Requirements Eng (2005)
DOI 10.1007/s00766-005-0012-7

ORIGINAL ARTICLE

Steven J. Bleistein · Karl Cox · June Verner
Keith T. Phalp

Requirements engineering for e-business advantage

Received: 10 August 2004 / Accepted: 24 May 2005

Abstract As a means of contributing to the achievement of business advantage for companies engaging in e-business, we propose a requirements engineering framework that incorporates a business strategy dimension. We employ Jackson's Problem Frames approach, goal modeling, and business process modeling (BPM) to achieve this. Jackson's context diagrams, used to represent business model context, are integrated with goal models to describe the requirements of the business strategy. We leverage the paradigm of projection in both approaches as a means of simultaneously decomposing both the requirement and context parts, from an abstract business level to concrete system requirements. Our approach maintains traceability to high-level business objectives via contribution relationship links in the goal model. We integrate use of role activity diagrams to describe business processes in detail where needed. The feasibility of our approach is shown by a well-known case study taken from the literature.

1 Introduction

Much evidence indicates that companies are able to gain business advantage over their direct competitors via strategies that leverage IT [1–8]. However, this advantage is made sustainable only through managerial skills and understanding of the use of IT within a strategy for competitive advantage, rather than by superior IT infrastructure or competency of IT staff alone [9, 10]. An organization faces many challenges in order to achieve sustainable business advantage over its competitors: it must not only devise effective business strategies to compete with business rivals, but it is also critical that it ensure its IT systems are in harmony with and provide support for its business strategy [11].

An e-business system enables marketing, buying, selling, delivering, servicing, and paying for products, services, and information, primarily across nonproprietary networks, in order to link an enterprise with other participants; i.e., current and target customers, agents, suppliers, and business partners [12]. One of the challenges of enabling business advantage in an organization's e-business initiative is ensuring that the e-business system addresses the real-world problems the business intends to solve. This means understanding the activities and business processes through which the organization intends to generate value; i.e., its business strategy [11]. Business strategy is thus within the bounds of the requirements problem domain of e-business systems.

An organization's business strategy can be defined as "the understanding of an industry structure and dynamics, determining the organization's relative position in that industry and taking action either to change the industry's structure or the organization's position to improve organizational results" [13]. Business strategy includes both the rationale for and the means by which a business organization competes with industry rivals [14]. The requirements engineering research literature proposes a number of approaches that address various aspects of organizational IT [15–25], all of which ignore business strategy and focus instead on operational rather than strategic concerns. Some literature addresses aspects that include organizational structure and dependency relationships among actors in a system [15, 16], economic and business value analysis [17], alignment of business processes with business goals [18–20], elicitation of business goals from which to derive requirements [21], and cost-benefit analysis of requirements risk [22–24].

S. J. Bleistein (✉) · K. Cox · J. Verner
Empirical Software Engineering Research Program,
National ICT Australia, Sydney 1430, Australia
E-mail: Steven.Bleistein@NICTA.com.au

S. J. Bleistein · K. Cox
School of Computer Science and Engineering,
University of New South Wales, Sydney 2052, Australia

K. T. Phalp
Emprical Software Engineering Research Group,
Bournemouth University, Dorset, UK

Other research proposes a requirements methodology that emphasizes "early requirements analysis," to "capture and analyze the goals of stakeholders" [25] with no capture, much less analysis of the strategic goals of executive management stakeholders when applied in an industry study [26]. Use of the requirements approaches mentioned above provide requirements engineers with little means to validate system requirements against the intentions of executive management stakeholders, who make the business strategy that they expect an eventual IT system to support.

We do not propose that a requirements engineer create an organization's business strategy to gain advantage over its business rivals. However, a requirements engineer can contribute to an organization's competitive advantage by ensuring that requirements of its e-business systems align with, provide support for, and enable its business strategy. To achieve this alignment, requirements engineers must at a minimum understand the business strategy, and have a means of representing strategic context within a requirements engineering framework in order to integrate and link the organization's business model with the model of systems requirements.

We thus propose a requirements engineering approach for e-business systems that incorporates business strategy and business process dimensions as a means of contributing to a company's achievement of competitive business advantage. This paper extends [27, 28]. Our approach integrates Jackson's *problem diagrams* [29] with goal modeling. We employ Jackson's *context diagrams* to describe business problem context, and goal-modeling to capture all desired properties of the system, including business goals, strategic objectives, activities and any other business or systems requirements. We leverage the paradigm of projection in both approaches as a means of simultaneously decomposing both context model and goal model parts of the requirements problem down to system requirements. We use role activity diagrams to model business processes where needed.

For initial validation of a new technique in requirements engineering, it is acceptable practice to apply the technique to an exemplar appropriate to demonstrating the specific capabilities of the technique [30]. However, we have found the requirements engineering research literature to be devoid of well-documented examples of organizational IT that encompass business strategy. We therefore developed a requirements engineering example suited to the objective of demonstrating a capability of verifying and validating requirements in terms of alignment with business strategy. We base the example on research on Seven-Eleven Japan's (SEJ) IT appearing in both management and information systems literature [12, 31–36]. This example allows us to verify that our approach can be used to model both system requirements and business strategy, and to demonstrate alignment between these.

The rest of this paper is organized as follows: Sect. 2 presents the background to our work; Sect. 3 describes our approach and shows how a Problem Frames approach, goal modeling, and business process modeling (BPM) are integrated; Sect. 4 presents a proof-of-concept case study from the literature describing SEJ; Sect. 5 evaluates the approach and the case study; Sect. 6 offers conclusions.

2 Background

This section discusses previous research and the requirements engineering techniques used in our approach. Section 2.1 reviews requirements engineering research that addresses e-business issues. Section 2.2 introduces problem frames. Section 2.3 discusses uses of goal modeling to refine business goals and strategy to system requirements. Section 2.4 reviews BPM in requirements engineering.

2.1 Requirements engineering for e-business

Most requirements engineering research addressing e-business does so indirectly in the context of requirements for Web-based systems or Web applications development [37–40]. Web-based systems research, however, focuses on architectural, usability, and design concerns rather than business aspects. Also, by virtue of being "Web-based", this research effectively excludes issues of e-business systems that do not use the Internet for connectivity or Web browsers for user interfaces. Other research addresses issues of value analysis of e-commerce applications development, but neglects *requirements analysis* [17, 41]. A different view is taken in [42], in which a requirements-driven systems engineering approach that considers organizational aspects in an industrial e-business project is presented; however, the focus is primarily on dependencies between actors and goals rather than on business strategy and processes.

Overall, with the exception of [42], what little e-businesses systems requirements engineering research there is, fails to propose concrete requirements engineering approaches. The methods and techniques proposed tend to focus on producing end-products of architectural and usability design or value analysis rather than on producing system requirements. None of the research directly addresses issues of business strategy and business process, upon which competitive business advantage is based.

2.2 Problem frames

Problem Frames are used to describe real world problems in the context of known software solutions [29]. The Problem Frames approach captures and classifies software development problems. It structures the analysis of the problem within its problem space. It describes what is in the real world and how the machine is intended to change or guarantee real-world conditions such that they meet the requirements. With its emphasis on problems rather than solutions, the Problem Frames approach uses an understanding of a problem class to allow the problem owner with his specific domain knowledge to drive the requirements engineering process.

A *Problem Frame* is the means by which recurring classes of problems in software development are captured. Problem Frames are akin to design patterns [43], except that they describe problems rather than solutions. They are thus a means of understanding and describing the problem context for which software will provide a solution, either entirely or in part.

Figure 1 illustrates the basic elements of the original Problem Frames model. The *Real World Problem Context* provides information about the structure, processes and tasks that are true of the problem domain. The *Requirement* states which properties are to be true given a built software solution; i.e., the *Machine* that will work within its real-world context. The connection between the real world problem context and the machine is represented by shared phenomena at the boundary between the problem and the solution.

Thus, there are three separate descriptions documented: the problem context, the requirements, and the specification. These are couched in terms of two *moods* (in the grammatical sense): *indicative* referring to *problem context*, used to represent what is always true of the problem domain, and *optative* referring to the *requirement*, used to represent how we would like the world to be upon deployment and use of the machine. A *problem diagram*, containing the same elements as in Fig. 1, describes a particular software problem showing the problem parts consisting of *problem context* and the *requirement*. *Problem frames* are derived through decomposition of *problem diagrams*. Even though the software/hardware system may consist of multiple devices or computers, for the purpose of a context diagram these are represented as a single *machine*. Decomposing *problem diagrams* reveals a greater level of detail, including separate distinct machines.

Much research on Problem Frames has tended to focus on what the requirements engineer does when he or she has determined a problem frame and wants to engineer a solution from there [44–46]; however, this type of research does not to address issues of higher-level problem analysis and decomposition. Other problem frames research in contrast has focused on higher-level problem analysis specifically in the context of e-business systems [28, 47–53]. The research presented in this paper extends this category of Problem Frames research. For a complete review of the Problem Frames literature, see [54].

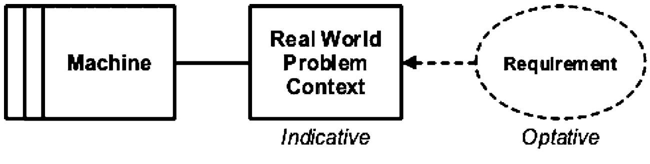

Fig. 1 Elements of the Problem Frames model

2.3 Goal modeling, business objectives, and strategy

Goal-oriented modeling techniques in requirements engineering use projection for goal refinement. As such, goal modeling serves as a means of linking high-level strategic goals to low-level systems requirements [55]. A number of goal-oriented techniques have been proposed for modeling business goals and objectives in requirements engineering [21, 56–59]. While some of this research treats business goals as discrete, independent entities, other approaches assemble business goals and their sub goals into structures representing complete business strategies, and then anchor requirements to the strategy model [60, 61].

However, despite their application to modeling business goals and strategy, goal-oriented modeling techniques have a number of shortcomings. First, they tend to be deficient in describing problem context [60]. Second, goal models tend to bloat quickly, threatening manageability [61, 62]. This bloating is potentially a show-stopping problem in the development of a large e-business system. Third, as goals are inherently hierarchical, it can be difficult to discern where a business goal is situated in the hierarchy and how it relates to the business problem context. Moreover, for every business goal, there is always a discoverable super goal. Thus goal-modeling requires upper bounding of the problem domain [29, 63].

2.4 BPM in requirements engineering

A business process is a ‘‘set of partially ordered activities intended to reach a goal'' [64]. Requirements engineering techniques have been used to address issues of business process; however, most of these techniques are inadequate when applied to e-business systems.

At the highest level, structured analysis (SA) models processes within a context diagram [65, 66]; however, in a context diagram SA considers only data flows between external entities and the system, and thus effectively ignores processes and interactions between external entities [29, 67]. In e-business systems, the *participants* are the external entities. Describing their direct relationships and interactions is fundamental to understanding the e-business problem [12].

Use cases are sometimes employed to describe business processes [68], but they have been viewed in the requirements engineering community as inadequate for describing complex business requirements [46, 69], including processes. In contrast to standard use cases, Buhr's *use case maps* provide an expressive notation to represent complex architectural behavior processes [70, 71]. They treat processes primarily as machine activities, typically ignoring the human and organizational aspects of business processes, which are critical to describing e-business systems.

Eriksson et al. [72] propose modeling business processes with UML activity diagrams; however, activity diagrams were originally designed to describe how activities affect the state of software-focused objects, not business processes. Eriksson et al. do not explain why UML activity diagrams are better suited to describing business processes than recognized BPM notations.

To overcome the inadequacies of requirements engineering approaches to BPM, we employ role activity diagrams [73], a well-recognized BPM notation. A role activity diagram (RAD) has various components, the most common of which are illustrated in Fig. 2 below.

All roles start in an initial state. For example, role A starts in some initial state and then has an event, an action, 'do work', which is independent of other roles. On completion of the work, the role would be said to have moved to a new state of work completed. Although states are often omitted (as in Fig. 2), a formal view would be that the event, and action of role A, has a pre-state of 'initial' and post state of 'work completed'.

Some work is then delegated to a colleague. This is a shared event. Although the mechanism of delegation is immaterial, the result is that both roles involved move to the state of work delegated. These shared events are termed interactions. Although there is no sender and receiver as such, role A is said to initiate (be the active role) whilst role B is passive in this interaction. Role B is then in a state to independently 'do work'. Role B then 'returns work to colleague', role A, who is in a state to receive it and so on.

3 Addressing the e-business problem

In this section, we present a framework to address requirements for e-business systems such that the requirements align with and provide support for the business strategy that the e-business system is intended to support. The framework consists of Jackson problem diagrams in progression [29], integrated with a goal model, and RADs [73]. An organization's business strategy at the highest level can be represented as a Jackson problem diagram. The context part of the problem diagram represents business model context. The requirement part of the problem diagram represents the objectives of business strategy in the form of a goal model. The context diagram defines the scope of the e-business model by identifying the e-business model participants and the relationships between them. Business context is decomposed down to system context in parallel with goal refinement from strategic objectives to system requirements in a progression of problem diagrams. Use of the goal model is important because it enables traceability from low-level system requirements to high-level strategic concerns via goal contribution links [55], which is critical when validating strategic alignment of requirements. RADs are linked to goals and context, and used to describe business process detail in cases where process detail is critical to understanding the requirements.

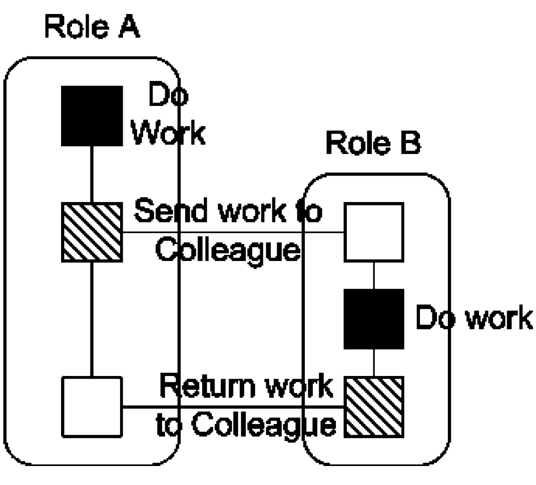

Fig. 2 Elements of a role activity diagram

The rest of this section is organized as follows: Sect. 3.1 justifies application of the Problem Frames approach to business strategy. Section 3.2 discusses both the idea of a *progression of problems* and why it is appropriate to the e-business domain as a means of expressing context. Section 3.3 shows how goal modeling can represent the requirement set. Section 3.4 integrates role activity diagrams to describe business processes.

3.1 Business strategy as a problem diagram

Oliver defines business strategy as "the understanding of an industry structure and dynamics, determining the organization's relative position in that industry and taking action either to change the industry's structure or the organization's position to improve organizational results" [13] as mentioned in Sect. 1. This definition of strategy is similar to Jackson's definition of a *problem diagram*. As discussed in Sect. 2.2, Jackson describes the world according to two moods; *indicative* mood, which refers to "the way the world is" or problem context, and *optative* mood which is "the way in which we want to change the world" or the requirements [29, 67]. Oliver's "understanding of an industry structure and dynamics", and "determining the organization's relative position in that industry" is Jackson's *indicative* mood. "Taking action either to change the industry's structure or the organization's position to improve organizational results," is Jackson's *optative* mood, the way in which the organization desires to change the real world. We thus propose that an e-business strategy can be represented as a *problem diagram*, in which the e-business system is represented as the *machine*, as illustrated in Fig. 3.

We recognize that an e-business system is in fact a collection of many machines working in concert, but at this level of abstraction, we represent the entire system as one *machine*, in accordance with Jackson's rule [29].

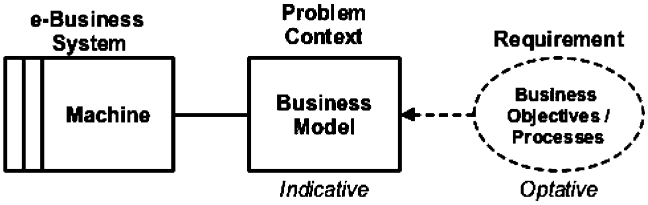

Fig. 3 Business strategy as problem diagram

The participants in an e-business system represent *domains of interest* [29, 67] within problem context. As noted above, the *requirements* are the *optative* part of the strategy; i.e., the objectives, activities, and business processes of an organization through which it attempts gain advantage over its business rivals and generate value. We consider all optative properties of a system to be requirements, including business goals, objectives, activities, business processes, policies, and any other business or systems requirements.

3.2 A progression of problems

E-business problems at the highest level of business strategy are in fact very distant from the *machine*. To refine requirements from high-levels of abstraction down to the *machine*, the paradigm of a *progression of problems*, illustrated in Fig. 4, is particularly useful. The complexity of e-business systems as well as the need to align requirements with the highest levels of business strategy has in fact pushed the requirements problem into what Jackson would describe as "deep in the real world" [29].

The *domain* DA in Fig. 4 represents the *indicative* properties of the e-business problem context at the level of business strategy. *Requirement* RA represents the *optative* properties of strategy. Domain DB is a projection of the context in DA at a lower level of abstraction. Through analysis of DA and RA, it is possible to find a requirement RB that refers only to DB while satisfying

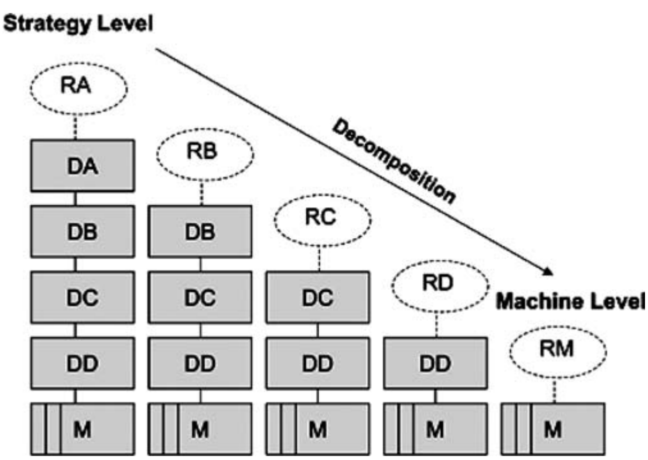

Fig. 4 A progression of problems (adapted from [29] p. 103)

RA [29]. DB represents the projection of DA, but at a lower level of abstraction. Similarly, *domain* DC is a projection of the context in DB, and through analysis of DB and RB, it is possible to find a requirement RC that refers only to DC while satisfying RB. Through this process of analysis, ultimately it is possible to find a requirement RM that refers only to *machine* context M. RM contains the requirements for the system.

While the paradigm of a *progression of problems* serves as a powerful framework for decomposing e-business strategy down to *machine* requirements, the Problem Frames approach provides little explicit linkage between requirements at different levels of the progression. Relating requirements from higher to lower levels of abstraction is important in problem decomposition of complex systems, in which complex problems whose requirements may be very abstract, are projected into increasingly detailed sub-problem diagrams, to make the requirements more concrete. In order to ensure that system requirements are indeed in harmony with and provide support for business strategy, explicit traceability from lower level requirements to the highest level is necessary.

In the example above, requirement RB must satisfy requirement RA, and RC must satisfy RB, which satisfies RA, and so on. However, while Jackson proposes analysis of DA and RA in order to find RB [29], a framework for doing so is not described. The detailed description of explicit linkages tracing between requirements in problems and those in the projections of their sub-problems in Jackson's progression of problems is effectively missing. We thus propose integrating the use of goal modeling to address the missing direct linkages between requirements in progression, which we describe in Sect. 3.3 below.

3.3 Integrating goal modeling with progression of problems

Goal modeling is a useful technique to describe explicit linkages between lower-level requirements and higher-level objectives [55]. Goals represent objectives that the system ought to achieve, and refer to properties that are intended to be ensured [63]. Goals are thus requirements at a higher level of abstraction, and we treat goals as *optative*, as we would a requirement, equally bounded by the problem domain [29, 67]. Because goals can be formulated at different levels of abstraction, from high-level strategic concerns to low-level technical ones [55], goal modeling is a useful tool for describing the requirement part of problem diagrams in a progression of problems. We therefore propose the integration of goal modeling with problem diagrams as a means of helping to ensure that lower-level requirements of the system are in harmony with, and provide support for, higher-level requirements of business strategy. The integration of a goal model with a progression of problems is illustrated in Fig. 5 below.

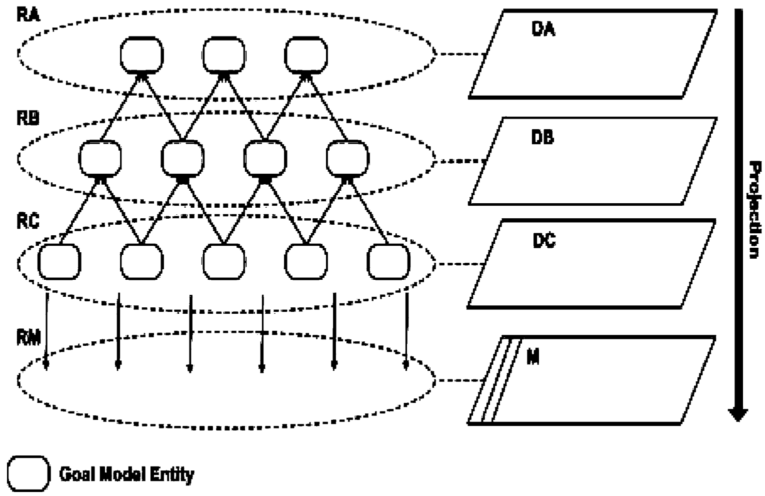

Fig. 5 Goal model integrated with progression of problems

The requirements, in the optative mood, at each level of the progression are described in terms of a portion of a larger goal model. The goal portions represent requirements at a level of abstraction equivalent to that of the *domain* to which they refer within the progression of problems. Each goal entity refers to specific *domains of interest* within the referred domain. The goal model enables explicit connections to requirements at adjacent levels in terms of super goals and sub goals. The sub goals are in fact projections of their super goals, and satisfaction of the sub goals guarantees satisfaction of the super goals in the same way that satisfaction of RB guarantees satisfaction of RA in Fig. 5. The context diagrams in the progression of problems, DA, DB, DC, etc., complement the goal model by providing problem context at various levels of abstraction with explicit linkage to requirements. Moreover, the integration of context diagrams with goal modeling also improves manageability of goal models in complex systems. The sub problems enable a decomposition of the requirements, represented as portions of the goal model, into manageable chunks, while still maintaining explicit linkages. Also, individual business goal entities are situated in the context of the problems at explicit levels of problem abstraction.

3.4 Business process model

Jackson's problem diagrams, even when augmented with goal models, are inadequate for describing business processes. While we can represent discrete activities that make up a process as optative properties of a problem, there is no notion of order in problem diagrams or goal modeling to enable description of these activities as a process. In addition, goal models when decomposed down to the level of atomic activities in a process inflate at the bottom-level and become unmanageable. We thus propose integrating BPM to alleviate these concerns as shown in Fig. 6.

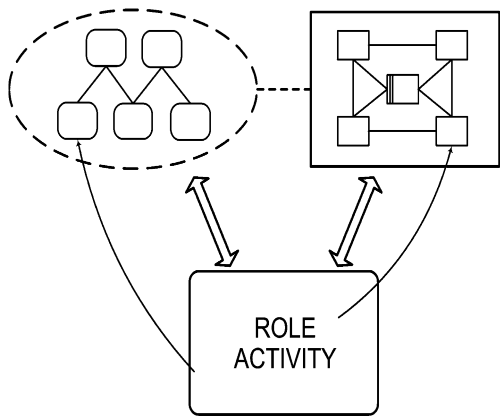

Fig. 6 RAD providing business process detail to problem diagram

A RAD describes business processes in great detail, including both indicative and optative properties, in a clear and succinct manner. A role represents a domain of interest, an indicative property of domain context. An activity, which is an action or interactions between roles, represents optative properties of behavior of the system. The goals that business processes achieve are represented in the goal model.

A role can be viewed as either black box or in more detail as white box, allowing representation of different abstraction levels of a particular problem contained solely within the role. This gives a requirements analyst the opportunity to understand not only how domains interact but also how they act internally.

It might be argued that if we can show both indicative and optative properties in a business process model, this would then be sufficient on its own to describe an e-business problem; however, business processes only deal with discrete goals that the process's activities achieve. Without the wider perspective of the goal model there is no notion of where the process fits in the overall business strategy. Also, while describing business process is important for certain aspects of the e-business problem, not all aspects of the problem involve discrete, ordered processes.

4 Proof of concept case study: SEJ

We illustrate our approach using a business case study example of Seven Eleven Japan's e-business system. As mentioned in Sect. 1, we developed this example from numerous sources in the literature [12, 31–35]. This example is compelling because the highest level of Seven Eleven's business strategy to gain an advantage over its competitors via its use of IT has direct implications for the lower level requirements of its e-business system.

4.1 Overview of SEJ's business strategy

Seven-Eleven Japan, like its US progenitor, manages a national network of convenience stores. Unlike Seven-Eleven USA, SEJ generates value by leveraging and controlling ownership of information to optimize efficiency across a value chain with an unparalleled manner of sophistication. SEJ positions itself in the center of a value chain that includes suppliers, third-party logistics providers, and franchise shops, all of whom are independently-owned companies, yet all of whose objectives are maximizing throughput of products ultimately sold to franchise shop end-customers.

SEJ bases its strategy for competitive advantage on an extremely high level of competency at anticipating consumer purchases store-by-store, item-by-item, hour-by-hour, and then enabling franchise stores to provide customers with products they want when they want them. SEJ's strategy leverages IT to accomplish its strategic objectives, and gain a competitive advantage over its business rivals. SEJ's ownership of information enables sophisticated supply chain management to reduce inventories, lower costs, and increase sales. SEJ moves information between itself and its partner companies via an ISDN network (incidentally, SEJ's e-business strategy neither uses the Internet for its network, nor Web browsers as human interfaces). To better understand customer demand, SEJ actively gathers and analyses purchasing information in real time, and correlates this with other social and environmental factors, including neighborhood demographics, planned local events like festivals, and the weather. SEJ then uses an acutely tuned just-in-time delivery system to meet that demand, generating remarkable value. It is these activities and their objectives that constitute the optative part of the SEJ e-business problem.

4.2 Progression of problems of SEJ

We now examine the progression of problems of SEJ's e-business system from the top, macro-level of business strategy down to the machine devices used in the franchise shops as illustrated in Fig. 7 below. Note that for the purposes of describing the approach, we are concerned only with a particular sub-problem within Fig. 7 and that Fig. 7 describes only part of the SEJ e-business system problem. The macro-level business strategy is the top-level problem that is deepest into the world. It is here that we bound our problem, because it is here that SEJ bounds its problem.

The progression of problems consists of an indicative part, which we describe as a progression of context diagrams, and an optative part, which we describe as a progression of goal refinements in a goal model. We chose to represent the goal model in Goal-Oriented Requirements Language (GRL) notation [58, 74] because of its expressiveness in representation of both abstract and non-abstract goals, tasks, and resources, which we felt would be helpful in modeling requirements for SEJ's complex e-business system. GRL integrates two previously developed goal-oriented requirements notations: the NFR [62] and *i** frameworks [75].

Please note that the entities in the goal model are grouped by dashed-line ellipses, RA, RB and RC, as shown in Fig. 7. The goal entities within the ellipses represent requirements referring to context diagrams in the progression at equivalent levels of abstraction, DA, DB and DC, as shown in Fig. 7. The integration of the goal model and the context diagram at each level in the progression presents a *problem diagram* for that particular level of abstraction. We now describe this progression in finer detail.

4.2.1 Description of requirement set RA and domain DA

Our aim in the example presented here is to demonstrate tracing to and alignment with requirements at higher-levels, such as those related with business strategy. To understand the optative part of the business strategy, we explore the goal model at its highest level RA. SEJ's requirement is to *Stock products that customers want when they want them according to changing needs*. This meets the goals *Reduce lost opportunity/customer* and *Minimize unsold perishables* and is achievable by *Just-in-time delivery*, which in turn supports the goals of *Maximize use of limited floor space, Shorten inventory turns* and *Maintain constant freshness of perishable goods*. These can be met by *Development of effective decision support systems*. The scope of the requirement set can be understood only by an exploration of its context.

The corresponding context diagram DA shows the *machine* domain *SEJ Value Net Integrator System*. This retrieves the *Just-in-time* data it needs from the *Franchise Store* domain, interface *a*. To know what to deliver just in time, a goal in RA, the needs of the *ShopCustomer* must be understood, interface *b*. The *machine* domain provides the necessary information to the *Supplier*, interface *f*, which in turn uses a *Logistics Partner* to deliver the goods, supporting the goal *Just-in-time delivery*. The shared phenomena *e* represents the delivery schedule, the goods themselves and delivery address. The *Logistics Partner*

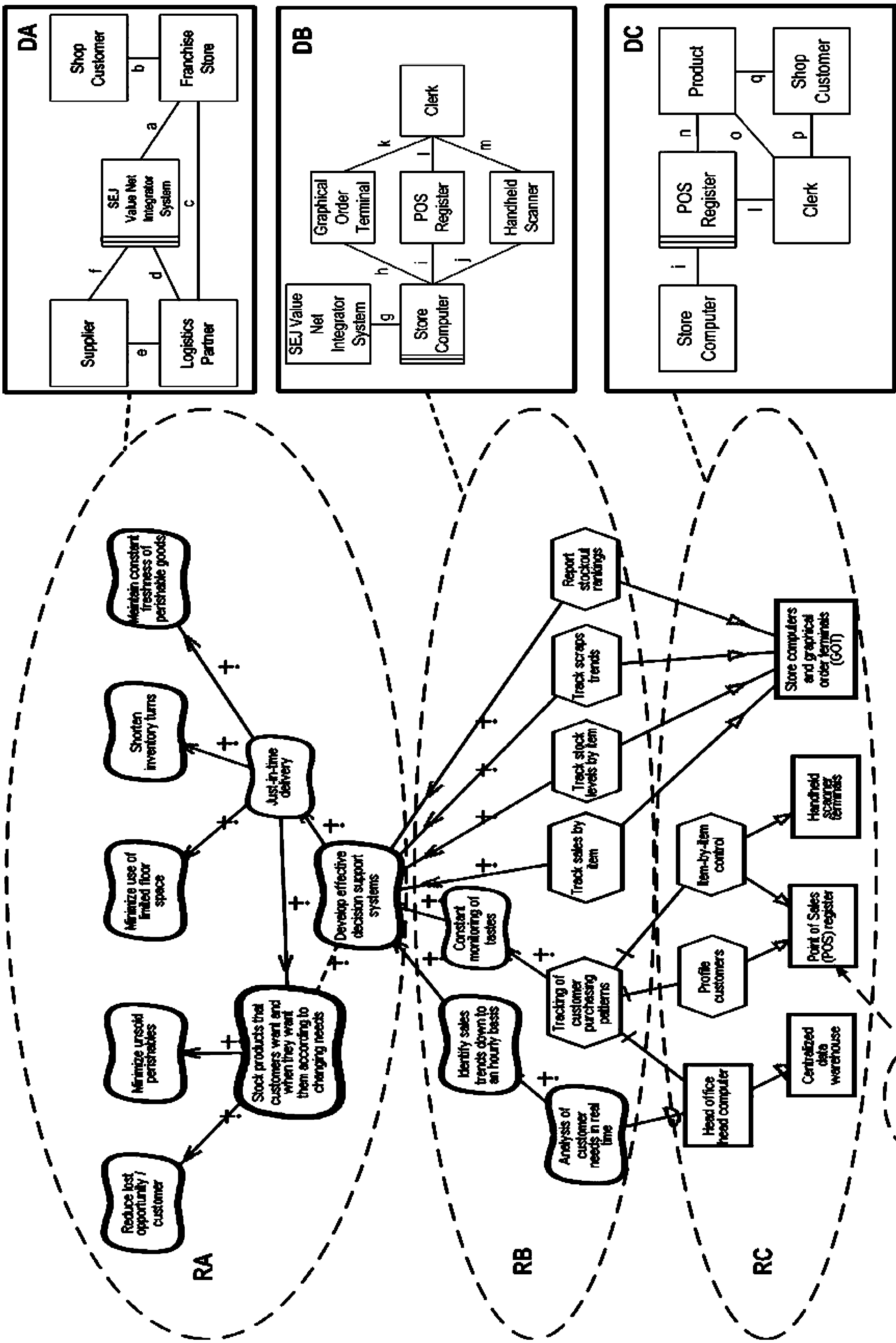

Fig. 7 SEJ progression of problems: integrated goal model and context diagrams

must also provide its schedule details back to the *SEJ system*, interface *d* about its delivery, interface *c*. The *Franchise Store* also provides details of the sales of perishable goods, how the store is stocked and how this affects the sale of goods. Inventory and sales information is highly automated; its requirements can only be understood by decomposing the problems.

4.2.2 Description of requirement set RB and domain DB

To meet the goal to *Develop effective decision support systems* in RA that helps achieve the requirements of RA, the Requirement Set RB has three goals and a number of supporting tasks. RB focuses on how the *Franchise Store* can work effectively to meet SEJ's requirements. Thus, in order to *develop effective decision support systems* one must *identify sales trends down to an hourly basis*. To meet this requirement one must have *analysis of customer needs in real time*. Allied to *sales trends* is the *constant monitoring of tastes* . The context diagram at DB is a progression from that of DA. To meet the Requirement RB, DB's context shows the composition of the *Franchise Store* of DA. The *Graphical Order Terminal (GOT)* is a device that allows the *Clerk* to track and report on sales and stock that is held in the store, interface *k*. The *GOT* accesses the *Store Computer* by interface *h* in order to do this. The *Handheld Scanner* is a device that allows the *Clerk*, interface *m*, to scan product barcodes of items on the shelves and in the shop storeroom for *Item-by-item control* of inventory. The *Handheld Scanner* accesses the *Store Computer* via interface *j* in order to provide regular updates. The *Clerk* also interacts with the *Point of sale register (POS)* to take customer purchases, interface *l*, and the *POS* informs the *Store Computer* of these, interface *i*, which are described in the next paragraphs. The *Store Computer* processes and then relays information to the *SEJ Value Net Integrator*, in real time, interface *g*, thus meeting the goals in RB critical to the success of the strategy captured in RA.

4.2.3 Description of requirement set RC and domain DC

Referring to the goal model, the requirement set RC contains a number of devices. However, our focus in this example is the *POS*, represented as a GRL *resource*. It has two tasks that have to be performed to satisfy *Tracking customer purchase patterns* in RB. These are *Profile Customers* and *Item-by-item control*. We thus present the domains of interest in the context of the *POS* in DC.

In DC, the *ShopCustomer* takes his *Products*, interface *q*, to the *Clerk* for purchase, interfaces *p* and *o*, and then pays for them, interface *p*. The *Clerk* scans the *Product* information via the barcode, interface *n*, into the *POS*. The *Clerk* enters the *Shop Customer* profile and payment details into the *POS*, interface *l*. Finally, the customer profile and product information is sent to the *Store Computer* by the *POS*, interface *i*, for storage, processing, and transmission to SEJ, meeting its goal in RB, *Analysis of customer needs in real time*, and task, *Tracking of customer purchasing patterns*.

While in our model, our requirement RM refers to the *POS* register directly, we recognize that the *POS* is in fact a fairly complex machine. Its problem context would likely be decomposed into a domain DD, and further into recurring problem frames. We do not illustrate this here, because this is not the focus of our paper. Jackson describes numerous examples of this type in his book [29].

4.3 Business process representation

The business process is shown as a RAD in Fig. 8. The *Shop Customer* in the RAD presents his products for purchase to the *Clerk*. The *Clerk* scans the products to record the product details in the *Point of Sales Register* (*POS*), which keeps a running price total. When all products have been scanned, the *POS* presents a final total amount payable. The *Clerk* informs the *Customer* of the amount payable, who then presents payment to the *Clerk*. We do not discuss how payment is made since this is another sub-problem and not part of our example.

In order to achieve the task *Profile customer*, the *POS* prompts the *Clerk* to enter the *Customer's* age, followed by gender, prior to concluding the payment transaction. One of the requirements of the *POS* is that the cash drawer not open and sales cannot complete until the *Clerk* has entered this information. The *Clerk* then enters the payment into the *POS*, which prints a receipt. The *Clerk* hands this to the *Customer* who takes his shopping and leaves.

This process thus meets the objectives *Item-by-item control* and *Profile Customer*. The *POS*, meanwhile, *Registers the Customer sale* with the *Store Computer*. This interaction contains both product information and the profile information of the customer who bought the products. As products are scanned, the *POS* records the bill of sale enabling *Item-by-item control* of inventory as products are sold off the store's shelves. Product data is also associated with the customer profile data, and time and date of purchase, which helps enable *Tracking of customer purchasing patterns* in RB.

The RAD describes the activities in RC involved in achieving the requirements *Profile customers* and *Item-by-item control*. These activities can be traced to higher-level objectives in the goal model. The roles in the RAD are taken from the context diagram DC ensuring that the process model describes both the optative and indicative properties at the equivalent level in the progression of problems.

4.4 Case study summary

The context diagrams in the progression of problems and the goal model mutually complement and support

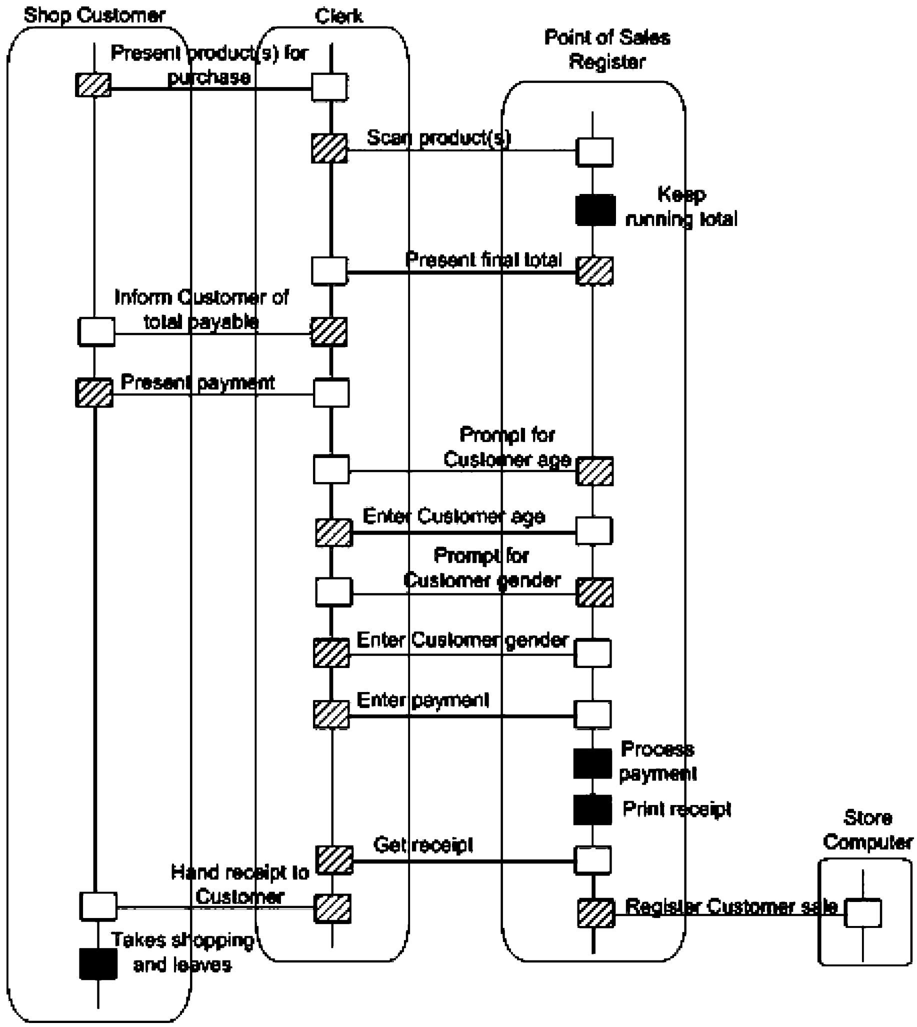

Fig. 8 Business process RAD describing RC-DC

each other. Goal modeling provides explicit linkage between requirements in problem diagrams at different levels of abstraction as determined by the context diagrams. This integrated approach thus offers a means of helping ensure that requirements are in harmony with and provide support for business strategy. This in turn helps enable business advantage assuming that the strategy is correct, as requirements are aligned top-down from the highest level of problem context and business strategy.

Problem context diagrams improve manageability of goal models of complex systems, by breaking down requirements into more manageable goal model sections. Moreover, the context diagrams enable us to explicitly situate individual business goal entities in the context of the problems they address at equivalent levels of abstraction. Finally, the context diagram at the top-level of the progression of problems bound the goal model as it bounds the problem from SEJ's point of view.

However, representing e-business problems with problem diagrams is not enough. A problem diagram provides no means of describing process. The RAD in Fig. 8 describes explicitly how the interactions between the domains in DC achieve requirements in RC in a business process. Understanding this process in detail is fundamental to understanding the nature of the e-business problem.

5 Evaluation

Integrating goal-modeling techniques with the Problem Frames approach is not without its difficulties. We found some awkwardness in mapping GRL goal models to context diagrams, particularly regarding GRL resources, such as the POS register appearing in RC in Fig. 7. A GRL resource is a type of entity that attempts to describe a limited aspect of problem context within the goal model. A GRL resource is thus also represented as a domain of interest in a context diagram. This representation led to a degree of redundancy in domain representation and unnecessarily cluttered the goal model. We found it difficult to reconcile GRL's way of describing context with that of the context diagrams in the Problem Frames approach. While each approach of goal modeling and context diagrams employs a projection paradigm of decomposition, they do not treat context entities equivalently. In the GRL goal model, the entity, a resource, appears only once. Goal modeling with GRL does not allow multiple representations of the same resource entity. The role of a resource at different levels of a goal model is made clear via GRL contribution links. Yet, the same resource, represented as a domain of interest in a context diagram, may appear in other context diagrams, as it may be re-expressed in different projections of the problem. For example, the POS domain of interest appears in DB, it does not appear at all in RB, and it appears only in RC in Fig. 7. Yet the POS appears as a domain of interest in both DB and DC because it is important to understanding the context of the goals in both RB and RC. This multiplicity causes some confusion regarding a resource that is relevant to problem context at multiple levels of abstraction, and also creates inconsistencies between the requirement and context parts of the Jackson problem diagrams. We suggest that when integrating goal-modeling techniques with the Problem Frames approach, it may be better to refrain from expressing problem contextual entities, such as resources, in the goal model even though such entities may exist in the goal-modeling notation chosen. Instead, it may be better to show explicit links between goals and specific domains of interest in the context diagrams. This alternative expression has the added benefit of reducing clutter and forcing a separation of concerns.

Figure 7 illustrates the connection between requirement and context in a general way for the sake of simplifying the large diagram of the progression of problems. A rigorous application of Jackson's Problem Frames approach would require that we connect discrete requirements to the specific domains of interest that those requirements constrain or to which they refer. It would be difficult to illustrate all of those connections in Fig. 7 without hopelessly cluttering the diagram. We suggest that it might be better to illustrate the connections at each level of requirements and context in a separate, detailed diagram for that level represented as a projection of the big picture diagram, while maintaining reference to the big picture diagram. Modeling and analysis in this manner is consistent with [29], in which Jackson contends that to understand a problem it is necessary to decompose it into its constituent parts and recommends re-composition only when needed, and not before.

In the SEJ case study we performed a requirements analysis only to a limited extent. The requirements identified in RC are still at a relatively high level of abstraction. However, we believe that we have refined the requirements problem down to a level of abstraction equivalent to the starting point of many examples found in Jackson's book [29], and as such we do not continue the process here. We suggest that once a problem diagram has been refined to a low level of abstraction, it might be more practical to continue refinement as a standard Jackson problem diagram without using goal modeling, as the link with business strategy has already been achieved. An example of this appears in [49].

Modeling business strategy from text documents is easier said than done. In our approach, we attempted to translate business strategy into the highest-level goals and at the same time determine the context in which that strategy sits. We gradually refined both the goal and context models in parallel. We reached the diagram represented in Fig. 7 through an iterative process of refinement consisting of analysis, modeling, and evaluation. We came to increasingly higher degrees of understanding of the SEJ business case and related business strategy via this iterative process. By ensuring that lower-level objectives and goals in RB and RC support strategic objectives in RA in Fig. 7, we are able to have a high degree of confidence that the model is a reasonable reflection of how SEJ's IT enables its business strategy. We were able to cross-validate this understanding based on documentation in the case study literature [12, 31–33, 35], as well as with a documented interview with SEJ's CEO [34]. In a live project, we would naturally seek validation of the model from executives, managers, or any other appropriate stakeholder.

6 Conclusion

In requirements engineering for e-business systems, business strategy is a driving concern. Requirements engineering research has yet to address this. In this paper, we present a framework for integrating a business strategy dimension. Our framework combines recognized requirements engineering techniques. Problem diagrams provide context for the business model and can be decomposed down to system context or the *machine*. Coupled with this, goal modeling captures the requirements that fit the problem context from the level of business strategy, with its associated objectives, activities, and business processes, down to system requirements. Each projected sublevel of the goal hierarchy in itself represents the requirements set for the context at that level in the projection. When appropriate, we use business process models to describe the optative and indicative properties of the e-business system combined.

Jackson describes a requirement as "the effects in the problem domain that your customer wants the machine to guarantee" [29]. Organizations engaging in e-business rely on their systems to enable their strategy and gain business advantage. It is thus at the level of strategy that companies like SEJ bound the requirements problem for their e-business systems. While we do not propose that requirements engineers make business strategy, they can contribute to achievement of business advantage by ensuring that IT systems requirements are aligned with, provide support for, and enable business strategy.

While the approach we propose is based on research that is still in its early stages, the integration of the

Problem Frames approach, goal-oriented modeling techniques, and BPM may offer promise as a requirements engineering tool for e-business systems. We are continuing our research by applying the approach to rich sets of data on past projects of Australian companies, and evaluating the results. Ultimately, we aim to use action research to apply the approach to a live project within an organization.

Also, while we have developed our approach in the context of e-business systems development, we recognize that e-business systems are not the only type of IT system in which business strategy is a driving concern. In the future, we hope to apply the requirements engineering approach we have developed to other types of systems.

References

1. Stratopoulos T, Dehning B (2000) Does successful investment in information technology solve the productivity paradox? Inform Manage 38:103–117
2. Andersen TJ (2001) Information technology, strategic decision making approaches and organizational performance in different industrial settings. J Strateg Inform Systems 10:101–119
3. Bharadwaj A (2000) A resource-based perspective on information technology capability and firm performance: an empirical investigation. MIS Quart 24:169–96
4. Feeney D, Ives B. (1990) In search of sustainability: reaping long-term advantage from investment in information technology. J Manage Inform Systems 7:27–46
5. Kosynski B, McFarlan W (1990) Information partnerships—shared data, shared scales. Harvard Bus Rev 52:90–102
6. Mata F, Fuerst W, Barney J (1995) Information technology and sustainable competitive advantage: a resource-based analysis. MIS Quart 19:387–401
7. McFarlan FW (1984) Information technology changes the way you compete. Harvard Bus Rev 62:98–103
8. Porter M, Millar V (1985) How information gives you competitive advantage. Harvard Bus Rev 63:149–160
9. Dehning B, Stratopoulos T (2003) Determinants of a sustainable competitive advantage due to an IT-enabled strategy. J Strateg Inform Systems 12:7–28
10. Sambamurthy V (2000) Business strategy in hypercompetitive environments: rethinking the logic of it differentiation. In: Zmud RW (eds) Framing the domains of IT management. Pinnaflex Educational Resources, Cincinnati, OH, pp 245–61
11. McKeen JD, Smith H (2003) Making IT happen: critical issues in IT management. Wiley, Chichester
12. Weill P, Vitale M (2001) Place to space: moving to ebusiness models. Harvard Business School Publishing Corporation, Boston
13. Oliver RW (2001) What is strategy, anyway? J Bus Strateg, pp 7–10
14. Porter M (1996) What is strategy? Harvard Bus Rev 74:61–78
15. Yu E (1993) Modeling organizations for information systems requirements engineering. In: Presented at IEEE international symposium on requirements engineering
16. Giunchiglia F, Mylopoulos J, Perini A (2002) The Tropos software development methodology: processes, models and diagrams. Agent-Oriented Softw Eng Iii 2585:162–173
17. Gordijn J, Akkermans J (2003) Value-based requirements engineering: exploring innovative ecommerce ideas. Requirement Eng J 8:114–135
18. Loucopoulos P (2001) The S3 (strategy-service-support) framework for business process modelling. In: Presented at 3rd international conference on enterprise information systems (ICEIS 2001), Setubal, Portugal
19. Loucopoulos P, Kavakli E (1995) Enterprise modeling and the teleological approach to requirements engineering. Int J Cooperat Inform Systems 4:45–79
20. Bubenko JA, Rolland C, Loucopoulos P, DeAntonellis (1994) Faciliating fuzzy to formal requirements modelling. In: Presented at first international conference on requirements engineering, Colorado Springs
21. Rolland C, Souveyet C, Ben Achour C (1998) Guiding goal modeling using scenarios. IEEE Trans Softw Eng 24:1055–1071
22. Feather MS, Cornford SL, Dunphy J (2002) A risk-centric model for value maximization. In: Presented at 4th international workshop on economics-driven software research, Orlando
23. Feather MS, Cornford SL, Hicks KA (2002) Descoping. In: Presented at 27th NASA/IEEE software engineering workshop, Greenbelt
24. Kiper JD, Feather MS (2005) A risk-based approach to strategic decision-making for software development. In: Presented at 38th annual Hawaii international conference on system sciences (HICSS'05), Big Island
25. Giorgini P, Mylopoulos J, Pistore M (2004) The Tropos methdology: an overview. In: Bergenti F, Gleizes M-P, Zambonelli F (eds) Methodologies and software engineering for agent systems: the agent-oriented software engineering handbook. Kluwer, Boston, pp. xxx, 505 p
26. Garzetti M, Giorgini P, Mylopoulos J, Sannicolo F (2002) Applying Tropos methdology to a real case study. In: Presented at joint workshop from objects to agents: from information to knowledge (WOA 2002), Milan, Italy
27. Bleistein SJ, Cox K, Verner J (2004) RE approach for e-business advantage. In: Presented at 10th anniversary international workshop on requirements engineering: foundation of software quality (REFSQ), Riga, Latvia
28. Bleistein SJ, Cox K, Verner J (2004) Requirements engineering for e-business systems: integrating Jackson problem diagrams with goal modeling and BPM. In: Presented at 11th Asia Pacific software engineering conference, Busan, Korea
29. Jackson M (2001) Problem frames: analyzing and structuring software development problem, 1st edn. Addison-Wesley, Reading
30. Feather MS, Fickas S, Finkelstein A, van Lamsweerde A (1997) Requirements and specification exemplars. Automat Softw Eng 4:419–438
31. Bensaou M (1997) Seven-Eleven Japan: managing a networked organization. INSEAD Euro-Asia Centre, Case study
32. Whang S, Koshijima C, Saito H, Ueda T, Horne SV (1997) Seven Eleven Japan (GS18). Stanford University Graduate School of Business
33. Rapp WV (2002) Retailing: Ito-Yokado Seven-Eleven Japan. In: Information technology strategies: how leading firms use IT to gain an advantage. Oxford University Press, New York, pp. 163–186
34. Makino N, Suzuki T (1997) Convenience stores and the information revolution. Japan Echo 44:44–49
35. Kunitomo R (1997) Seven-Eleven is revolutionising grocery distribution in Japan. Long Range Plann 30:887–889
36. The Economist Newspaper Limited (2001) Over the counter e-commerce; e-strategy brief: Seven-Eleven; Japan's Seven-Eleven. Economist (US), vol 359, pp 77
37. Lowe D (2003) Web system requirements: an overview. Requirements Eng J 8:102–113
38. Overmeyer S (2000) What's different about requirements engineering for web sites. Requirements Eng J 5:62–65
39. Standing C (2001) Methodologies for developing web applications. Inform Softw Technol 44:151–159
40. Zowghi D, Gervasi V (2001) Why is RE for web-based software development easier. In: Presented at seventh international workshop on requirements engineering: foundation for software quality (REFSQ'01), Interlaken, Switzerland
41. Hahn J, Kauffman RJ, Park J (2002) Designing for ROI: toward a value-driven discipline for e-commerce systems design, In: Presented at 35th Hawaii international conference on system sciences, Hawaii

42. Castro J, Kolp M, Mylopoulos J (2002) Towards requirements-driven information systems engineering: the Tropos project. Inform Syst J 27:365–389
43. Gamma E, Helm R, Johnson R, Vlissides J (1995) Design patterns: elements of reusable object-oriented software. First edn. Addison-Wesley, Reading
44. Hall J, Jackson M, Nuseibeh B, Rapanotti L (2002) Relating software requirements architectures using problem frames. In: Presented at RE'02 – 10th international conference on requirements engineering. Essen, Germany
45. I. Bray (2002) An introduction to requirements engineering, 1st edn. Pearson Addison Wesley, Reading
46. Kovitz BL (1999) Practical software requirements: a manual of content and style. Manning, Greenwich
47. Cox K, Phalp K (2003) From process model to problem frame. In: Presented at 9th international workshop on requirements engineering: foundation for software quality - REFSQ'03, Velden, Austria
48. Bleistein S, Cox K, Verner J (2004) Problem frames approach for e-business systems. In: Presented at the 1st international workshop on advances and applications of problem frames (IWAAPF) at the international conference on software engineering (ICSE'04), Edinburgh
49. Bleistein SJ, Cox K, Verner J (2004) Modeling business strategy in e-business systems requirements engineering. In: Presented at fifth international workshop on conceptual modeling approaches for e-business (eCOMO 2004), Shanghai, China
50. Bleistein SJ, Cox K, Verner J (2005) Strategic alignment in requirements analysis for organizational IT: an integrated approach. In: Presented at 20th ACM symposium on applied computing (SAC'05), Santa Fe, NM, USA
51. Bleistein SJ, Cox K, Verner J (2005) Validating strategic alignment of organizational IT requirements using goal modeling and problem diagrams. J Softw Systems (in press)
52. Cox K, Bleistein SJ, Verner J (2004) Connecting role activity diagrams to the problem frames approach. In: Presented at 9th Australian workshop on requirements engineering (AWRE'04), Adelaide, Australia
53. Cox K, Phalp K, Bleistein SJ, Verner J (2005) Deriving requirements from process models via the problem frames approach. Inform Softw Technol 47:319–337
54. Cox K, Hall J, Rapanotti L (2005) Editorial: a roadmap of problem frames research. J Inform Softw Technol (in press)
55. van Lamsweerde A (2001) Goal-oriented requirements engineering: a guided tour. In: Presented at 5th IEEE international symposium on requirements engineering, Toronto
56. Gross D, Yu E (2001) From non-functional requirements to design through patterns. Requirements Eng J 6:18–36
57. Yu E, Liu L (2001) Modelling strategic actor relationships to support intellectual property management. In: Presented at 20th international conference on conceptual modelling, ER-2001, Yokohama, Japan
58. Liu L, Yu E (2001) From requirements to architectural design - using goals and scenarios. In: Presented at ICSE-2001 (STRAW 2001), Toronto, Canada
59. Anton AI, Potts C (1998) The use of goals to surface requirements for evolving systems. In: Presented at ICSE-98: 20th international conference on software engineering, Kyoto
60. Kolber AB, Estep C, Hay D, Struck D, Lam G, Healy J, Hall J, Zachman JA, Healy K, Eulenberg M, Fishman N, Ross R, Moriarty T, Selkow W (2000) Organizing business plans: the standard model for business rule motivation. The Business Rule Group November 15
61. Bleistein SJ, Aurum A, Cox K, Ray P (2004) Strategy-oriented alignment in requirements engineering: linking business strategy to requirements of e-business systems using the SOARE approach. J Res Pract Inform Technol 36:259–276
62. Chung LE, Nixon B, Yu E, Mylopoulos J (1999) Non-functional requirements in software engineering, vol. 5, 1st edn. Kluwer, Dordrecht
63. Zave P, Jackson M (1997) Four dark corners of requirements engineering. ACM Trans Softw Eng Methodol 6:1–30
64. Hammer M, Champy J (1993) Reengineering the corporation: a manifesto for business revolution, 1st edn. HarperBusiness, New York
65. DeMarco T (1979) Structured analysis and system specification. Prentice-Hall, Englewood Cliffs
66. Yourdon E (1989) Modern structured analysis. Yourdon Press, Englewood Cliffs
67. Jackson MJ (1995) Software requirements and specifications: a lexicon of practice, principles, and prejudices. ACM Press; Addison-Wesley, New York; Reading
68. Jacobson I, Ericsson M, Jacobson A (1995) The object advantage: business process reengineering with object technology. Addison-Wesley, Wokingham
69. Arlow J (1998) Use cases, UML, visual modelling, and the trivialisation of business requirements. Requirements Eng J 3:150–152
70. Buhr RJA (1998) Use case maps as architectural entities for complex systems. IEEE Trans Softw Eng 24:1131–1155
71. Buhr RJA, Casselman RSO (1996) Use case maps for object-oriented systems. Prentice Hall, Upper Saddle River
72. Eriksson HE., Penker M, Lyons B, Fado D (2004) UML 2 toolkit. Wiley, Indianapolis
73. Ould MA (1995) Business processes: modelling and analysis for re-engineering and improvement. Wiley, Chichester
74. University of Toronto (2003) Goal-oriented requirements language, University of Toronto
75. Yu E (1997) Towards modelling and reasoning support for early-phase requirements engineering. In; Presented at 3rd IEEE international symposium on requirements engineering (RE'97), Washington