Requirements Engineering for Enterprise Systems: What We Know and What We Don't Know?

Maya Daneva and Roel Wieringa

Abstract This chapter presents research progress in Requirements Engineering (RE) for enterprise systems (ES) with a view to formulating current challenges and a promising research agenda for the future. In the field of ES, many RE approaches have been launched and tried out in the past decade, however most of them are over-expensive and their effectiveness is unpredictable. Our goal in this chapter is to make an inventory of the approaches discussed in literature, to evaluate the quality of evidence available regarding whether these approaches actually worked, and to identify promising directions for future RE research efforts. Our results indicate (i) that while there are significant achievements, the primary goal of RE for ES is only partly achieved and (ii) that the field is likely to remain very challenging due to the increasingly more pronounced cross-organizational aspects of RE in ES projects (e.g. cross-organizational coordination, trust). At the same time, the need for practical, efficient and effective RE approaches will grow as the importance of ES in today's extended enterprises is growing.

1 Introduction

For at least a decade, the elicitation, documentation and negotiation of the requirements for systems based on commercial off-the-shelf (COTS) components have been regarded as an important sub-area of Requirements Engineering (RE). An important example of a project dealing with COTS-based system is the implementation of an enterprise solution based on packaged software, or the so-called Enterprise Systems (ES).[1] Typically, ES are large and multi-component systems that

M. Daneva (✉)
University of Twente, Drienerlolaan 5, P.O. Box 217, 7500, EA Enschede, The Netherlands
e-mail: m.daneva@utwente.nl

[1]We prefer to use this more general term over the more traditional "enterprise resource planning (ERP)" because today's ES have an architecture and functionality of a greater variety than traditional ERP systems.

S. Nurcan et al. (eds.), *Intentional Perspectives on Information Systems Engineering*,
DOI 10.1007/978-3-642-12544-7_7,

provide cross-functional services to a business. They often impact data semantics and business processes across more than one functional area of a business and today, they increasingly perform cross-organizational services. This sub-area of RE is becoming even more important as ES bring the vital capabilities for modern companies to network with others in forming extended enterprises [59].

The requirements for ES concern the business processes and the data flows that the ES should support as well as the key information entities in the subject domain of the system. These requirements reflect the needs of organizational units in one or more companies for a system that helps solve coordination and collaboration problems related to processing, for example, a purchase order, a good receipt, a sales order, or managing inventory levels. RE for ES is about composition and reconciliation of conflicting demands [13]. The RE process usually starts with a general set of business process and data requirements, then helps explore standard ES-package functionality to see how closely it matches the ES adopting organization's process and data needs [13]. This typically happens in an iterative fashion and includes (1) in today's cross-organizational case, mapping each partner company's organizational structure into the ES-package's predefined organization units; (2) defining a scope for business process standardization using standard application modules; (3) creating business process and data architectures specific to the extended enterprise based on predefined reusable package-specific process and data models; (4) specifying data conversion, reporting, and interface requirements. Currently, vendors of business software packages and their consulting partners provide standard RE processes for ES projects. In addition, a number of creative solutions were proposed by researchers and practitioners to further reduce the cost of RE-for-ES by avoiding scope creep, involving the right stakeholders, allocating sufficient resources, adopting goal-directed project management practices, and enlisting the vendors' and consultants' support to those problems [13, 28, 41]. Despite these efforts, it is still very difficult to find a match between the flexibility often required by the business and the rigidity usually imposed by the ES-package modules [14, 15]. In this chapter, we set out to identify the need for future research that addresses this difficulty. Our goal is to make an inventory of the approaches discussed in the RE literature, to evaluate the quality of evidence available regarding whether these approaches actually worked, and to identify contemporary currents which shape the future focus of RE research efforts.

The scope of this chapter is restricted to elicitation and modeling/documentation techniques and the main unit of analysis is at the micro level, i.e. projects and organizations, rather a business sector or even a geographic zone (e.g. North America, Europe, Asia). Some good studies that compare RE practices at macro level are presented in [21, 30, 45]. This chapter will not address the matter of industrial take-up of RE practices except in as much as this relates to parts of the RE for ES. For a thorough example of analysis of RE practices, we refer interested readers to [21, 30, 45].

The chapter is organized as follows: We start with a description of the results of a literature review of published research and experience reports in both journals and research-oriented conferences. This is followed by an evaluation of research progress in (i) requirements elicitation, (ii) modeling, and (iii) the impact of a few notable current trends on these two major RE sub-areas. We, then, lay out a set of further research directions that we inferred from our reflection on good recent progress, from examining past failures and from our knowledge about new business developments in the ES marketplace.

2 Identifying Areas of RE Publication Activity

In the RE community, there is a consensus that the main problem in RE for ES is the misfit between business requirements of ES adopters and ES functionality [14, 15, 20, 23]. Both RE researchers and practitioners agree that there is a gap between the functionality required by an organization and functionality offered by the various packages in the ERP marketplace. In the past decade, the RE community came up with a significant number of ideas meant to solve a broad variety of RE issues related to this gap. One reason for this growth in proposals is that an increasingly large number of companies have adopted packaged solutions and many of the adopters started reflecting and reporting on their implementation experiences, including their RE practices. Case studies and experience reports about ES implementations are now being published by companies in virtually any industry sector. In addition, there is much greater awareness of the importance of good RE practices and their adoption.

To illustrate the increase of RE publication on ES, we did a quick search of literature sources in a few prominent bibliographic databases (IEEE Explore, ACM Digital Library, Springerlink), which yielded Fig. 1.

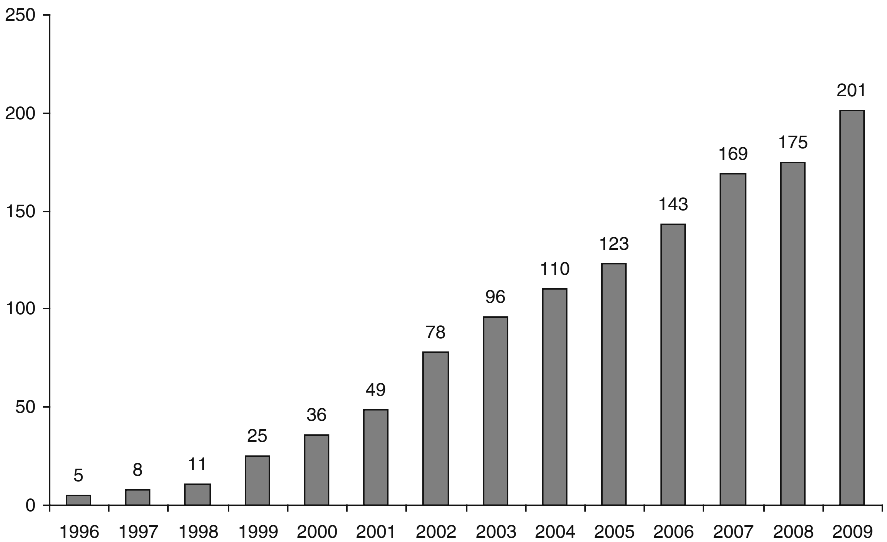

Fig. 1 Number of publications in three bibliographic databases in the last 15 years

Figure 1 indicates the number of papers, that have been published in (RE-related) journals and conferences and that include one of the strings "Requirements Engineering", "ERP", "Enterprise System", "Supply Chain Management System", "Business-to-Business Systems", "COTS", "Customer Relationship Management System" either in their title, or in their abstract. As Fig. 1 indicates, we found that the number of papers published between 1996 and 2010 grew up from 5 to 201. For the purpose of this chapter, we applied the following process of reviewing the content of these publications:

1. We merged the results of the search in the three databases and then eliminated from the resulting list those papers which were only remotely connected to the topics of eliciting and modeling requirements in ES projects.
2. The remaining papers were classified in two groups based on the two RE sub-areas which we deal with in this chapter (namely, elicitation and modeling).
3. We took notes on the key ideas included in the approaches from the papers that we classified in step (2) and on the practical application of the approaches.

We make the note that we did not go further to assess the actual evidence regarding the value of the proposed approaches, because our goals in this chapter are to take a snapshot view of the RE-for-ES field and to complement it with our current knowledge of market changes and, then, shortlist an initial agenda for future research.

Our merging of the search results in each database yielded a total of 110 papers, out of which we considered 53 for inclusion in this chapter. The specific aspects which we identified in these publications and which we selected for discussion in this chapter are listed as follows:

- requirements elicitation (Sect. 3.1), which is concerned with finding, communication and validation of facts and rules about the business,
- requirements modeling (Sect. 3.2), which is concerned with the business processes and data representation and analysis of the gap between the enterprise requirements and the package functionality, and
- type of empirical evaluation of the requirements elicitation and modeling approaches (Sect. 3.3), which is to understand (based on the RE publication authors' claims) those ideas that worked in real-life settings.

As the readers can expect, one can identify a number of overlapping aspects that pertain to elicitation and modeling of requirements in ES projects. However, we think that our classification of the RE approaches observable in the literature, is good enough for the goals of this chapter as we search for indicative observations. With this, we mean observations (i) that pertain to the state-of-the-art in RE research or practice and (ii) that suggest themes worthwhile investing our research efforts in the future.

3 Progress to Date

We checked the published approaches regarding their underlying ideas and assumptions, and the availability of empirical evidence about their effectiveness and the known problems about their use in practice.

3.1 Elicitation Techniques

Our review found that more than 20 requirements elicitation approaches have been proposed and tried out in real life project settings. In our observation, all these approaches are based on domain knowledge, however they differ regarding how they organize domain knowledge and how they create a domain dictionary (e.g. what knowledge sources they use for this). We clustered the elicitation methods in five groups:

1. *Process-mining based methods*, which employ a kind of process-mining technology in support of requirements elicitation activities. Examples of such approaches are presented in [11, 57]. These researchers came up with specialized tools to capture the complete business environments in which the ES will be put in operation. The result of using such a tool is then considered a first sketch of the ES process requirements. The idea of process-mining first surfaced in the 1990s, when Intellicorp, an SAP Development Partner, launched the LiveModel tool capable of identifying all transactions being in use in the current SAP environment in a company. The process models generated through this tool have been used by SAP implementation teams to draft the first version of process requirements in SAP upgrade, consolidation, or migration projects.
2. *Reference-model-based approaches*, which rely on predefined process and data models (termed reference models) and help clarify the questions of (1) what tasks must be performed and what package-embedded business functionality can support these tasks, (2) which organizational units should execute these tasks, (3) what information is needed for executing the tasks in a more efficient way, and (4) what information exchange must happen among tasks and how the package and other applications would support this. We refer to the reference models as to reusable and general descriptions of the commonalities in organizations, business sectors or systems that can be used as a base to derive other models from. In the ES RE field, we distinguish between two types of reference models: (i) software-independent reference models which represent generic descriptions of business processes and data flows in a certain enterprise area (e.g. accounting) or in a certain business sector (e.g. telecommunications), and (ii) ES reference models which are "conceptual descriptions of customizable software" (as defined by [43]). We must note that in RE-for-ES, there is a number of reference-model-based elicitation approaches that proved their market value in the past 20 years. Among the software-independent-reference-model-based

elicitation techniques, the ARIS framework [48] is one of the most popular. It provides RE professionals with a ready-to-use repository of industry-sector-specific business process models, meant to help structuring requirements elicitation interviews and making them more effective. The large consulting companies (e.g. Accenture, Cap Gemini, IBM, and others) have also developed proprietary reference-model-based elicitation approaches which rest on the very same idea as ARIS does. Another example of a business-area-specific elicitation approach is SCOR [31], which proposed an extended reference model of supply chain processes, including the structuring of information exchanged between business processes.

Furthermore, since 1992, the use of reference models has been encouraged by all major ES package vendors (SAP, Baan, Oracle and PeopleSoft). They documented the functionality of their respective packages in the form of ES reference models that also come for free to ES adopters as part of the ES itself. This made it possible, for RE staff and clients engaged in elicitation, to inform themselves quickly about the concerned ES functionality in business terms by navigating from the ES process and data models to the relevant piece of online documentation and to the smallest unit of software functionality, namely the transactions [12].

3. *Quality-model-based approaches*, which focus on the joint elicitation of functional and non-functional requirements by using standard underlying quality models or quality frameworks. The proposals in [6, 17] help building quality models for ES by deploying the ISO/IEC-9026 model, while [49] presents a Fuzzy Quality Function Deployment approach that helps translating functional requirements expressed with linguistic variables into non-functional requirements.
4. *Feature-based approaches*, which draw on the idea of top-down refinement of both functional and non-functional requirements. Similarly to the package-specific-reference-model-based approaches, the feature-based approaches help elicit domain knowledge through the investigation of the specification of the ERP package. These approaches term a function (or a quality attribute of the package) "a feature" [22]. Examples of feature-based approaches are the PORE method [28, 29, 33] and the PAORE approach [22]. When using these approaches, the elicitation analyst first shortlists suitable packages that match the ERP-adopter's requirements, and then elicits and documents in detail the requirements by presenting one concrete package's specification to the clients and by adding those client-specific requirements which are not included in the original package specification.
5. *Constructionist and organization-theory-based approaches*, which consider (i) ES requirements as a specific form of knowledge representation, (ii) the ES as an organizational transformation system, e.g. a system that changes its users' work patterns, and (iii) the ES project reality as socially constructed [2, 4, 24, 39]. These approaches rest on the position that our understanding of the ES requirements can be complete only when we understand the organizational transformation that the ES enables and the effect of the transformation on the

users. As Ramos and Berry explain, the transformation redefines the job of the elicitation analysts in that he/she must be aware of how and when particular pieces of knowledge are created in the RE process in order "to know when to be observing and what to be looking during the observation" [39]. The constructionist approaches generally propose to complement the elicitation techniques that focus on enterprise and system requirements with observation-based techniques (e.g. ethnographic methods) to elicit stakeholders' tacit knowledge and emotional requirements (e.g. values, beliefs). Emotional requirements are deemed [39] as important for the project as enterprise, functional and non-functional requirements are. Based on extensive case study research [39], Ramos and Berry convincingly justify why projects that include emotional requirements are more effective than projects that merely use the elicitation techniques mentioned earlier in the other four classes in this section. Examples of constructionist approaches are proposed in [2, 39]. In [39], the approach provides a list of symptoms and emotional responses which the elicitation analyst should watch for. In [2], the authors provide characterizations of five "roles that an ES can play" for its users. The job of the elicitation analyst is to first map the ES system in the concrete ERP-adopting organization against these roles, and then to structure his/her elicitation efforts based on the characteristics related to the particular role at hand.

It is important to note that most of the authors of the surveyed papers on elicitation techniques (discussed earlier in this section) carried out empirical evaluation research to demonstrate that their proposed techniques meet the goals that have been set for them in the first place. This commitment of RE researchers to the use of empirical research methods as well as the remarkable variety of elicitation techniques motivated us to search for publications that compare the techniques regarding, e.g. their effectiveness, the assumptions about the context that the techniques need to satisfy so that they are useful, or the business goals that can be best achieved by using these techniques. Our search yielded no publication that dealt explicitly with these questions. Instead, we found fragmentary evaluative information in those papers only, which discussed how vendor-provided ES-reference-model-based elicitation compares to process-mining-based elicitation. In all these cases, researchers used the comparison to stress the key limitation of vendor-provided approaches, namely that they are package-specific (and therefore they rarely could be used in projects that implement other packages). We think, therefore, that the search for insights on and improved understanding of when to use which technique, is the next big step in ES requirements elicitation research.

The elicitation techniques also seem to assume that the ES-based solution includes one vendor's product and is implemented in one company. No technique explicitly addresses today's case of cross-organizational ES implementations, in which the solution to be implemented includes more than one package, which may not all be provided by the same vendor and which may not all be used by all partner companies in an extended enterprise. The matter that the setting is cross-organizational poses new challenges, for example, how to elicit the requirements in the face of different partner companies using different ways to organize domain knowledge and to create their domain dictionary. Would it be possible at all for the partners in an extended enterprise to come up with one common way of approaching the requirements elicitation tasks? What kind of coordination models make sense to use so that partner companies coordinate their elicitation efforts? Future research in these areas is needed.

3.2 Modelling Techniques

In our literature review, we made a number of common observations referring to all the surveyed approaches to requirements modeling and documentation in ES projects. First, we found that all approaches are multi-perspective in nature (that is, they use multiple viewpoints to document the ES requirements). This is unsurprising, given the highly complex context where requirements are to be documented, which calls for using viewpoints as a tactics to cope with complexity.

Second, the RE publications agree on that in ES projects, requirements modeling addresses: (i) the selection of a package (and hence, the need to document the domain), and (ii) the alignment of a selected package to the ERP adopter's business (and hence, the need to model the functionality embedded in the package).

Third, the RE researchers give evidence confirming the viability of both top-down and bottom-up approaches to analyzing the gap between enterprise requirements models and system models. These two types of approaches differ regarding the starting point of the requirements documentation process. While bottom-up approaches imply to start from the review of the package specification and proceed with documentation of the domain requirements, the top-down approaches mean to starts from the (solution-independent) domain requirements that are to be further refined by using information about the concrete package functionality. The top-down approaches are preferred in contexts in which (i) modeling is a prerequisite for a package selection exercise, or (ii) it supports a business reengineering effort in an organization. In both cases, the outcomes of the modeling process are solution-independent requirements. The bottom-up approaches suit best those contexts, when the decision on a package has been made and modeling is a part of a business process/ES alignment effort. RE researchers [53] argue that unlike the bottom-up approaches, which target the alignment of a package, the top-down modeling approaches are capable of addressing both the selection of a package and the alignment of a selected package. In our view, regardless the focus of the modeling approaches, they both aid in jointly carrying out problem analysis and solution design activity (that is, joint RE and architecture design).

Fourth, our review indicates one common theme which is inherent to the research on requirements modeling, namely the exploration of the fitness relationship between domain models and system functionality. It is worth noting that those authors, who proposed requirements representation techniques, also investigated the fitness relationship. Their research efforts yielded quantitative models [16, 25, 46, 54] that help understand the fitness relationship and plan actions to preserve it when requirements change. The RE community is especially indebted to Colette Rolland and her team for the number of fitness analysis studies (e.g. [16, 25, 46]) which they carried out in this area.

Fifth, the RE community is united on that it is a good practice to represent both the domain models and the system models by using the same modeling language, because both types of models relate to business issues and in ES projects it is unnatural to segregate them. This position is shared by both researchers [42, 44, 53, 54] and practitioners [5, 12, 13, 36, 48]. Indeed, two of the market-leading packages, SAP and BAAN, provide modeling processes, tools and repositories of (solution-specific) models which describe the package functionality in business terms [12, 36]. SAP-adopting organizations may use the ARIS modeling languages [48], which were used to document the SAP application suite, while BAAN-adopters may use the Dynamic Enterprise Modelling approach [36] which is implied in the BAAN package. Presently, new variants of these modeling techniques have been proposed, e.g. the configurable reference-model approach [38, 44] to smooth even more the gap analysis process and the identification of the best possible configuration options within stated enterprise requirements.

However, the RE community also recognized that not all ERP packages have ready-to-use solution models and, in turn, spent significant efforts to solve the challenges related to this case. In the last decade, Colette Rolland [40] was the first (1999) to develop and evaluate a map representation that is to be applied in both domain requirements modeling and COTS systems modeling. Drawing on her experience, Rolland and Prakash [41, 42] redefined the map representation to cover the special case of ES as a major class of COTS-based projects. In 2000, Illa et al. [19] built the SHERPA method for documenting ES requirements and propose a formal language for modeling the application domain, translating user needs into requirements over the ES products, and for reflecting how concrete ES products adjust to these requirements. In 2002, the UML was customized to the ES project context [26]. In 2003, Arinze et al. [1] proposed an object-oriented framework to ease the gap analysis of enterprise and ES models, and Soffer et al. [53] developed and empirically evaluated the Object Process ERP representation that also is able to capture the so-called “ERP optionality” levels, that is, both the full scope of ES-embedded process and data control options, and the interdependencies among them. Building on it, Soffer et al. developed in 2005 a bottom-up reverse-engineering based modeling approach [54] to solve the problem of aligning a selected package to enterprise requirements. In 2004–2007, Carvallo et al. [8, 17] gave a new dimension to the discussion of requirements modeling approaches in ES by contributing to engineering the COTS (or ES) non-functional requirements. Based on case study research, these authors propose the RECSS method [8], a goal-oriented approach which helps describing enterprise requirements as well as functional and non-functional requirements of the system. By applying this method, requirements analysts can create a goal model of the system environment and also include those external elements that interact with the system. Complementarily to this, the RECSS method also uses a decomposition process through which one can build quality models for the system modules based on the ISO/IEC software quality standard.

Other RE researchers suggested the use of process modelling tiers to manage the complexity of enterprise and ES process modeling [18], the technique of the Requirement Integration Model [32] to account for interdependencies in business workflows, and the Data Activity Model for Configuration approach [34] meant to help align a package to the organization by the joint engineering of data and process requirements. The authors of [3, 52], also proposed ontology-based approaches to the representation and gap analysis of enterprise requirements and package-embedded functionality. For example, Babkin et al. [3] developed a requirements modeling approach that defines four sub-ontologies: ontology of requirements, of main data objects, of business processes and of configuration objects. The first ontology helps the elicitation process, while the other three ontologies are to support business process modeling activities and the activities of data requirements configuration requirements documentation, respectively. It is worth noting that the authors of these approaches [54] posed the question of how their proposals compare with the vendor-imposed modeling approaches (e.g. ARIS [12] and DEM [36]). They found that when using a modeling technique that is not part of the package, there are some extra costs involved in creating the models of the system functionality. Because the modeling approaches are not common standards in the ES field, for RE professionals to use them in a broader practical context, they first have to create a system model of their selected package.

We also make the note that all the proposed modeling techniques have been evaluated as a minimum by their authors by means of empirical research methods. Some techniques, e.g. the event-driven-process chain modeling method of ARIS [56, 58] have been validated by researchers that worked independently from the authors who originally proposed the technique.

Similarly to our survey in the sub-area of requirements elicitation, we also checked whether there are publications that compare the surveyed modeling techniques for their effectiveness. We found three studies [27, 37, 55] that compared business process modeling approaches. In [55] the authors compared them against a set of criteria which are reportedly critical to ES adopters. In [37], the authors compared two variants of the event-driven-process chain modeling technique (which is part of ARIS [12]) regarding perceived usefulness and easy of use from the perspective of modelers, by carrying out an experiment with postgraduate students. In [27], the authors compared business process modeling languages against a five-perspective-meta-model that helps judge the ability of a modeling language to capture the essential elements of the business context and the subject domain. While the authors in [55] indicate when to use which technique, the authors of the other two studies [27, 37] attempted to answer the question which of the compared techniques is better for a specific purpose/RE task.

While analyzing the existing modeling techniques, we also found that all modeling methods make tacit assumptions that might not be realistic in all situations. For example, RE authors seem to still assume that modeling is manifestly more useful than well written textual requirement documents. In our view, reality may question the extent to which this assumption is realistic. RE publications say very little about those contexts in which modeling would yield marginal benefits or be a financial burden and a project “over-kill”.

Furthermore, ES requirements modeling approaches have the tacit expectation that the resulting models are sufficiently understandable for those who are to review them and make decisions based on them. Our survey found that understandability of both enterprise and systems requirements models was addressed in very few papers and whenever it was addressed, it was from the perspective of the requirements engineer (also called requirements analyst). This finding agrees with a finding from a recent mapping study [10] that we did on empirical evaluation of the quality attributes of requirement specifications. Therein, we found that understandability was the most frequently researched quality aspect of requirements specifications, yet we found no paper that evaluated understandability of ES specifications. This finding is a surprise as it indicates a paradox: on one side, the authors of modeling techniques do acknowledge the importance of the social aspects in ES projects and the purpose of the models as communication vehicles to help establish a common understanding among stakeholders; on the other side, the RE research community published very little on the extent to which the models, produced by using the proposed modeling techniques, are understandable for the relevant project stakeholders.

Next, the papers which are focused on modeling for the purpose of gap analysis rest on the tacit assumption that the better the fit (that is, the closer the match achieved between business processes and ES solution), the more the value achieved. In reality however, the “aligned” ES solution becomes available for clients at earliest six months after the gap analysis took place and the value that clients receive at that time is far below the expectations. Indeed, the practice shows that only one out of five companies achieves more than half of anticipated benefits [59]. This alone questions the fundamental assumptions behind most gap analysis techniques and makes us think that we should re-evaluate our understanding of “business/ES fit” from the perspective of ES project goals and business value. Most of the ES projects have measurable goals and a gap analysis is rarely performed without considering the business case for the ES project. Therefore, it is worthwhile uncovering the relationship among the concepts of business/ES fit, project goals, and business case. This forms a direction for future research. We think that only when we understand sufficiently well this relationship, could we better leverage the RE community’s collective knowledge of business/ES fit and of gap analysis techniques so that it adds more value to RE practitioners and ES-adopters.

Next, most of the papers on modeling techniques do not address the costs involved in using these techniques. Those paper which do so, implicitly assume that the cost and effort needed are acceptable. This assumption might not be realistic in all ES contexts. For example, Soffer et al. [54] indicate that it would cost extra effort to run the OPM modeling process for aligning a package to enterprise requirements, as the analyst first have to create the model of the package functionality. In our view, it's also interesting to understand how much time (e.g. in person hours), it would take to create a model of a specific package component, e.g. accounting, in a company of a specific size. It is also worthwhile knowing how much time it would take to learn a modeling/gap analysis technique and its application process. Answers to these questions are important to make a decision on how to deploy the technique in a particular context. For example, the first author's personal experience suggests that a two-day training on the ARIS modeling technique was not enough for business owners to get comfortable in reading the SAP models without the help of external consultants. In that case, it turned out that hiring a specialist in ARIS-modeling on a permanent basis was much more cost-effective for the ERP-adopter than training all relevant stakeholders on how to use the ARIS methodology.

Last, the published ES modeling techniques tacitly assume that it's possible to scale them up to large projects. Today, this type of projects is, more often than not, cross-organizational, which increases the complexity of ES RE even more. If we are to apply a process-mining or reverse-engineering based approach to such a setting, this assumes that all partner companies in a extended enterprise are prepared to disclose their process and application landscapes (so that the respective tools capture completely their business environments). Assessing how realistic is to assume this means including the concept of trust in the discussion on ES requirements. This alone forms another line of research for the future.

3.3 Did These Techniques Work?

As indicated earlier, many papers on ES elicitation and modeling present empirical evaluations. In our review, we consistently observed that when authors propose a technique, they either provide a detailed account of its application in an industrial setting, or they run a complete action case study research intervention in a company and reflect on their learning from it. Table 1 presents the type of empirical research done in the papers which we cited in Sects. 3.1 and 3.2. In this table, the first column refers to the paper that published a RE-for-ES approach. The second column reports on whether this approach is for elicitation, or for modeling, or for both. The last column indicates the context where the empirical study in the paper has been done. The table shows that RE-for-ES researchers have been actively involved in action research with big companies. Some authors also include empirical research in the IT department of their institution (e.g. studies on ES implemented in a university). The brief indication of empirical studies in the table shows that researchers prefer action case studies for their evaluation. This increases the realism of the study but makes generalizability an important issue to consider.

Table 1 Empirical studies in RE for ES

References	Sub-area Elicitation	Modeling	Context of empirical study
[1]		+	SAP environment
[2]	+		Case study in a BAAN project at six ABB companies
[3]	+	+	Proof-of-concept in SAP CRM project
[8]	+		Proof-of-concept in COTS/Mail server system
[11]	+		Case study in SAP environment
[18]		+	Case study in SAP projects in the power generation sector
[19]		+	COTS projects in Spanish companies
[22]	+		Proof-of-concept in planning sales management project
[26]		+	Case study in SAP environment
[28]	+	+	COTS projects in various UK-based companies
[31]	+		Case study in SAP environment
[32]		+	SAP implementation project at a university in Thailand
[36]	+	+	Case study in Baan implementation projects
[37]		+	Case study in SAP environment
[39]	+		ERP case studies in Portuguese companies
[40]		+	Proof-of-concept in a COTS project
[41]		+	Case study in SAP environment
[44]		+	Case study in SAP environment
[48]	+	+	Case study in SAP implementation projects
[49]	+		A case study in a large ES project in 5 business domains
[53]	+	+	Case study in ES environment
[54]	+	+	Case study in ES environment
[57]	+		Case study in SAP environment

4 Directions for Future Research

4.1 Directions from our Analysis of RE Research

In this section we derive clusters of activities for future research, while reflecting on our findings in Sects. 3.1, 3.2 and 3.3.

Our review confirmed the presence of a multiplicity of RE approaches to ES projects. This is unsurprising, as no one approach is demonstrated to be superior to another. In addition, we observe that the variety of elicitation and modeling approaches brought a variety of empirical studies in which practitioners and researchers have used these approaches and shared their lessons learnt. We consider this use of empirical research methods beneficial to the RE community, especially when the studies are done independently by different researchers and not by the authors of the RE techniques themselves (e.g. [56, 58]), as this means a reduced bias. Moreover, the industrial studies refer to various domains in which ES were implemented and in a variety of business sectors. This is a positive development as well, because it opens up opportunities for cross-case analysis of the lessons learnt. Realizing these research opportunities is a worthwhile endeavor for the future.

Furthermore, today's ES packages no longer compete on business functionality but on quality attributes, that is on how well they meet the quality requirements (or non-functional requirements) of the ES adopters. Finding an ideal match between system configuration options and business processes would not be worth, unless it meets certain performance, availability, security, interoperability requirements (just to name a few). In the literature, we observe a number of publications [8, 9, 49] that acknowledge both the importance of quality requirements and the need to develop systematic approaches to address them in ES projects. However, how to trade-off these requirements, what represents the "right balance" among them, and when it is realistic to achieve the right balance (in intra-company and in cross-organizational settings) is largely unknown. Understanding the challenges this question poses and proposing approaches to counter these challenges represents a viable line for future research. Specifically, we mean understanding the contextual mechanisms that impact the process of joint RE and architecture design in ES projects.

The following two directions are closely connected and motivated by the increased use of ES as cross-organizational coordination support technology and the increased needs of ES adopters to design and redesign ES-supported coordination and collaboration processes within extended enterprises. The first direction refers to making the cross-organizational coordination requirements an explicit part of the requirements elicitation and modeling in ES projects. More in detail, our motivation of the importance of this topic for the future RE research is presented in [14]. In this review we found that with very few exceptions, the elicitation and modeling approaches subsume the coordination requirements into process and data requirements. An overall observation is that all the publications on techniques presented in Table 1 offer very little and fragmentary discussion on coordination, and when they add it, it refers to intra-organizational and not to cross-organizational coordination. We think that while in intra-organizational settings, this might not represent an acute RE problem, in cross-organizational context if we keep using the existing elicitation and modeling techniques as they are, it is likely to be sub-optimal because they are not geared to this context. We therefore think that these techniques should be extended (or even completely re-stated) to explicitly handle cross-organizational coordination requirements [14]. The second and related direction for future research is about getting actively involved in empirical evaluation of the existing techniques in cross-organizational contexts. Based on our recent research on cross-organizational ES, we identified seven characteristics of these projects which have implications for ES RE:

1. The projects deliver a shared system which lets the business activities of one company become an integral part of the business of its partners.
2. The projects create system capabilities far beyond the sum of the ES components' individual capabilities, which, allows the resulting system to qualitatively acquire new properties as result of its configuration.
3. The solution-to-be may well include diverse configurations, each of which matches the needs of a unique stakeholder group, which, in turn, implies the presence of coordination mechanisms unique to each configuration.

4. The projects deliver a system which is far from complete once the ES project is over, because a cross-organizational ES solution must mirror rapidly-changing business requirements, and so be adjusted regularly to accommodate current business needs.
5. The resulting solution does not have an identified owner at cross-organizational system level, as the system is shared.
6. These projects may well have a low level of organizational awareness of what RE activities (e.g. eliciting coordination requirements, identifying and analyzing coordination capability gaps, investigation and mapping of coordination mechanisms [14]) are to be used to elicit and model the requirements as completely as possible.
7. The solutions are not "built" in the sense that a master architect envisions the parts and their relationships; rather they evolve into existence and change through their life cycles as new shared pieces of functionality are built, existing intra-organizational systems connect to become shared, and shared parts of the system are disintegrated as soon as needs of sharing processes and data disappear.

We think that these characteristics pose elicitation and modeling challenges which are well beyond those presently addressed in the RE-for-ES literature. For example, these characteristics might make it overall difficult to use predefined business-sector-specific solution-independent reference models, as such models merely can not exist for all various collaborative arrangements that business partners may creatively come up with. In contrast, these characteristics may rather favor the use of constructionist elicitation methods in an extended enterprise settings as they explicitly account for the organizational transformation inherent to cross-organizational ES projects. The point we would like to make here is not that cross-organizational ES are different; it is that the assumptions which we usually make when we elicit and model the requirements in ES projects do not apply. We think that a RE analyst needs to know both (i) the elicitation and modeling vehicles at his disposal and (ii) whether or not the implicit and explicit assumptions about the use of these vehicles match the project settings. We saw in Sects. 3.1 and 3.2 that most of the assumptions we typically make in elicitation and modeling do not apply to a shared ES solution. Therefore, more research effort needs to be put into evaluating our existing techniques and understanding their strengths and possible weaknesses when deployed in a cross-organizational project context.

4.2 Directions from Examining Failures

One observation which we made consistently across the papers in our review is that almost all projects that the papers' authors described were reportedly kinds of successes. This clearly indicates the researchers' practice to learn from success; nevertheless we should not underestimate the benefits of learning from failed projects

[35]. Maybe, because of the prevailing culture to encourage researchers to publish more about the lessons they learn from success than about their learning from failures, in the RE literature we found no study that gives failed examples of using RE techniques in real projects. We must remind that in other disciplines, learning from failures has motivated innovation and we see no particular reason of why learning-from-failed-projects can not be useful for the RE community as well. In the experience of the first author, RE professionals do experience failure but the field does not profit from these failure experiences. The prevailing "we-can-fix-it-later-on" attitude, which is also compatible with the project management practice of compressing deadlines, brings many ES projects teams in a working mode that undermines the role of requirements. If a system fails at the go-life stage, then teams rarely get back to the RE process and look into RE malpractices, discern patterns of failure, and think of what they could do differently the next time. This situation is partly attributable to the prevailing business practice that consultants are contracted for six-month cycles and that, by the time RE mistakes are revealed in a project, they rush to their next project, which may be in another business sector and they may not see an immediate value of the reflection on what they could have done differently should they go through the same project again. Moreover, most of the consulting companies who employ the consultants are focused on securing their next contract engagement in another organization and, therefore, may have little time to spend on accumulating RE knowledge through systematic post-mortems of past projects. We support the position that to start learning from failures, we first need a few published examples of RE malpractice in ES projects. However, these examples are not readily available at the present time and it is a challenge to build up archives of bad examples and failures. By this, we do not just mean a set of poorly specified requirements, e.g. diagrams, or suboptimal gap analyses, but sufficiently documented explanations of why a RE method did not work as originally thought. We think that learning about the mechanisms that are at place and that possibly condition the success and failure of a RE practice will extend our repertoire of knowledge that can assist us in deciding which practice to use in which context. For example, it is well known that business owners in ES projects do not like reading technical descriptions (e.g. data models). However, there are RE teams in (assumingly) mature organizations who apply alternative (more creative) techniques for getting the business data (and conversion) requirements in a way that minimizes that chance of RE failure or even a project failure. What approaches do consultants deploy in getting the data requirements right? We think, these skills could and should be explicated and shared with others.

4.3 Directions from Existing Market Trends

In the last decade, there are a number of changes in the market demands that have implications for RE research for ES. For RE-for-ES to remain an industry-relevant research field, it must be able to keep up with these changes. This section lists some trends, which in our view restate and redefine the known RE-for-ES challenges, or pose entirely new challenges to RE for ES. We make the note that our list below may seem eclectic, reflecting our perception of particularly acute needs.

1. *The increased penetration of free and open source ES (FOS-ES) solutions.* Recent market research reports that ES adopters have become more receptive to FOS-ES [47]. A major reason for this trend is that FOS-ES means reducing licensing costs. In a recession-hit economy, FOS-ES solutions have become a feasible strategy for many small to midsize companies that want to automate their cooperation and coordination process. The technology of service-oriented architecture (SOA) made it possible for these cost-conscious ES-adopters to efficiently embed a FOS-ES-based solution within their processes and application landscapes, and also to customize or improve their systems on ongoing basis [51]. A recent review of the most popular FOS-ERP products is presented in [50].

   This trend introduces some changes that have RE implications [6, 7, 20]. For example, the distance between the user and the developer gets smaller, because the role of the ES adopter is changing from a consumer to a prosumer; this is an active role in which the adopter assumes the process of adapting software, reporting bugs, submitting feature requests, and posting messages to FOS-ES community lists. Based on their willingness to share information, smart prosumers will also provide bug fixes, new features and even entire modules. In this setting, it is expected [20] that the smaller distance between the user and the developer will alleviate the problem of misfit between the FOS-ES functionality to the enterprise requirements. This, however, has not been investigated yet by means of rigorous empirical research methods and we think it is a candidate line of research for the future.

   Moreover, becoming prosumers means to ES adopters a shift from a client viewpoint to a developers' viewpoint, which also means adopting a new mindset and accepting low level of managerial control, as the FOS-ES development is a community-centric activity. Hence, the adopter will have to follow a RE cycle that is influenced by many members of the community, which may incur massive coordination costs. How to create a cost-effective RE process for ES adopters and what coordination-enhancing activities should it include is an open question and warrants future research.
2. *The trend to form vendor-supported community collaborations for ES implementation.* In order to lower the ES implementation cost and shorten the ES project duration for their clients, ES vendors built online communities [47, 60] where ES-adopters can share their knowledge of aligning the respective vendor's package to enterprise requirements For example, two of the major ERP vendors, SAP and Oracle, have built, respectively, the SAP Developer Network and the Oracle Technology Network. The sharing platforms typically are Web 2.0 knowledge repository systems, which facilitate the members of the community to practice RE processes that actively involve case-based reasoning (e.g. exploring past cases, short-listing similar cases and reusing the solutions from the past cases to the particular context in question). Research [60] indicates that these repositories

streamline the collaborative execution of the knowledge-intensive activities in RE-for-ES within and beyond the ES-adopter's organizational boundaries, which can be invaluable in identifying the ways to improve the fit between enterprise requirements and ES functionality. We make the note that despite the collaborative nature of RE-for-ES, the forms of collaborations between the ES-adopters and consultants as well as among ERP-adopters themselves, has received in the RE literature only scant attention. Understanding the forms of collaborative RE-for-ES and the case-based reasoning models that serve best in the alignment of a package to enterprise requirements is a worthy line of research for the future.

3. *The trend to use agile RE approaches.* These have been gaining momentum among RE methodologies and are now entering the realm of ES implementation. More often than before, prominent agile publication venues (e.g. AGILE and XP), report on companies' experiences of introducing agile approaches to ES projects. (We searched the proceedings of these two conferences and found more than 10 papers on agile approaches in ES implementations at large companies). While one might think that the agile philosophy is incompatible with the ES project contexts, these companies experienced agile approaches as a viable option. We do not think that this is surprising, because the agile philosophy's focus on delivering business value and on satisfying clients is appealing to both ES-adopters and consultants who, especially in times of economic downturn, are pressed to demonstrate some specific instances of value of the ES-solution much earlier in the project. Second, at the heart of any agile approach is an assumption that regardless what the requirements might be at the project start, they will not be the same at the project end. It is intuitive to think, therefore, that the longer the project, the more realistic this assumption would be. In most situations, this assumption is realistic in the ES project contexts, notorious for their highly volatile requirements and prolonged duration. We think that the presence of agile approaches has certain implications for ES RE professionals and that it is a potential topic of future research to uncover what these implications are and how we can make a better use of the agile philosophy in RE for ES. In our view, the investigation of these implications is a mandate of the RE community and we should not leave this to the management science community or to the agile community and wait for them to come up with ideas for improving the existing RE-for-ES practices by using agile principles.
4. *The trend to deploy on-demand ES solutions.* The terms Software as a Service (SaaS) or on-demand ES refers to ES functionality being delivered over the Internet from a single application instance that is shared across all users. SaaS ES-solutions are rapidly increasing their share in some ES markets, notably CRM, and also penetrate into various business areas (e.g. financial accounting, human resource management). In uncertain economic conditions, particularly to cost-conscious small and mid-sized businesses, this type of ES solutions yields a number of cost benefits, including no up-front costs, no licensing fees and rapid, easy deployment. More and more companies are moving their mission-critical systems to the SaaS model to realize these benefits.

A SaaS-ES-solution is overall less flexible than on-premises ES in that the ES-adopter can not completely customize or rewrite its code. Because of this, the SaaS-ES-adopters must be prepared rather to change their business process to fit the solution than to align it to enterprise requirements. Also such adopters often face massive coordination effort because they have to integrate hosted software from various vendors with their existing ES solutions and/or legacy applications. How to run an effective RE process for SaaS-ES projects is by and large unknown. We, however, think that further research efforts in this direction are warranted, because SaaS-ES solutions represent an important development in the field.

5 Conclusion

This chapter surveyed the requirements elicitation and modeling approaches in the sub-area of RE for ES. We reasoned about some tacit assumptions these approaches make and why these assumptions might not be realistic in all ES contexts. Based on this we derived directions for future research. We acknowledge that such a survey can bring only a snapshot view on a fast-changing area. However, we think some lessons can be derived from it.

First, RE-for-ES has a long future ahead. ES will stay, though the on-premise ES solutions will have to live with new types of ES, namely FOS-ES and SaaS. The context of these projects gets increasingly more cross-organizational on both the ES adopters' side and the ES vendors' side. That the ES adopters are cross-organizational businesses calls for developing cost-effective approaches for handling requirements for business coordination. ES solutions include hosted and on-premise ES modules provided by multiple vendors, and this calls for cost-effective approaches to the complex problem of aligning the coordination mechanisms embedded in multiple packages to the coordination requirements of the ES-adopters. The elicitation and modeling approaches developed in the RE community in the past decade might only partly serve the needs of the ES projects embracing the current market trends.

Second, our analysis gives us enough evidence that ES implementations have impacted RE research regarding sub-areas as requirements elicitation and modeling. This means that RE researchers (active in non-ES project contexts) who design solutions to problems in those sub-areas should evaluate their proposed solutions regarding how they work in the ES context. In general, if a solution proposal is meant to be industry-relevant, then researchers have to evaluate and generalize its usefulness in various contexts. We think that ES is one significant context, for which such validation evaluations should take place.

Third, we witness that the majority of RE-for-ES techniques have been developed and evaluated by means of empirical research methods. This alone is an achievement, given the inherent difficulties in carrying out this type of research activity.

References

1. Arinze B, Anandarajan M (2003) A framework for using OO mapping methods to rapidly configure ERP systems. Commun ACM 46(2):61–65
2. Askenäs L, Westelius A (2000) Five roles of an information system: a social constructionist approach to analyzing the use of ERP systems. In: Proceedings of 21st international conference on information systems, Association of Information Systems, Brisbane, Australia, pp 426–434
3. Babkin E, Potapova E (2009) Using ontology for implementing enterprise resource planning systems. In: Proceedings of IEEE/ACS international conference on computer systems and applications, IEEE Computer Science, Los Alamitos, pp 67–68
4. Bergman M, King JL, Lyytinen K (2002) Large-scale requirements analysis revisited: the need for understanding the political ecology of requirements engineering. Reqs Eng 7(3):152–171
5. Brinkkemper S (1999) RE for ERP: requirements management for the development of packaged software Baan company. In: Proceedings of 4th international symposium on requirements engineering RE, IEEE CS, Los Alamitos, p 159
6. Carvalho RA (2006) Issues on evaluating free/open source ERP systems, research and practical issues of enterprise information systems, Springer, pp 667–676
7. Carvalho RA, Monnerat RM (2008) Development support tools for enterprise resource planning. IEEE IT Professional 10(5):39–45
8. Carvallo JP, Franch X, Quer C (2008) Requirements engineering for COTS-based software systems. In: Proceedings of the 2008 ACM symposium on applied computing, ACM, New York, pp 638–644
9. Colombo E, Francalanci C (2004) Selecting CRM packages based on architectural, functional, and cost requirements: empirical validation of a hierarchical ranking model. Reqs Eng 9(3):186–203
10. Condori-Fernánsdez N, Daneva M, Sikkel K, Wieringa R, Dieste O, Pastor O (2009) A systematic mapping study on empirical evaluation of software requirements specifications techniques. In: Proceedings of the 3rd symposium on empirical software engineering and measurement, IEEE Computer Science, Los Alamitos, pp 502–505
11. Chiplunkar C, Deshmukh SG, Chattopadhyay R (2003) Application of principles of event related open systems to business process reengineering. Computers Industrial Eng 45(3): 347–374
12. Curran C, Keller G (1998) SAP R/3 business blueprint: understanding the business. Prentice Hall, Upper Saddle River
13. Daneva M (2004) ERP requirements engineering: lessons learnt. IEEE Softw 21(2):26–33
14. Daneva M, Wieringa RJ (2006) A requirements engineering framework for cross-organizational ERP systems. Reqs Eng 11(3):194–204
15. Daneva M, Wieringa R (2008) Cost estimation for cross-organizational ERP projects: research perspectives. Softw Quality J 16(3):459–481
16. Etien A, Rolland C (2005) Measuring the fitness relationship. Reqs Eng 10(3):184–197
17. Franch X, Carvallo JP (2003) Using quality models in software package selection. IEEE Softw 20(1):34–41
18. Gulla JA, Brasethvik T (2000) On the challenges of business modeling in large scale reengineering projects. In: Proceedings of the 4th international conference on requirements engineering, IEEE Computer Science, Los Alamitos, pp 17–26
19. Illa X, Franch X, Pastor JA (2000) Formalising ERP selection criteria. In: Proceedings of the 10th international workshop on software specification and design, ACM, New York, pp 115–122
20. Johansson B, Carvalho RA (2009) Management of requirements in ERP development: a comparison between proprietary and open source ERP. In: Proceedings of the ACM symposium on applied computing (SAC), Enterprise information systems track, ACM, New York, pp 1605–1609

21. Juristo N, Moreno AM, Silva A (2002) Is the European industry moving toward solving requirements engineering problems? IEEE Softw 12:70–77
22. Kato J, Nagata M, Yamamoto S, Saeki M, Kaiya H, Horai H, Watahiki K (2003) PAORE: package oriented requirements elicitation. In: Proceedings of the 10th Asia-Pacific software engineering conference software engineering conference, IEEE Computer Society, Los Alamitos, pp 17–26
23. Kohl RJ (2005) Requirements engineering changes for COTS-intensive systems. IEEE Softw 22(4):63–64
24. Krumbolz M, Maiden NAM (2001) The implementing of ERP packages in different organizational and national cultures. Info Systems J 26(3):185–204
25. Le T, Rolland C (2001) Functional matching in COTS-based development context. Actes du XIXème Congrès INFORSID, Martigny, Suisse, pp 87–110
26. Linvald J, Østerbye K (2002) UML tailored to an ERP framework. In: Tolvanen J-H, Gra M. Rossi M (eds) Second workshop on domain specific visual languages. Companion of the 17th ACM SIGPLAN conference on object-oriented programming, systems, languages, and applications, New York
27. List B, Korherr (2006) An evaluation of conceptual business process modelling languages. In: Proceedings of the ACM symposium on applied computing, ACM, New York, pp 1532–1539
28. Maiden NAM, Ncube C (1998) Acquiring COTS software selection requirements. IEEE Softw 15(2):46–56
29. Maiden NAM, Ncube C, Moore A (1997) Lessons learned during requirements acquisition for COTS systems. Commun. ACM 40(12):21–25
30. Morris P, Masena M, Willikens M (1998) Requirements engineering and industrial uptake. Reqs Eng 3(2):79–83
31. Millet P-A, Schmitt P, Botta-Genoulaz V (2009) The SCOR model for the alignment of business processes and information systems. Enterprise Info Systems 3(4):393–407
32. Mutchalintungkul A, Oonhawat J, Pholpipatanaphong K, Sutivong D, Prompoon N (2006) Experience from applying RIM to educational ERP development. In: Proceedings of 28th international conference on software engineering, ACM, New York, pp 620–624
33. Ncube C, Maiden NAM (1999) Guidance for parallel requirements acquisition and COTS software selection. In: Proceedings of international conference on requirements engineering, IEEE Computer Science, Los Alamitos, pp 133–143
34. Negi T, Bansal V (2009) Integrating process and data models to aid configuration of ERP packages. In: Proceedings of 12th international conference on business information systems. LNBIP, vol 21. Springer, Heidelberg, pp 228–239
35. Petroski H (1992) To engineer is human: the role of failure in successful design. Vintage books, New York
36. Post HA, van Es RM (eds) (1996) Dynamic enterprise modelling: a paradigm shift in software implementation. Kluwer, Dordrecht
37. Recker J, Rosemann M, van der Aalst W (2005) On the user perception of configurable reference process models – initial insights. In: Campbell B, Underwood J, Bunker D (eds) Proceedings 16th Australasian conference on information systems, Sydney, Australia
38. Recker JC, Mendling J, van der Aalst WM, Rosemann M (2006) Model-driven enterprise systems configuration. In: Proceedings of Professional conference on advanced information systems engineering. LNCS, vol 4001. Springer, Heidelberg, pp 369–383
39. Ramos I, Berry D, Carvalho J (2005) Requirements engineering for organizational transformation. Info Softw Technol 47:479–495
40. Rolland C (1999) Requirements engineering for COTS based systems. Info Softw Technol 41(14):985–990
41. Rolland C, Prakash N (2000) Bridging the gap between organisational needs and ERP functionality. Reqs Eng 5(3):180–193

42. Rolland C, Prakash N (2001) Matching ERP system functionality to customer requirements. In: Proceedings of international symposium on requirements engineering, IEEE Computer Science, Los Alamitos, pp 66–75
43. Roseman M (2001) Requirements engineering for enterprise systems. In: Proceedings of 7th Americas conference on information systems, AIS, pp 1105–1110
44. Roseman M, van der Aalst W (2007) A configurable reference modelling language. Info Systems 32(1):1–23
45. Sadraei E, Aurum A, Beydoun G, Paech B (2007) A field study of the requirements engineering practice in Australian software industry. Reqs Eng 12:145–162
46. Salinesi C, Rolland C (2003) Fitting business models to system functionality exploring the fitness relationship. In: Proceedings of conference on advanced information systems engineering. LNCS, vol 2681. Springer, pp 647–664
47. Sang M, Lee MS, Olson DL, Lee S-H (2009) Open process and open-source enterprise systems. J Enterprise Info Systems 3(2):201–209
48. Scheer A-W (1996) Business process engineering: reference models for industrial enterprises. Springer, Berlin
49. Şen CG, Baraçl H (2010) Fuzzy quality function deployment based methodology for acquiring enterprise software selection requirements. Expert Systems Appl 37(4):3415–3426
50. Serrano N, Sarriegi JM (2006) Open source software ERPs: a new alternative for an old problem. IEEE Softw May/June:94–97
51. Smets-Solanes J-P, de Carvalho RA (2003) ERP5: a next-generation, open-source ERP architecture. IEEE IT Professional, July/August:38–44
52. Soffer P, Golany B, Dori D, Wand Y (2001) Modelling off-the-shelf information systems requirements: an ontological approach. Reqs Eng 41:183–199
53. Soffer P, Golany B, Dori D (2003) ERP modeling: a comprehensive approach. Info Systems 28(6):673–690
54. Soffer P, Golany B, Dori D (2005) Aligning an ERP system with enterprise requirements: an object-process based approach. Computers Industry 56(6):639–662
55. Toshiki A, Sommer R (2007) Comparison and evaluation of business process modelling and management tools. Int J Services Standards 3(2):249–261
56. Van de Aalst WMP (1999) Formalization and verification of event-driven process chains. Info Softw Technol 41(10):639–650
57. Van der Aalst WMP, Weijters AJMM (2004) Process mining: a research agenda. Computers Industry 53(3):231–244
58. Van Dongen BF, Jansen-Vullers MH (2005) Verification of SAP reference models. In: Proceedings of international conference on business process management. LNCS, vol 3649. Springer, pp 464–469
59. Ward J (2006) Benefits management. Wiley, Chichester
60. Wu H, Cao L (2009) Community collaboration for ERP implementation. IEEE Softw 26(6):48–55