using questionnaires (in the more recent studies via the web and previously by sending the questionnaires via mail). In individual cases, also explorative approaches were used. The vast majority of the questions were asked as closed questions, and sometimes it was attempted to derive correlations using quantitative data. In the context of information modelling practice, this includes above all to the size of the company or the experience of the modelers and the intensity of use of information models. The works of Davies et al. (2006), Fettke (2009), Sarshar et al. (2006) document a considerably higher use of information models in larger companies than in smaller ones.³ This statement is important for the further argumentation, because the author does not want to expose himself to the accusation of generality and will conduct the further discussion exclusively for the class of enterprises, for which the use of information models was judged to be significant.

2 Reference Models and Their Use for Application Systems

2.1 Reference Application System Models

Reference models (as a short form of reference information models) represent a subclass of information models. They not only represent a specific situation—regardless of whether existing or ideal—but claim to be suitable for a class

²There are older empirical studies, cf. the list by Fettke (2009, p. 555) as well as the work by Batra and Marakas (1995), Maier (1998) which, due to their age, are no longer considered here because they no longer seem appropriate for the evaluation of current modelling practice. The empirical study by Schütte (1998, p. 367ff.) will be used in the further course in order to be able to carry out a longitudinal section analysis.

³It should be noted that this is not a continuous process (cf. Fettke, 2009, p. 573ff.), but all studies come to the conclusion that the intensity of use in the class of the largest companies in the study is significantly higher than in the class of small companies. This applies in particular to the reference application system models to be addressed later, e.g. cf. the works by Sarshar et al. (2006).

of applications⁴ (Delfmann, 2006; Fettke, 2006; Fettke & Loos, 2004; Knackstedt, 2006; Scheer, 1997; Schütte, 1998; Thomas, 2006; vom Brocke, 2003; vom Brocke & Fettke, 2016). This assumes reuse or reusability in different situations. This situation is present *prima facie* with standard application systems, since these are used in different enterprise contexts, which constitute a special type of reference models: the reference application system models. The main purpose of these is to document an existing standard system in order to enable a concrete technical system to be designed on this basis, as they are based on concrete application systems that have been designed not only for one case, but for several situations. Because of reuse in different situations, drawing on reference application system models is likely to be more economical than company-specific models. In any case, this presupposition is included in various versions in the literature.

With regard to reference models for standard application systems, the SAP reference model takes a special position, since it is regarded in the literature as the most developed and comprehensive reference model (on the history, see Fettke & Loos, 2004, p. 331; on the typology, see IWi 2018; on the evaluation as a particularly important application case of reference modelling, see Gottschalk, van der Aalst, & Jansen-Vullers, 2007; Leimeister, 2015).

The degree of usage concerning information modelling in business practice is not generally proven in empirical studies, even though the previously cited analyses show a usage.⁵ In the various studies, however, there was always a one-dimensional question as to whether information modelling would be used and for what purpose it would be used. Whether an *intensive* use of information models takes place remains open in the investigations. It seems trivial that in a larger company in a department, one would find at least one employee describing data with an ER model or a process using a formal language. The dominance of languages for data modelling found by Davies et al. (2006, p. 368), Fettke (2009, p. 560) in the form of ERM or the more data-oriented UML at least indicates that such artifacts are mandatory for the development and adoption of software systems. However, the actual significance of information models is not documented by the fact that something is used. One question would be to answer *in what ways* and *how intensively* the use is in order to be able to evaluate a degree of utilization and thus also the actual benefit of information models. The methodical deficits in the empirical studies are therefore to be evaluated as considerable and consequently hardly allow an evaluation of the use in practice.

This problem seems to be even more pronounced for reference application system models: the adoption of a standard system should always involve the use of information models. However, the question arises as to whether information models are used by the software manufacturer or whether own information models are created and which of the models were actually reference application system models. In the study by Sarshar et al. (2006) this does not become clear, so that the conclusions of the explorative study represent a starting point for the subsequent critical discussion.

⁴Therefore, the distinction between reference models in the field of objects and statements is artificial in Fettke and Loos (2004, p. 332).

⁵See the work outlined in the previous chapter on information model usage in general.

The study by Sarshar et al. (2006) assumes a positive correlation between the size of the company and the use of information modelling. Furthermore, the adoption of SAP is emphasized as a further influencing factor for more intensive use. The purposes of information model use are seen in “configuration of standard software” and “business process management”. The preference for these application purposes was also the result of the study on reference model usage at Schütte (1998), so that in the sense of a longitudinal view the hypothesis is supported that those are the two dominant objectives for reference model usage.

In his professional past, the author of this article has been responsible for or accompanying several SAP or other standard application systems adoptions in large companies and cannot share the conclusions from the literature about the use of information models for the adoption of standard systems. In some sense, there is no conflict between the findings of the empirical studies and the evaluation by the author, because information models are used in operational practice. The author has not encountered a project in which at least one data model with an ERM or a class diagram with UML or a process model with event-driven process chains, with BPMN or any process description had not been used. The use of tools, as found for example in Sarshar et al. (2006) also indicates that although there is a use, it can hardly be described as intensive. With drawing tools such as Visio, other Microsoft Office products, etc. models can be created, but serious information modelling in the sense of long-term use is not possible. In contrast to the rather positive empirical studies, which—in the absence of an analysis of the intensity of information model use in real projects—only arrive at superficial confirmations, the experiences of the author are sobering. The importance of information models for the adoption of standard application software has declined significantly since the start of SAP’s efforts, and sometimes the impression can be gained that information models no longer play a serious role in the adoption process.⁶ This observation in more than a decade of adoption practice at various large companies should serve to point out the problems of reference modelling by explicating various propositions.

The demarcation point of the argumentation, which the author will use for his discussion of the use of reference application system models, must first be concretized. The empirical studies on information modelling in general have shown that the intensity of use increases with the size of the company and the modelling experience of the users. Modelling experience is also likely to be more pronounced in larger companies in percentage terms, as the proportion of employees with an academic degree is usually higher. Thus, it makes sense to choose the use of information models in large companies with experienced modelers as a starting point. This situation is particularly likely to affect large technology companies that want to describe their standard application systems during or after development. In addition to company size and

⁶The year 1993 was chosen as the reference time, as SAP documented the system processes using EPCs and data models during the introduction of R/3 and promoted the introduction of reference models in marketing terms. There were separate departments within SAP for these tasks, which was driven forward not least by Gerhard Keller as a former employee of August-Wilhelm Scheer for the process models and also by Michael Seubert for data modeling. Also confer a similar evaluation by Fettke and Loos (2004, p. 331), who see 1993 as the first high point of reference modeling practice.

modelling experience, there is also the situation that the models are reusable several times in different contexts, which should take particular account of the economic efficiency of model creation and use (cf. also the first empirical study on reference modelling at Schütte (1998, p. 367ff.), Sarshar et al. (2006)). In summary, based on the empirical studies, it can be concluded that the use of information models in general and reference models in particular would have to be especially pronounced for the adoption of SAP systems in large companies with many modelling experts.

2.2 Propositions on the Limited Usability of Reference Application System Models

2.2.1 Model Use Requires the Availability of Models from the Software Manufacturer

The reference model of the standard software manufacturer forms the basis for the use of the reference application system models. In recent decades, the application landscape of manufacturers has changed dramatically. Not least due to the course of digitalization, the view of one large application system is increasingly obsolete. Individual comprehensive applications are being replaced by architectures with many applications in which the scope of heterogeneous technology components has also increased.

In contrast to the first climax of reference modelling, which was advanced by SAP and August-Wilhelm Scheer with IDS—in cooperation with SAP—the prerequisites for the usage have changed since then, because the development of standard software has become more distributed and different products exist in the respective ecosystems, which are developed “independently” and in parallel. The standard software vendors are also very product-centered due to acquisitions (for example, SAP Business Objects, Hybris, Success Factors, Concur, etc.; on Oracle’s company acquisitions cf. Oracle 2018) and cloud development and are to be understood as software ecosystems only. There are three main reasons for the poor availability of reference models, which will be briefly explained below: The complexity of the product variety of real application architectures, the platform problems and the changed development process of software.

The *enormous increase in complexity* of the solutions of the standard software providers has led to the fact that methodical guidelines for the development and methodical guidelines for the documentation of the software seem hardly to exist. In the course of this constant differentiation of the solutions offered, it is no longer possible to speak of a reference model anyway. A solution like ARIBA from SAP is in no way comparable to S/4 Finance, neither technologically nor in terms of documentation and implementation requirements. SAP currently provides the reference models via Solution Manager 7.2, whereby the individual reference models must be downloaded from the Internet, depending on the components to be implemented

. However, the availability of the models for individual applications is very unsatisfactory and only helpful in a few cases for the purpose of system implementation. In addition, Solution Manager provides tool support for creating own process models using BPMN in the phase of the business concept. Information models are used in many implementation projects. Last but not least, DSAG—the German-speaking SAP user group—recommends using the tools around SAP Solution Manager (DSAG, 2013) to create process models and implement Business Process Management. This also includes information on how the reference processes documented in the Solution Manager can be used. However, this is still not sufficient for the actual and subsequent adoption process of the software. The enormous change in the field of software solutions has contributed to the fact that software manufacturers—in this case SAP—have not been able to provide current reference models. The complexity of products and their combinatorics seem to make it impossible to keep reference models up-to-date. For reasons of market availability, the technical product is more important than the conceptual documentation. At the same time, this means that information modelling has not yet established itself in the creation of standard software. If an information model were created *ex ante* during the creation of standard software—in the sense of classical requirements engineering—the availability of the models would be guaranteed. The non-existence of information models after the development of a product proves that such a development practice is rarely found in software companies.

The non-existent reference models of standard software manufacturers are proof that reference models do not seem to have any added value for the manufacturers. If technology companies could be successful with even a minimum efficiency, then the question must also be asked why of all companies, those for which development is the very area of value creation, have not become pioneers of information model use (in contrast to the research expectations, see Becker, Delfmann, & Rieke, 2007). This circumstance should raise questions for the entire information modelling research as to whether the right problems are still being researched. Particularly due to the pronounced requirement pluralism in the implementation of requirements (since not only one situation but several are to be implemented) in standard software systems, it would seem obvious that information models are used due to the complexity of the task. The practice of software manufacturers not to resort to comprehensive information models before development can only be explained by the fact that it is either not economically advantageous or not yet economically necessary. The latter is the case if there is insufficient competition between the individual suppliers in the field of development which is important for the success of software companies. In this situation, the non-use of an important efficiency-enhancing instrument could be explained. However, in view of the fact that there has already been a significantly higher penetration of SAP in information modelling in the past, this does not appear to be a plausible reasoning.

The reasons for not applying information modelling at standard software vendors may have something to do with the division of labor in the development of software products. The development towards platforms such as Android, iOS, the HANA Cloud Platform, microservice architectures, etc. have led to software development becoming more distributed and above all “atomic”. These are largely uncoordinated

and there is an atomization of development, which may entail conceptual shortcomings (e.g. data integrity), but is accompanied economically by significantly faster availability of the solutions. The standard software manufacturers have recognized that they are less and less able to deliver additional functionality in a few releases, but instead have to rely on the power of the platform with the corresponding developers. The software market has developed into a platform market that can be described by Tirole's "two sided market" concept (Rochet & Tirole, 2003). The network effect now dominates the software world and this network effect was not that effective 25 years ago in software development, where software ecosystems were easier to control and were subject to other economic laws. It was possible to make guidelines and ensure coordination from a central location.

A third aspect, which can certainly be linked to the economic platform aspect, has something to do with the way of software development. Today, software is no longer developed on the basis of requirement documents alone as it has been in the past. Instead, the developers draw on frameworks within the programming environments to a considerable degree. This means that programming is no longer exclusively geared to the type of requirement, but must be integrated into the thought logic of the framework, as the developer interprets it. The components of the framework are thereby well known to the developers and are no longer "covered" by classical information models, so that the use of information models from a technical perspective is no longer mandatory. This technical view corresponds directly with the previously developed economic argumentation of the platform and the division of labour, because the technical change makes the outlined economic consequences possible. Taken together, the situation at the standard software manufacturers seems to speak against rather than in favor of an intensive reference model creation. It is also possible that the necessary framework conditions for information modelling are responsible for the existence of this situation. It would be the task of research to investigate this situation by means of explorative studies in order to develop solutions based on this in our construction-oriented information systems discipline.

2.2.2 Use of Reference Models During Adoption: The View of the Domain Companies

According to empirical studies, the domain companies that use information models in general and reference models in particular when adopting standard software strive to use models at least at the beginning of an implementation project. The main focus is on procedural considerations of BPM, as documented in the corresponding publications of DSAG (2013). The connection of the models with the configuration of software is rather implicit, because there is no model-driven customization of application software. As a rule, the reference models provided by standard application manufacturers have a very high level of abstraction in order to reflect the different application situations. The explicit presentation of variants that have also been the subject of considerations on language extensions in information systems (Schütte, 1998; Gottschalk et al., 2007; Rosemann & van der Aalst, 2007) has not yet gained

acceptance. For domain companies, the focus is on the two main purposes “improvement of business processes” and “configuration and development of software” (Fettke, 2009, p. 558; Davies et al., 2004, p. 39, 2006, p. 371).

The purpose of improving business processes with reference models is not to be expected directly in reference application system models. In the literature, no distinction was made in the empirical studies between reference application system models and reference organization models. However, the study by Sarshar et al. (2006), which focuses exclusively on the adoption of standard software, yields surprising results. Standard software will only become operational through the process of de-standardization (Mormann, 2016). It is not the standard that is used, but the individualization of the software is imperative from the processes, the structures and the individual data elements in the system. The work in sociology and business administration documents in their research results that the economic legitimacy of the institutions is hardly likely to lie in their interchangeability. In view of the brevity of this contribution, this fundamental knowledge cannot be elaborated. However, the optimization of processes is less the task of a reference application system model than the subject of an individual project for the adoption of software. This purpose-pluralism of reference models, from which the distinction between organization-oriented and application-oriented reference models also arises, has been problematized much too little, although this integration perspective is an essential subject area of information systems (on the multi-perspectivity of reference models, see Rosemann & Schütte 1999). The high degree of abstraction of reference application system models, which is problematized below, leads to the loss of individuality, even though the process-related design of which should have the advantage for companies. This aspect will be even more far-reaching in times of digitalization, because the systems are penetrating evermore areas of application and disregarding the differences would lead to the interchangeability of companies.

With regard to the configuration and development of software using reference application system models, existing models from the standard software manufacturer are used for customization purposes and, if necessary, for customer-specific development. Two aspects need to be taken into account. First, there are so large deficits in the availability of reference models when adopting application architectures that continuous use via the subprojects of a comprehensive solution is often impossible (cf. Sect. 2.2.1). The adoption of ARIBA as a platform for tenders and networking with suppliers is impossible because of the lack of available reference models.⁷ The master data creation that is to take place with an SAP S/4 HANA in conjunction with an SAP MDG is not represented in this scenario.⁸ Secondly, it seems essential that the prefabricated product “standard software” becomes a final product for the companies only through a project. At the end of the adoption project,

⁷ARIBA is SAP’s procurement and network platform, which SAP acquired in 2012. There processes of strategic purchasing, procurement, service procurement, etc. are supported.

⁸SAP MDG is SAP’s master data government product, S/4 HANA stands for the core products of the new SAP generation, which is documented in the 4 for fourth generation. The S stands for Simple and replaces the old R for Realtime (which extends to R/3).

a first product version is used. From the point of view of the adopting company, the importance of the project is often more important than the selected prefabricated product. For information modelling, this means that considerable modelling work still has to be done and that the degree of abstraction of the manufacturer's models is usually too high. Thus, the configuration, the company-specific refinement of the manufacturer's models, and also the enrichment required for process design purposes must take place. Due to the need for versioning of the models, resulting from the software manufacturer's constantly developed releases, a continuous and very complex modelling has to be carried out by the domain company itself. Very often companies succeed in modelling to some extent during the adoption of standard software and then, after the adoption, this task becomes obsolete. However, even in the initial phase there are considerable restrictions to be made, because it would be necessary for documenting the requirements, which is prepared by consultants and system analysts or specialist department employees, to know the models of the software manufacturer. If these are known, it should be noted that the models are usually so abstract that the consultant and developer knowledge is much more profound, so that they would first have to refine the software model. Alternatively, the models remain at the high level of abstraction and are used as guidelines. The documentation of customization and the developments required for individualizing the system is made in text form. In the author's experience, the latter procedure is the dominant one. The information models are therefore of little use for the adoption of standard software, which may explain the low usage intensity. They are mainly used as an argumentation as to why the methodology used for the adoption was correct, not because the models have sustainable benefits. In many discussions, the models make a visualization contribution that supports the clarification of requirements up to a certain degree of abstraction. However, the use for the configuration or development of standard systems is not yet discernible to some extent.

3 Conclusion

The explanations have shown that there are good reasons for a low use of reference models, which already reside on the software manufacturers' side. The basis for the reference application system model research is above all one in which the software manufacturers create and consolidate the models, have a governance in the development, etc. The information systems discipline has hardly been able to provide any insight into the way standard software is developed using information models, which is a major deficit. There have been some research projects dealing with the extension of modelling languages—in cooperation with SAP (Rosemann & van der Aalst, 2007)—further research projects are not available. According to the author's assessment, the progress of knowledge over the last 20 years has been far too small; there is no adequate understanding of adoption projects with reference models.

In reality, the hopeful claims of researchers have not been proven, so their statements are more assertive than substantiating, which is scientifically disappointing. Information systems should be less concerned with myths and more with reality and consider how reference modelling research needs to be adapted to the practice-oriented nature of the discipline.

References

- Batra, D., & Marakas, G. M. (1995). Conceptual data modelling in theory and practice. *European Journal of Information Systems*, 4(3), 185–193.
- Becker, J., Delfmann, P., & Rieke, T. (2007). *Effiziente Softwareentwicklung mit Referenzmodellen*. (J. Becker, P. Delfmann, & T. Rieke, Eds.). Heidelberg: Physica.
- Becker, J., Holten, R., Knackstedt, R., & Niehaves, B. (2003). *Forschungsmethodische Positionierung in der Wirtschaftsinformatik: epistemologische, ontologische und linguistische Leitfragen* (Vol. 93). Münster.
- Becker, J., Rosemann, M., & Schütte, R. (1995). Grundsätze ordnungsgemäßer Modellierung. *Wirtschaftsinformatik*, 37(5), 435–445.
- Becker, J., & Schütte, R. (2004). *Handelsinformationssysteme: Domänenorientierte Einführung in die Wirtschaftsinformatik* (2nd ed.). Frankfurt am Main: Redline Wirtschaft.
- Davies, I., Green, P., Rosemann, M., & Gallo, S. (2004). Conceptual modelling—what and why in current practice. In P. Atzeni, W. Chu, H. Lu, S. Zhou, & T.-W. Ling (Eds.), *Proceedings of the 23rd International ER Conference on Conceptual Modeling* (pp. 30–42). Shanghai (CN).
- Davies, I., Green, P., Rosemann, M., Indulska, M., & Gallo, S. (2006). How do practitioners use conceptual modeling in practice? *Data & Knowledge Engineering*, 58(3), 358–380.
- Delfmann, P. (2006). *Adaptive Referenzmodellierung: Methodische Konzepte zur Konstruktion und Anwendung wieder verwendungsorientierter Informationsmodelle*. Berlin: Logos Verlag.
- DSAG. (2013). *DSAG Leitfaden—business process management—version 1.0*. Walldorf.
- Fettke, P. (2006). *Referenzmodellevaluation: Konzeption der strukturalistischen Referenzmodellierung und Entfaltung ontologischer Gütekriterien*. Berlin: Logos Verlag.
- Fettke, P. (2009). Ansätze der Informationsmodellierung und ihre betriebswirtschaftliche Bedeutung: Eine Untersuchung der Modellierungspraxis in Deutschland. *Schmalenbachs Zeitschrift Für Betriebswirtschaftliche Forschung*, 61(5), 550–580.
- Fettke, P., & Loos, P. (2004). Referenzmodellierungsforschung. *Wirtschaftsinformatik*, 46(5), 331–340.
- Frank, U. (1999). Conceptual modelling as the core of the information systems discipline—perspectives and epistemological challenges. In *Proceedings of the 5th Americas Conference on Information Systems* (pp. 695–697). Milwaukee, WI (USA).
- Gottschalk, F., van der Aalst, W. M. P., & Jansen-Vullers, M. H. (2007). Configurable process models—a foundational approach. In J. Becker & P. Delfmann (Eds.), *Reference modeling: Efficient information systems design through reuse of information models* (pp. 59–77). Heidelberg: Physica.
- Grochla, E. (1974). Modelle und betriebliche Informationssysteme. *Integrierte Gesamtmodelle der Datenverarbeitung: Entwicklung und Anwendung des Kölner Integrationsmodells (KIM)* (pp. 19–33). München, Wien: Hanser.
- IWi. (2018). Reference model catalogs. Retrieved October 2, 2018, from <http://rmk.iwi.uni-sb.de/catalog.php>.
- Knackstedt, R. (2006). *Referenzmodellierung einer Managementunterstützung mit quantitativen und qualitativen Daten. Methodische Konzepte zur Konstruktion und Anwendung*. Berlin: Logos Verlag.
- Leimeister, J. M. (2015). *Einführung in die Wirtschaftsinformatik*. Berlin, Heidelberg: Springer.

- Loos, P., & Scheer, A.-W. (1995). Vom Informationsmodell zum Anwendungssystem—Nutzenpotentiale für den effizienten Einsatz von Informationssystemen. In W. König (Ed.), *Wirtschaftsinformatik '95—Wettbewerbsfähigkeit, Innovation, Wirtschaftlichkeit* (pp. 185–201). Heidelberg: Physica.
- Maier, R. (1998). Nutzen und Qualität der Datenmodellierung—Ergebnisse einer empirischen Studie. *Wirtschaftsinformatik*, 40(2), 130–140.
- Millet, P. A. (2013). Toward a model-driven, alignment-oriented ERP methodology. *Computers in Industry*, 64(4), 402–411.
- Mormann, H. (2016). *Das Projekt SAP: Zur Organisationssoziologie betriebswirtschaftlicher Standardssoftware*. Bielefeld: Transcript.
- Oracle. (2018). Strategische Akquisen durch Oracle. Retrieved October 1, 2018, from <https://www.oracle.com/de/corporate/acquisitions/index.html>.
- Partsch, H. (2010). *Requirements-Engineering systematisch Modellbildung für softwaregestützte Systeme*. Heidelberg: Springer.
- Persson, A., & Stirna, J. (2002). An explorative study into the influence of business goals on the practical use of enterprise modelling methods and tools. In *New perspectives on information systems development* (pp. 275–287). Boston: Springer. https://doi.org/10.1007/978-1-4615-0595-2_22.
- Pohl, K. (2010). *Requirements engineering: Fundamentals, principles, and techniques*. Berlin, Heidelberg.: Springer.
- Rochet, J.-C., & Tirole, J. (2003). Platform competition in two-sided markets. *Journal of the European Economic Association*, 1(4), 990–1029.
- Rosemann, M., & Schütte, R. (1999). Multiperspektivische Referenzmodellierung. In J. Becker, M. Rosemann, & R. Schütte (Eds.), *Referenzmodellierung: Reference modeling for business systems analysis* (pp. 22–44). Heidelberg: Physica.
- Rosemann, M., & van der Aalst, W. M. P. (2007). A configurable reference modelling language. *Information Systems*, 32(1), 1–23.
- Sarshar, K., Loos, P., & Weber, M. (2006). Einsatz der Informationsmodellierung bei der Einführung betrieblicher Standardssoftware. *Wirtschaftsinformatik*, 48(2), 120–127.
- Scheer, A.-W. (1997). *Wirtschaftsinformatik: Referenzmodelle für industrielle Geschäftsprozesse* (7th ed.). Berlin: Springer.
- Schütte, R. (1998). *Grundsätze ordnungsgemäßer Referenzmodellierung: Konstruktion konfigurations- und anpassungsorientierter Modelle*. Gabler.
- Thomas, O. (2006). *Management von Referenzmodellen: Entwurf und Realisierung eines Informationssystems zur Entwicklung und Anwendung von Referenzmodellen*. Berlin: Logos Verlag.
- vom Brocke, J. (2003). *Referenzmodellierung: Gestaltung und Verteilung von Konstruktionsprozessen*. Berlin: Logos Verlag.
- vom Brocke, J., & Fettke, P. (2016). Referenzmodellierung. In *Enzyklopaedie der Wirtschaftsinformatik*. Retrieved from <http://www.enzyklopaedie-der-wirtschaftsinformatik.de/lexikon/datenwissen/Informationsmanagement/referenzmodellierung>.
- Wagner, J., Andres, T., & Lauer, Y. (2007). Vom Prozess zur Ausführung—Modellgestützte Entwicklung betriebswirtschaftlicher Software. In J. Becker, P. Delfmann, & T. Rieke (Eds.), *Effiziente Softwareentwicklung mit Referenzmodellen* (pp. 61–76). Heidelberg: Physica.
- Wand, Y., & Weber, R. (2002). Research commentary: Information systems and conceptual modelling—a research agenda. *Information Systems Research*, 13(4), 363–376.
- Wolf, S. (2001). *Wissenschaftstheoretische und fachmethodische Grundlagen der Konstruktion von generischen Referenzmodellen betrieblicher Systeme*. Aachen: Shaker.
- Wyssusek, B. (2004). *Methodologische Aspekte der Organisationsmodellierung in der Wirtschaftsinformatik—Ein soziopragmatisch-konstruktivistischer Ansatz*. TU Berlin.
- Zelewski, S. (1995). *Petrinetzbasierte Modellierung komplexer Produktionssysteme—Eine Untersuchung des Beitrags von Petrinetzen zur Prozesskoordination in komplexen Produktionssystemen, insbesondere Flexiblen Fertigungssystemen*. Leipzig.



Prof. Dr. Reinhard Schütte holds the Chair of Information Systems and Integrated Information Systems at the University of Duisburg-Essen. He earned his doctorate in business informatics at the Westfälische Wilhelms-Universität Münster and researched and taught at the universities of Münster, Essen, Koblenz and Friedrichshafen. In addition to his scientific career, he has gained extensive experience in the introduction of large systems and system landscapes in retail companies. He is the author of various books and scientific articles published in journals. His areas of work include information modeling, enterprise systems, enterprise architecture management, digitization of institutions, and scientific theory.