Requirements Eng (2006) 11: 194–204
DOI 10.1007/s00766-006-0034-9

ORIGINAL ARTICLE

Maya Daneva · Roel J. Wieringa

A requirements engineering framework for cross-organizational ERP systems

Received: 10 October 2005 / Accepted: 22 March 2006 / Published online: 5 May 2006

Abstract The development of cross-organizational enterprise resource planning (ERP) solutions is becoming increasingly critical to the business strategy of many networked companies. The major function of cross-organizational ERP solutions is to coordinate work in two or more organizations. However, how to align ERP application components and business requirements for coordination and cooperation is hardly known. This paper reports on the outcomes of applying a coordination theory perspective to an analysis of the ERP misalignment problem. We present a conceptual framework for analyzing coordination and cooperation requirements in inter-organizational ERP projects. The framework makes explicit the undocumented built-in assumptions for coordination and cooperation that may have significant implications for the ERP adopters and incorporates a library of existing coordination mechanisms supported by modern ERP systems. We use it to develop a proposal for how to achieve a better alignment between ERP implementations and supported business coordination processes in inter-organizational settings. We report on some early assessments of the implications of our framework for practicing requirements engineers. Both our framework and library rest on a literature survey and the first author's experience with ERP implementation. In future empirical research, we will further validate and refine our framework.

Keywords Requirements engineering process · Business/IT misalignment · Enterprise resource planning · Cross-organizational coordination

M. Daneva (✉) · R. J. Wieringa
Department of Computer Science, University of Twente,
Drienerlolaan 5, P.O. Box 217,
7500 AE Enschede, The Netherlands
E-mail: m.daneva@utwente.nl
Tel.: +31-53-4892889
Fax: +31-53-4892927
E-mail: roelw@cs.utwente.nl

Abbreviations ERP: Enterprise resource planning · RE: Requirements engineering

1 Introduction

The changing nature of competition has opened up, paradoxically, the means for many companies to cooperate and build value webs in the networked marketplace [1]. As a recent Harvard Business Review report indicates, the typical corporation owes 15–20% of its total revenues, assets, or income to its participation in networks [2]. The orchestration of cross-organizational change requires unprecedented openness and collaboration between companies and modern enterprise resource planning (ERP) systems are becoming the fixture these companies are using to get the coordination support they need in order to stay interconnected with each other as well as with their customers. By 'network' we mean a set of different, independent, or nearly independent, businesses forming value webs—for example, the business network of WalMart Stores, Inc. which collaborates with a large number of non-U.S. companies and gives them direct access to the American market [1]. A typical large company has structured itself as a set of nearly independent business units, each responsible for its own profit and loss. For example, Colgate–Palmolive, as part of a turnaround effort, retooled its supply chain through a SAP solution to link all its business partners involved in purchasing, inventory management, manufacturing, and delivery. Despite the ambitious intention to use ERP to enable everyone involved in the value chain to make decisions based on the latest and best data available from everyone else, building connected and well-functioning business coordination process environments has proven to be arduous and much more difficult that anticipated by visionaries [3, 4]. For several reasons, implementing ERP in such a networked context is considerably more difficult than in an intra-organizational context. Firstly, direct decision making is pushed down

into an organization, as well as out into other organizations. Secondly, each company behaves in a way that maximizes its own interests, while assuming, often wrongly, that by doing so they are maximizing the common interests too. Thirdly, each business brings to the network different infrastructures, different enterprise systems, different business processes, different semantics of data, different authorization hierarchies, and different notions of collaboration. When such a business ventures out to cooperate for a particular purpose, all these differences still exist and none of the participating parties are prepared to change their infrastructure, business processes, and semantics, just for this particular cooperation, or to reveal the confidential business rules embedded in its processes and applications. Yet, to build a profitable ERP-supported network, each business actor must be able to decide which processes it will carry out itself and which ones it will perform for or with other actors [1]. These decisions should be made at the requirements engineering (RE) stage of any cross-organizational ERP project.

Enterprise resource planning implementation is the customization and introduction of an ERP system in a, possibly networked, business. RE is one of the most crucial project tasks in which the properties of the ERP system to be implemented and the requirements of the businesses that will use it are mutually aligned. Currently, ERP vendors and their consulting partners are providing standard RE processes for ERP projects. These are augmented by a number of creative solutions proposed to further reduce the cost of ERP RE 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 [5–13]. Despite these efforts, it is still next to impossible to find a match between the flexibility often required by the business and the rigidity usually imposed by the ERP system. This difficulty is recognized to be the central problem of ERP implementation.

The problem of flexibility versus rigidity is further aggravated in a cross-organizational context because, as we will see, the rigidity of an ERP system is imposed by assumptions embedded in the system about the business semantics, business processes, business communication channels, and business goals of the ERP adopter. If these hidden assumptions do not match the business, the business will experience the ERP as being rigid and not able to meet the business's requirements. In a networked context, there is a mismatch between the ERP and each of the participating businesses.

This paper proposes to tackle this problem from the point of view of the coordination theory [4, 14–17]. Since an ERP system is a coordination support system, we should be able to identify the coordination mechanisms supported by any ERP system. If we explicitly specify these mechanisms in a cross-organizational setting, then the requirements engineer should be able to find a match between the coordination support offered by the ERP system and the coordination mechanisms required by the cooperating companies.

The paper makes an inventory of the coordination mechanisms implicitly assumed by the ERP systems, and analyzes the role that the selection of these mechanisms plays in balancing rigidity imposed by an ERP system against the flexibility required by the cooperating organizations. We will see that rigidity allows the benefits of a cross-organizational cooperation to be reaped, while flexibility decreases the benefits and at the same time increases the cost of implementing and maintaining the ERP system.

The remainder of the paper is structured as follows. Section 2 reviews work related to our approach. Section 3 makes an inventory of cross-organizational coordination mechanisms currently supported by ERP packages. Section 4 presents a hypothesized model to discern the major issues that accrue from literature survey on the mismatch between business flexibility and ERP rigidity. It also assesses the impact of the coordination mechanisms on this problem bundle. Section 5 provides an early analysis of the implications of our framework for ERP RE and of the hypotheses that we think are worth further research efforts.

2 Sources and problem area overview

Our research builds on three types of sources:

- experiences in ERP RE published by the first author previously [7–9] and the experiences reported in [5, 6, 11, 13];
- frameworks for analyzing the alignment perspective in ERP projects [12, 18–31]; and
- the existing body of knowledge in the field of coordination [15–17, 20, 32].

The ERP literature does report on the issue of misalignments in ERP projects. However, the mere assertion that mismatches arise from unmet organizational requirements masks the variety of sources of misalignments. Prior research delving into the nature of the ERP mismatches is scarce. Two exceptions are the misalignment typology by Soh et al. [30, 31] and the framework for enterprise information architecture fit by Clemmons and Simon [20]. Soh et al. adopt a dialectic perspective and categorize mismatches [30] on the basis of four pairs of opposing forces: (1) push towards integration versus differentiation, (2) process orientation versus functional orientation, (3) flexibility versus restrictiveness, and (4) package domain specifics versus organization domain specifics. However, this typology was empirically derived from cases in one intra-organizational integration project in one public sector organization. Our context of interest is the exploration of these opposing forces in the area of cross-organizational ERP implementation.

Next, Clemmons and Simon [20] illustrated how to apply control and coordination concepts in choosing

the optimum enterprise information architecture in a multinational corporation. They create a firm strategic orientation grid which is then matched to an ideal ERP configuration. However, their solution is designed to respond to the specific problem of determining the point at which it is beneficial to implement variations in ERP-supported processes in various globally dispersed locations. The authors do not discuss anything about the mechanics behind arriving at a consensus on shared goals, process and information environments, and communication channels.

In our research, the theoretical standpoint that we found most helpful in examining the issue of misalignments and their sources is from the field of coordination [15]. Adopting this standpoint rests on the observation that modern ERP systems are, by and large, used as administrative frameworks for planning, conducting and monitoring a large array of functionally segmented operations in ways that both (1) accommodate, in real time, the intrinsic cross-organizational interdependencies underlying these operations, and (2) enable their control. From the coordination theory perspective, these systems can be viewed as coordination technology.

We follow Malone and Crowston [15] in defining coordination as the management of shared actions by different business actors. Classic coordination mechanisms distinguished in economic sociology include market-based coordination, in which goods and services are exchanges based on price, and relational coordination, in which actors work towards a common goal based on the shared and implicit norms of behavior [2, 16, 17]. IT mechanisms that support coordination include, among others, shared ERP systems and data warehouses [44].

Our effort addresses the need to answer two questions.

- 'Which ERP mechanisms are available for ERP adopters to support different coordination mechanisms?' and
- 'What coordination misalignments can be avoided if coordination requirements are identified and thoroughly specified earlier in the ERP implementation cycle?'

The relevance for cross-organizational RE is that an analysis of the desired coordination mechanisms of a set of organizations will lead to requirements for ERP package implementation. Usually, preferences about coordination mechanisms are implicitly anchored in the ERP packages and therefore lead to unpleasant surprises when they are not made explicit, and the ERP package chosen turns out not to match at all the implicitly desired but undocumented coordination mechanisms.

To the best of our knowledge, at the time of writing this paper, no systematic analysis of the role of coordination requirements in ERP projects had been carried out. There is neither a unified framework for describing the various kinds of coordination mechanisms nor a systematic set of rules for dealing with the coordination needs of organizations. Intuition, experience, and largely ad-hoc problem-solving processes are what requirements engineers and business representatives rely on most when confronting coordination misalignment issues. Moreover, the few ERP publications that include coordination aspects in the assessments of ERP systems [20, 31] describe these aspects only in general terms, without characterizing in detail differences between (1) how agreements on joint actions are achieved, and (2) how the default coordination mechanisms in ERP address those needs. This ambiguity makes it difficult (1) to determine what alternative coordination mechanisms might be useful in a given organizational context or (2) to directly translate these alternative coordination process designs into specifications of individual activities or uses of ERP to support a process, e.g., as part of a business process redesign effort [1, 22, 23, 34].

3 What coordination mechanisms are there?

We classify coordination mechanisms based on the scheme shown in Fig. 1.

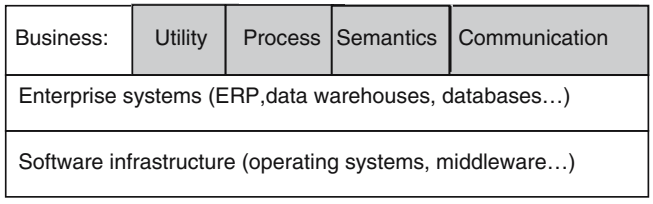

Fig. 1 The framework for business–IT alignment

The horizontal layers classify entities in a service provisioning hierarchy in the operational process of a business: software infrastructure provides services to enterprise systems which provide services to businesses. In the business layer, we take four views on businesses: a business provides services that have utility, it performs processes, it communicates with other businesses, and while doing that, the businesses exchange data with semantics. This framework is taken from our previous research on business–IT architecture alignment. For motivations, we refer the reader to those papers [35, 36].

Our interest is in the upper two layers of the framework, because these layers are where the process and systems alignment in networked organizations take place. A review of the literature on ERP implementation [19, 21–24, 27, 33, 37–43], and of our own experience in implementing ERP solutions based on the SAP package [7–9], reveals that there is a small number of coordination technologies in use at the enterprise systems layer:

- shared database,
- data warehouse,
- ERP functional application modules,
- workflow management systems,
- electronic marketplaces, and
- knowledge management systems.

There are additional technologies at the infrastructure layer, such as EAI middleware, mobile technologies for information sharing, programming languages mechanisms that make applications interoperate, and, still experimentally, web-service technologies. We surmise that there is little connection between the coordination processes on the business level and integration technologies at the infrastructure levels, and we will not pursue this here. Integration technologies at the enterprise system level, on the other hand, have built-in assumptions about coordination processes at the business level, and this is the topic of this paper. Note incidentally that on the enterprise system level, ERP is but one of the different possible integration technologies, among a variety of technologies such as shared databases, data warehouses, ERP modules, workflow management systems [44], and electronic marketplaces [37]. Although these technologies can be provided together by one ERP producer, a networked business may decide to use any combination of them and we consider each combination as a distinct technology. However, we hypothesize that at least some of our findings can be generalized to the other integration technologies at the enterprise system layer.

Our literature review has yielded the following coordination mechanisms at the business level:

- *utility-oriented mechanisms* referring to the partners' agreements on the goals and benefits of coordination;
- *process-oriented mechanisms* concerned with establishing end-to-end inter-organizational processes—for example, client order fulfillment processes or product-provisioning processes;
- *semantics-oriented mechanisms* concerned with the partners' agreements about the definition and the use of common meanings of key information entities; and
- *communication-oriented mechanisms* including the transmission and interpretation of information in the networked organization.

Tables 1, 2, 3, and 4 describe which coordination mechanisms our study covers and how these are supported in state-of-the-art ERP systems. The tables structure ERP support for coordination in a networked context and provide examples. The tables are meant to illustrate the different coordination mechanisms and are in no way exhaustive. Indeed, they cannot be exhaustive, as new approaches to cross-organizational coordination are being invented on an ongoing basis.

The crucial observation to make of these mechanisms is that each one starts with the word 'shared.' Now, inside one company it may be often true that these mechanisms are shared without anyone ever talking about them explicitly. After all, within one company, we can assume that there is one culture and one shared way of doing things. However, across different companies, what members of one business silently assume to be the regular way of working can be quite different from what members in another business silently assume about the regular way of working. Even within one company, there may be severe mismatches—for example, if the company consists of different nearly independent business units. Tables 1, 2, 3, and 4, therefore, represent a list of hidden assumptions that must be brought out in the open in any cross-organizational ERP implementation. The assumptions about shared coordination mechanisms may be quite different across different business partners.

Furthermore, if we compare the coordination support provided by today's ERP systems as indicated in Tables 1, 2, 3, and 4 to the general coordination mechanisms as studied in coordination science [15], it becomes clear that the latter are too general to be

Table 1 An inventory of the utility-oriented cross-organizational coordination mechanisms

Implicitly assumed coordination mechanisms	What does it mean to ERP adopters?	Examples
Shared vision of the overall benefits for the networked organization [22, 23]	Presenting one face to clients and sharing corporate identity	Unified brand management by using common order management, sales force, service, and marketing analytics applications
Shared view of services offered by the network to clients	Motivating dependencies between the services of different businesses	Implementing revenue- and risk-sharing models
Shared organizational control mechanisms [20, 59]	Ensuring that behaviors originating in partner companies are compatible and support common organizational goals	Creating monetary incentives and fairly distributing risks, costs, and rewards across the network to induce supply chain partners to behave in ways that are best for everybody [59]
Common framework for empowering staff	Keeping employees up to date and prepared to actively participate in decision making and problem solving	Building 'communities of practice' with partner companies' staff and involving them in problem solving [1]
Dual focus on local and overall performance	Connecting local partner operations to process mission by balancing local (partner) performance with the overall (network process) performance	Monitoring and analyzing the extended supply chain
Shared agreement on how the capabilities of the different partners are utilized in joint processes [1]	Assigning and accepting responsibility for each facet of the network business relationship	Altering the alignment of actions due to changes in business conditions or technology [59]

Table 2 An inventory of process-oriented cross-organizational coordination mechanisms

Implicitly assumed coordination mechanisms	What does it mean to ERP adopters?	Examples
Common agreement about business process environment [21–23]	Standardized operational procedures, access permissions, and control patterns	Common payment processing procedures
Agreement about process orientation [4, 21–23]	Reducing organizational operations to a large series of procedural steps tied together to sequences, sub-functional categories, modules, and cross-modular operations	Creating vendor master files or charts of accounts
Common agreement on management policies [22]	Sharing enforceable business rules that are explicit and consistent	Rules for tracking employee attendance and absence
Solution maps [39]	Descriptions of the most important business processes within an industry sector, the technologies (ERP elements, add-ons), and services needed to support the processes	Branch-specific solution maps, like mySAP aerospace and defense solution map [39]
Common secure or non-secure customer 'self-serve' practices	Opening up segments of the process logic that customers and suppliers can execute on their own	Interoperable supply chain processes in which suppliers of a manufacturing company are responsible for monitoring inventory levels on their own: when the materials they provide start to run low, they can recognize the status and replenish the stock without having to be notified by the manufacturing partner company [45]
Shared transaction processing engines [21]	Shared understanding of the position of ERP in the cross-organizational architecture	Trading partner portals, auction and exchange mechanisms, catalogs, as provided by Oracle Exchange [21] and SAP [39]
Common agreement on market-making mechanisms (matching and aggregation [37])	Sharing a base of potential suppliers and customers and using the potential for lower-cost purchasing through aggregation	Adopting a global e-business hub

Table 3 An inventory of semantics-oriented cross-organizational coordination mechanisms

Implicitly assumed coordination mechanisms	What does it mean to ERP adopters?	Examples
Shared data dictionary [7, 22, 33, 38]	Common definitions of information entities	Maintaining a centralized view of the network's customers and business partners data
Shared reporting formats and semantics [22]	Standard presentation formats and information content of the output	Integrating global data on site capacity, production and transportation costs, tariffs, and the demand to schedule across multiple sites [22]
Delegation about data access permission [42]	Distributed access to data and distributed application logic	Enabling open information sharing culture to balance top-down control with bottom-up empowerment
Shared access to information across the firewall [4]	Standard methods for producing a global, enterprise-wide view of business operations	Supply chain dashboard applications that consolidate supply chain information from manufacturing facilities at different locations, each one using ERP differently to track the inventory, purchases, manufacturing, and sales [4]
Common principles of cross-organizational data management [23, 39]	Data consistency [46] and alignment with businesses	Data ownership, data modularity, trust: no need of alternative sources to verify data accuracy [46]
Reference models [33, 38, 39, 42]	Representing practices embedded in the package in the form of reusable process and data models	The R/3 reference model [38, 41]
Shared product models [33]	Industry-specific solution aspects of the ERP package	Configurable master lists to allow product specification with variances, a requirement specific to the metal, paper, and textile industries

Table 4 An inventory of communication-oriented cross-organizational coordination mechanisms

Implicitly assumed coordination mechanisms	What does it mean to ERP adopters?	Examples
Agreements about communication channels and a common language	Shared understanding about the transmission and interpretation of complete and legally valid business documents	ERP-package compliant XML-based schemas for e-network transactions—e.g., RosettaNet for electronic manufacturing, semiconductor, and telecommunication industries [60]; ACORD for insurance, reinsurance, and brokerage [21]; and OASIS for the public sector
Sharing of knowledge	Bringing employees to the required level of understanding to get their job done	Role-specific portals allowing sales staff to call up customers' purchase history with the firm, external reports, and discussion items from other sales and service staff members who have dealt with the customers [21]
Shared understanding on the conformance on messaging services	Automating the sending and receiving of business documents between partners [60]	RossetaNET partner interface processes [60] that help structure each partner's action messages and define sequence in which messages are sent along with the quality of service attributes for message exchanges
Shared learning [1]	Using what each partner learns in the interaction with customers both to respond to immediate needs and to determine what future markets will require	Anticipating possible customer problems and getting appropriate responses ready and waiting

helpful in making choices in ERP implementations. However, we agree with these authors that a coordination mechanism can be characterized by the extent to which it is suited to different organizational tasks, corporate cultures, and environments. Thus, in a case of cross-organizational ERP projects, coordination mechanisms vary in (1) the degree to which coordination is prescribed at the time of RE, (2) the cost in terms of time and effort associated with setting up the mechanism in question, and (3) the degree of change a specific mechanism brings to the organization at the post-implementation stage.

Further research in the coordination perspective in ERP may indicate the need for more or finer distinctions. However, for the time being, we do believe that this inventory is adequate for understanding the choices for coordination mechanisms with which any ERP adopter is confronted.

4 Roles of coordination mechanisms in RE for cross-organizational ERP

We identified and assessed the impact of the library of ERP-supported coordination mechanisms on the multifaceted problem of balancing flexibility and rigidity typical for ERP implementation. We explored the link between flexibility and rigidity of ERP solutions as documented in the literature [4, 7–9, 12, 18, 19, 21–24, 26–28, 34, 43, 45–55] in order to understand which problems and misalignments could have been spotted and understood early in the ERP project, if the undocumented assumptions have been brought up as part of ERP RE and the coordination requirements have been specified as part of the business requirements document.

The review of these cited sources let us derive the detailed problem dependency map in an intra- and inter-organizational context shown in Fig. 3. As an introduction to the map, we first present a high-level summary of this map in Fig. 2, derived as an abstraction of Fig. 3. Figure 2 says that integration benefits increase through more sharing of standardized and harmonized processes and common data. Also, more sharing decreases the total costs of ownership. On the other hand, the more organizational processes get integrated via the shared process and data environment, the more they get adapted to the default ERP structures, the more the change imposed on the organization, and the more the organizational resistance to it. We believe Fig. 2 maps the basic problem of rigidity versus flexibility in ERP implementation.

Detailed analysis of the reported experiences brought us to the multifaceted representation of the ERP problem space in Fig. 3, in which the boxes represent typical issues that adopters encounter or try to avoid in ERP implementations, and the directed arrows show causality. The references that suggest the 33 links presented here are provided as the Appendix to this paper. Each arrow between two boxes in Fig. 3 is labeled with a number which we use to help the reader to easily find the link in Figs. 3 and 4 when we refer to this link in the text. This number is in brackets when in the text. For example, the link between rigidity and reuse is referred to as to (10).

Figure 3 reflects the opposing forces of process standardization and process diversity [1, 34, 45] and, also, shows that tradeoffs need to be made between

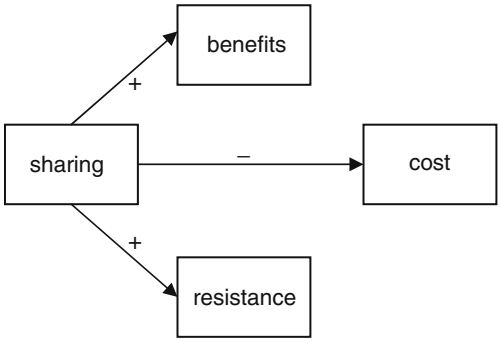

Fig. 2 High-level representation of dependencies in ERP implementations

rigidity and flexibility in ERP solution design. To check if our library of coordination mechanisms can help make those tradeoffs, we investigated how the four groups of mechanisms fit into the multifaceted problem description in Fig. 3.

We made the following observations. First, Fig. 3 not only expands the four boxes of Fig. 2 but it also adds new insights into how benefits are shifting in dependence on the coordination decisions that may favor either process standardization and rigidity of the ERP solution or process diversity and flexibility of the solution. For example, the more diverse the ERP adopters decide to keep their processes and the more flexible the solution they want to design, the more the options for fostering creativity and maintaining the spirit of innovation in the organization [46]. The latter two benefits, however, will be of less consideration, if the ERP adopters decide on a higher level of process standardization. This decision, in turn, favors the realization of the benefits due to sharing, namely reduced transaction costs, organizational transparency, data visibility, data accuracy.

Second, Fig. 3 reflects the fact that benefits from bringing ERP in never come cheaply. It makes it clear as to what price ERP adopters should expect to pay in order to realize the benefits due to sharing or the ones due to flexibility. Getting a more flexible solution means customizing the system to fit the business processes, which means also that cost, like customization (29), maintenance (30) and testing costs (33), and risks, like customization (27), system performance (31) and release lag risks (26), will increase. On the other side, opting for more sharing means incrementally or radically changing cross-organizational business processes to fit the system. It means less customization and maintenance costs as well. The price for implementing these changes, though, comes in the form of costs for managing and coordinating large-scale business process changes and coping with politics, resistance, and corporate inertia [19, 30, 31]. Third, as the coordination mechanisms from our list are available for ERP clients to achieve sharing, they clearly tend to support rigidity, reuse (10), standardization (12), and integration (13). These observations and the fact that the coordination mechanisms from our library are all about sharing encourage us to consider substituting the boxes in the sharing-labeled gray area in Fig. 3 with the groups of coordination mechanisms from Tables 1, 2, 3, and 4. Indeed, if we replace boxes 10, 12, and 13 in Fig. 3 with the groups of coordination mechanisms from our library, one can clearly see how our library fits in and what type of problems the library could potentially help to explain (Fig. 4).

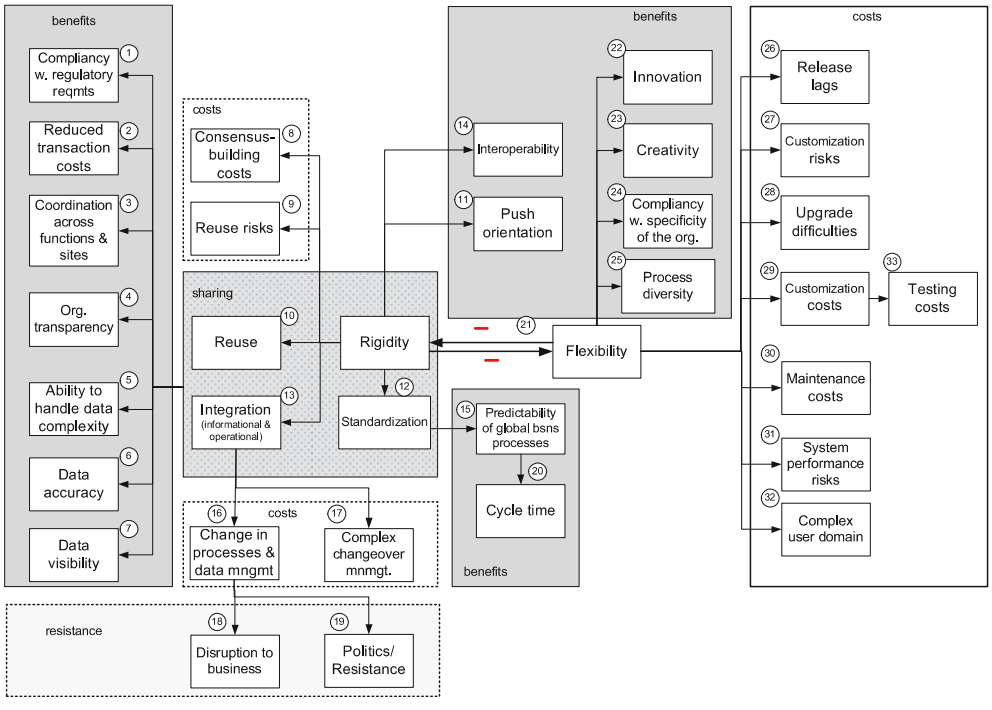

Fig. 3 The problem dependency map. The *numbers* are arbitrary labels to identify the *arrows*

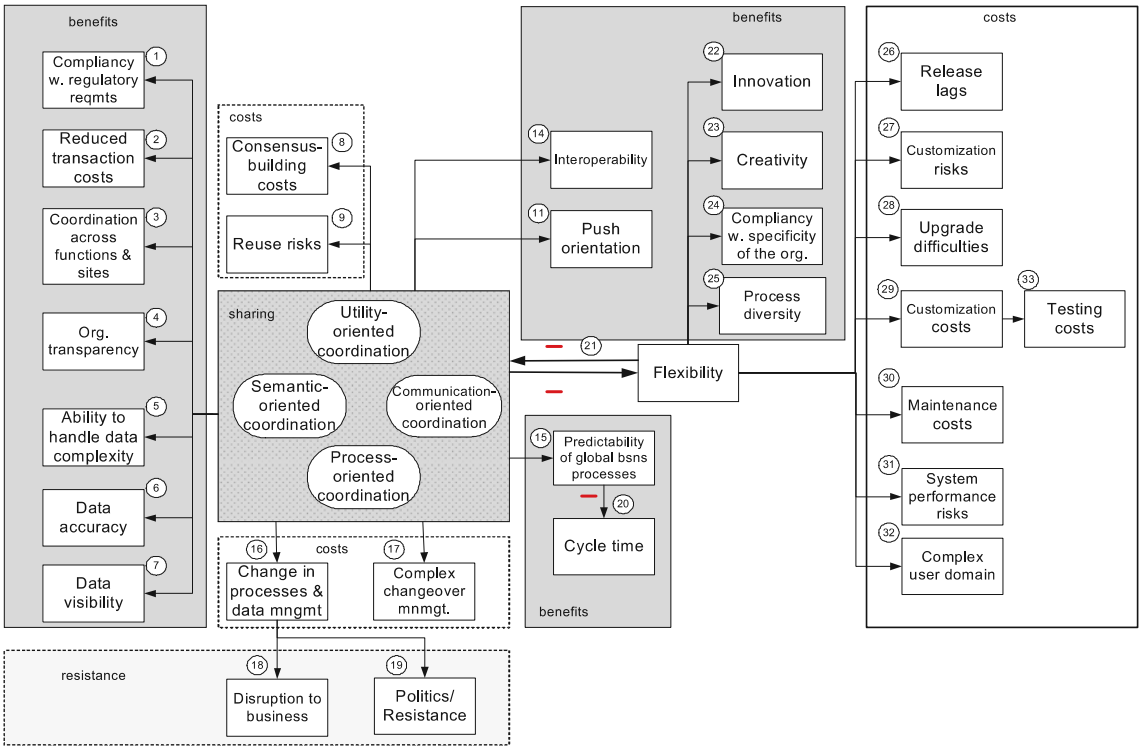

Fig. 4 The problem dependency map: coordination perspective

4.1 Utility-oriented mechanisms

An orientation towards shared benefits pushes the partners in the networked organization towards making decisions regarding (a) how each partner's processes must change, (b) for what benefits, and (c) with whose collaboration. For networked organizations to be economically advantageous, each partner company has to issue and receive transactions directly to and from its ERP system without any human intervention [21]. The orchestration of cross-organizational change requires changes in how things get done internally in each of the partner's organizations (16), as handling business transactions is no longer limited by organizational boundaries. The requirement for internal changes would most likely be in opposition to other forces arising from each partner's organization having its own systems.

4.2 Process-oriented mechanisms

A process orientation in an inter-organizational ERP context means predisposing each company (a) to manage itself along its own business processes, and (b) to make them transparent so that they are capable of being viewed not only by other partners, but also by all the other stakeholders in the enterprise. The first implies shared process ownership and requires redistribution of management responsibility in each of the partner's companies because shared process ownership cannot be imposed on a fragmented organization [21]. Differences between new agreements on shared processes and previously used business practices typically lead to disruption of the business (18), and, ultimately, to increased resistance to change (19). Next, making processes transparent means making employees of all partner companies aware of each others' needs and preferred cooperation patterns. A person is trusted to do the right thing for both the customers and the company when he or she is given the right information and when he or she knows how what he or she is doing affects the entire value chain [1]. As a result, transparent processes enhance the coordination across functions and sites (3) and increase the predictability of the global business processes (15), as corporate oversight will be reduced and unnecessary controls will be eliminated.

4.3 Semantics-oriented mechanisms

Next, a semantics orientation implied by the coordination mechanisms pushes the partners towards adopting a common terminology for those areas of business activity in which the partners want to do things together. This push gives rise to cross-organizational issues like what common data structures to agree upon, what data ownership and mastering concepts to adopt, how much data flow transparency is enough, and who is responsible for data entry and updates [31].

4.4 Communication-oriented mechanisms

Finally, communication-oriented coordination mechanisms tend to support interoperability standards that directly contribute to building 'collaborative communities' [4, 21]. The underlying assumption is that partner companies have to realign horizontally and the variety of shared tasks that are performed requires less flexibility. In a networked context, however, this assumption contradicts the need of each of the partner organizations to dynamically build connections that (a) handle a specific portion of the shared process, (b) change as business opportunities arise, and (c) are taken advantage of and then abandoned as their value diminishes.

The above observations give an indication that each item in the library of coordination mechanisms has its corresponding components in the problem dependency map in Figs. 3 and 4. Thus, our preliminary analysis allows us to draw two conclusions.

- Exploring coordination requirements by addressing the 24 coordination mechanisms in our library in Tables 1, 2, 3, and 4 can be an important step towards detecting the dimensions along which misalignments can arise.
- The library of mechanisms can serve as a preliminary inventory on which critical inter-organizational integration issues, costs, and risks can be surfaced early in the ERP project.

5 Summary and future research plans

The key contribution of this paper is the analysis of the roles that the undocumented ERP built-in assumptions play in inter-organizational ERP RE. We took an inventory of the existing coordination mechanisms and mapped them onto typically encountered problems identified in empirical studies. We attempted to define a RE framework based on a perspective that, we believe, helps the requirements engineers to develop an understanding of the opportunities and issues associated with the ERP coordination mechanisms as undocumented assumptions.

First, our problem dependency map is a problem domain theory; it allows the requirements engineer to reason about the impact of choices.

Second, the undocumented assumptions make the coordination choices more explicit. Our library cannot only facilitate interdisciplinary transfer of knowledge about ERP-supported coordination, but also provides a guide for analyzing organization-specific coordination needs and generating alternative ways to fulfill them. The variety of coordination mechanisms that we analyzed and included in our library is not found together in previous research. Also, we provided a start in organizing these coordination mechanisms. Moreover, we used real life examples to motivate our analyses.

We believe that our model of problem dependencies represents a significant step for understanding what to watch for in order to run a cross-organizational ERP RE process with more predictable architecture alignment results and better chances for success. However, a considerable amount of additional work needs to be done. Each directed arrow in our problem dependency map is based on the extant literature, but some of these have been presented in the literature without sufficient empirical evidence. We view each arrow as a hypothesis to be a subject to empirical validation studies. Thus, for IS scientists, Fig. 3 formulates 33 hypotheses with a very preliminary analysis indicating that it will be useful to validate them empirically. However, we should not and could not expect to get any final answers. We acknowledge the fact that it is hard to perform controlled experiments in business settings and that it is hard to isolate influencing factors in 'uncontrolled' experiments [56, 57]. With these constraints in mind, we adopt the perspective of the empirical software engineering researchers [56, 58] who maintain that any empirical validation study should be seen as a small step forward and as a vehicle we use to gather evidence.

A second line of additional research stems from the observation that for the framework to be useful at application and project level, more analytic capabilities need to be built-in. Therefore, our immediate plans are (1) to use the framework as a vehicle to explain typical misalignment phenomena in cross-organizational implementations, and (2) to refine it based on experiences that we will collect in case studies. Specifically, we are preparing a research initiative that we expect will deliver: (1) a RE approach for cross-organizational ERP implementation and architecture alignment projects, (2) a catalogue of good architecture alignment practices which are guidelines or practical suggestions for improving architecture alignment in networked organizations, and (3) an architecture alignment maturity framework which describes the breath and depth of alignment networked organizations can achieve and, thus, provide focus in introducing the good practices.

A third line of additional research is to apply the transaction cost theory to analyze the costs and benefits of different coordination mechanisms, and to derive RE guidelines from this analysis.

Because our proposal rests on cases from the ERP implementation practice, we believe that the generality of our ideas can be extended to projects implementing other technologies for inter-organizational integration, like data warehouses, workflow management systems, or EAI middleware [44]. This is a fourth line of additional research, which we intend to perform in the field.

Given ERP coordination mechanisms support a variety of intra- and inter-organizational interactions [20], to design a new RE process for cross-organizational ERP implementations, it will be useful to consider alternative coordination mechanisms that could be used to manage data and process sharing. Our analysis has barely explored our library's potential. A fifth and final

question that comes out of this paper and that, we think, is worth exploring, is: in what ways can an ERP system be arranged differently while achieving the same goals? Understanding the coordination problems addressed by a networked business suggests alternative coordination mechanisms that could be used, thus creating a space of possible business process designs. The combined synthesis of all these studies might give requirement engineers insights into the sources of misalignments that have evaded categorization thus far.

Acknowledgement We thank the anonymous reviewers for helping us improve this paper.

6 Appendix

Link	Proposition and relevant references
1	The tighter the integration of operational and informational procedures, the better the compliancy with regulatory requirements [27, 30, 31, 49]
2	The tighter the integration of operational and informational procedures, the lower the transaction costs [21–23]
3	The tighter the integration of operational and informational procedures, the better the coordination across functions and sites [21–23, 27, 30, 47]
4	The tighter the integration of operational and informational procedures, the more transparent the organization [21–23, 30, 31]
5	The tighter the integration of operational and informational procedures, the better the ability of the organization to handle data complexity [21–23, 33, 46, 55]
6	The tighter the integration of operational and informational procedures, the higher the level of data accuracy that the organization can achieve [18, 21–23, 48]
7	The tighter the integration of operational and informational procedures, the better the level of data visibility that the organization can achieve [21–23, 30, 31, 45]
8	The more rigid the solution, the higher the costs of building consensus among stakeholders [46]
9	The more rigid the solution, the higher the reuse risks [7–9]
10	The more rigid the solution, the higher the levels of reuse an organization can achieve [7–9, 33, 38]
11	The more rigid the solution, the stronger the push towards inter- and intra-organizational integration [19, 21–23, 27, 30, 31, 34, 53]
12	The more rigid the solution, the more standardized the business processes [7, 18–23, 30, 31, 34, 35, 40, 42, 55, 60, 61]
13	The more rigid the solution, the tighter the integration of operational and informational procedures that the organization can achieve [7–9, 19, 20, 22–26, 29–31, 33, 34, 42, 43, 46, 47, 50–55]
14	The more rigid the solution, the better the level of interoperability that can be achieved [33, 46, 47]
15	The more rigid the solution, the more predictable the global business processes [21–23, 25, 30, 31, 34]
16	The tighter the integration of operational and informational procedures, the greater the changes in processes and data management imposed on the organization [7–9, 20, 22–26, 29–31, 33, 34, 42, 43, 46–48, 50–54]
17	The tighter the integration in terms operational and informational procedures, the more complex the changeover management processes [19, 22]
18	The greater the changes in processes and data management imposed on the organization, the greater the disruption to business [21–23, 27, 30, 31, 34, 40]
19	The greater the changes in processes and data management imposed on the organization, the more organizational resistance to them and the more the potential sources for political issues [21–23, 27, 30, 31, 34, 47, 50]
20	The more predictable the global business processes, the less the cycle time and the better the control over cycle times [21–23, 33, 34, 45, 61]
21	The more rigid the solution, the less flexibility it offers to business users [7–9, 12, 18–20, 22–26, 29–31, 33, 34, 42, 43, 46, 47, 50–54]
22	The more flexible the solution, the more the options for fostering innovative thinking [25, 46]
23	The more flexible the solution, the more the options for inventing creative ways of working [25, 46]
24	The more flexible the solution, the more compliant it is with the specifics of the organization [30, 31]
25	The more flexible the solution, the more diverse the organizational business processes [7–9, 20, 22–26, 29–31, 33, 34, 42, 43, 46, 47, 50–54, 60]
26	The more flexible the solution, the higher the risks of release lags [43, 51]
27	The more flexible the solution, the higher the customization risks [19, 21–23, 26, 27, 52]
28	The more flexible the solution, the more the upgrade difficulties [2, 5, 21–23, 48, 52]
29	The more flexible the solution, the higher the customization costs [7, 20–23, 51]
30	The more flexible the solution, the higher the maintenance costs [7, 21–23, 51]
31	The more flexible the solution, the higher the system performance risks [30, 31]
32	The more flexible the solution, the more complex the user domain (e.g., the more the data views that need to be consolidated, the more the interfaces that need to be maintained) [7, 21–23, 51]
33	The bigger the scope of customization, the more the testing efforts that are required [7, 21–23, 51]

References

1. Champy J (2002) X-engineering the corporation: the next frontier of business performance. Warner Books, New York
2. Ernst D, Bamford J (2000) Your alliances are too stable. Harv Bus Rev 83:133–141
3. Champy J, Weger J (2005) Reengineering: the second time around. Strategy Leadership 33:53–56
4. Davenport T, Brooks J (2004) Enterprise systems and the supply chain. Enterp Inf Manage 17:8–19
5. Arinze B, Anandarajan M (2003) A framework for using OO mapping methods to rapidly configure ERP systems. Commun ACM 46:61–65
6. Dalal NP, Kamath M, Kolarik WJ, Sivaraman E (2004) Towards an integrated framework for modelling enterprise processes. Commun ACM 47:83–87
7. Daneva M (2004) ERP requirements engineering practice: lessons learnt. IEEE Softw 21:26–33
8. Daneva M (2004) Patterns of success and failure in ERP requirements engineering. In: Proceedings of the 12th international workshop on software measurement, Shaker, Aachen, pp 527–546

9. Daneva M (2002) Using maturity assessments to understand the ERP requirements engineering process. In: Proceedings of the joint international requirements engineering conference, IEEE, Los Alamitos, pp 255–262
10. Esteves J, Pastor JA (2001) Establishing the importance of ERP implementation—critical success factors along ASAP methodology processes. In: Proceedings of international conference on enterprise information systems, IEEE, Los Alamitos, pp 182–187
11. Maiden NA, Ncube C (1998) Acquiring COTS software selection requirements. IEEE Softw 15:46–56
12. Metaxiotis K, Zafeiropoulos I, Nikolinakou N, Psarras J (2005) Goal-directed project management methodology for the support of ERP implementation and optimal adaptation procedure. Inf Manage Comp Secur 13:55–71
13. Rolland C, Prakash N (2000) Bridging the gap between organizational needs and ERP functionality. Requirements Eng 5:180–193
14. Clemons EK, Reddi SP, Rows MC (1993) The impact of information technology on the organization of economic activity: the ''Move to the middle'' hypothesis. Manage Inf Syst 10:9–35
15. Malone TK, Crowston K (1994) The interdisciplinary study of coordination. ACM Comp Surv 26:87–119
16. Miles RE, Snow CC (1992) Causes of failure in network organizations. Calif Manage Rev 34:53–72
17. Powell WW (1990) Neither market nor hierarchy: network forms of organization. Res Organ Behav 12:295–336
18. Bendoly E, Jacobs F (2004) ERP architectural/operational alignment for order-processing performance. Oper Prod Manage 24:99–117
19. Brehm L, Heinzl A, Markus ML (2004) Tailoring the ERP systems: a spectrum of choices and their implications. In: Proceedings of the Hawaii conference on information systems, IEEE, Los Alamitos, CA
20. Clemmons S, Simon SJ (2001) Control and coordination in global ERP configuration. Bus Process Manage 7:205–215
21. Davenport T (2000) The future of enterprise system-enabled organizations. Inf Syst Front 2:163–180
22. Davenport T (2000) Mission critical: realizing the promise of enterprise systems. HBS Press, Boston
23. Davenport T (1998) Putting the enterprise into the enterprise system. Harv Bus Rev 76:121–131
24. Gattiker TF, Goodhue DL (2004) Understanding the local-level costs and benefits of ERP through organizational information processing theory. Inf Manage 41:431–443
25. Hong K-K, Kim Y-G (2002) The critical success factors for ERP implementation: an organizational fit perspective. Inf Manage 40:25–40
26. Madapusi A, D'Souza D (2005) Aligning ERP Systems with international strategies. Inf Syst Manage 22:7–17
27. Markus ML, Tanis C, van Fenema C (2000) Multisite ERP implementations. Commun ACM 43:42–87
28. Moeller CH (2005) ERP II, a conceptual framework for next-generation enterprise systems. Enterp Inf Manage 18:483–497
29. Scott J, Kaindl L (2000) Enhancing functionality in an enterprise software package. Inf Manage 37:111–122
30. Soh C, Kien SS, Tay-Yap J (2000) Cultural fits and misfits: is ERP a universal solution? Commun ACM 43:47–51
31. Soh C, Sia SK, Boh FW, Tang M (2003) Misalignments in ERP implementations: a dialectic perspective. Int J Hum Comput Interact 16:81–100
32. Alstyne M (1997) The state of network organizations: a survey in three frameworks. Org Comput 7:83–151
33. Keller G, Teufel T (1998) SAP R/3 process oriented implementation. Addison-Wesley, New York
34. Hammer M, Stanton S (1999) How process enterprises really work. Harv Bus Rev 77:108–118
35. van Eck PAT, Blanken H, Wieringa RJ (2004) Project GRAAL: towards operational architecture guidelines. Int J Coop Inf Syst 13:235–255
36. Wieringa RJ, Blanken HM, Fokkinga MM, Grefen PWPJ (2003) Aligning application architecture to the business context. In: Proceedings of the 16th international conference on advanced information system engineering (CAiSE 03). Springer, Berlin Heidelberg New York, pp 209–225
37. Biehl M (2004) Selecting internal and external supply chain functionality: the case of ERP systems versus electronic marketplaces. Enterp Inf Manage 18:441–457
38. Curran T, Ladd A (2000) SAP R/3 business blueprint. Prentice Hall, New York
39. Kagermann H, Keller G (2000) MySAP.com industry solutions. Addison-Wesley, New York
40. Sammon D, Adam F (2005) Towards a model of organizational prerequisites for enterprise-wide system integration: examining ERP and data warehousing. Enterp Inf Manage 18:458–470
41. Scheer A-W, Habermann F (2000) Making ERP a success. Commun ACM 43:57–61
42. Selchert M (2004) Enhanced project success through SAP best practices—international benchmarking study. SAP Press, Waldorf
43. Stefanou CJ (2001) A framework for the ex-ante evaluation of ERP software. Eur J Inf Syst 10:204–215
44. Markus ML (2000) Paradigm shifts—e-business and business/systems integration. Commun AIS 4:2–44
45. Attaran M, Attaran S (2004) The rebirth of re-engineering: x-engineering. Bus Process Manage 10:415–429
46. Kallinikos J (2004) Deconstructing information packages: organizational and behavioral implications of ERP systems. Inf Technol People 17:8–30
47. Al-Mashari M (2001) Process orientation through enterprise resource planning: a review of critical issues. Knowl Process Manage 8:175–185
48. Alshavi S, Themistocleous M, Almadani R (2004) Integrating diverse ERP systems: a case study. Enterp Inf Syst 17:454–462
49. Brown W, Nasuti F (2005) What ERP systems can tell us about Sarbanes-Oxley. Inf Manage Comp Secur 13:311–327
50. Grossman T, Walsh J (2004) Avoiding the pitfalls of ERP system implementation. Inf Syst Manage 21:38–42
51. Hirt SG, Swanson EB (2001) Emergent maintenance of ERP: new roles and relationships. Softw Maint Evol: Res Pract 13:373–397
52. Luo W, Strong DM (2004) A framework for evaluating ERP implementation choices. IEEE Trans Eng Manage 51:322–333
53. Ross JW, Vitale MR (2000) The ERP revolution: surviving vs. thriving. Inf Syst Front 2:233–241
54. Stevens C (2003) Enterprise resource planning: a trio of resources. Inf Syst Manage 20:61–67
55. Vosburg J, Kumar A (2001) Managing dirty data in organizations using ERP: lessons from a case study. Ind Manage Data Syst 101:21–31
56. Harrison W (2005) Skinner wasn't a software engineer. IEEE Softw 22:57
57. Zelkovitz M, Wallace DR, Binkley DW (2003) Experimental validation of new software technology. Lecture notes on empirical software engineering. World Scientific Publishing, River Edge
58. Tichy W (2000) Hints for reviewing empirical work in software engineering. Empir Softw Eng 5:309–312
59. Narayanan VG, Raman A (2004) Aligning incentives in supply chains. Harv Bus Rev 82:94–102
60. Kirchmer M (2003) E-business process networks—successful value chains through standards. Enterp Inf Manage 17:20–30
61. Ndede-Amadi AA (2004) What strategic alignment, process redesign, enterprise resource planning, and e-commerce have in common: enterprise-wide computing. Bus Process Manage 10:184–199