B-SCP: a requirements analysis framework for validating strategic alignment of organizational IT based on strategy, context, and process

Steven J. Bleistein[1 2], Karl Cox[1 2], June Verner[1 2], and Keith T. Phalp[3]

[1]Empirical Software Engineering Research Program,
National ICT Australia, Sydney 1430, Australia
[2]School of Computer Science and Engineering,
University of New South Wales, Sydney 2052, Australia
[3]Empirical Software Engineering Research Group,
Bournemouth University, United Kingdom
{Steven.Bleistein, Karl.Cox, June.Verner}@NICTA.com.au, kphalp@bmth.ac.uk

WORKING PAPER

UNSW-CSE-TR-0517
September 17, 2005

THE UNIVERSITY OF
NEW SOUTH WALES

The School of Computer Science & Engineering
The University of New South Wales
UNSW SYDNEY 2052

Abstract

Ensuring that organizational IT is in alignment with and provides support for an organization's business strategy is critical to business success. Despite this, business strategy and strategic alignment issues are all but ignored in the requirements engineering research literature. We present B-SCP, a requirements engineering framework for organizational IT that directly addresses an organization's business strategy and the alignment of IT requirements with that strategy. B-SCP integrates the three themes of *strategy, context,* and *process* using a requirements engineering notation for each theme. We demonstrate a means of cross-referencing and integrating the notations with each other, enabling explicit traceability between business processes and business strategy. In addition, we show a means of defining requirements problem scope by applying a business modeling framework as a Jackson problem diagram. Our approach is illustrated via application to an exemplar. The case example demonstrates the feasibility of B-SCP, and we present a comparison with other approaches.

Keywords: requirements engineering, strategic alignment, business strategy, business modeling, goal modeling, business process modeling, Jackson Problem Frames.

1 Introduction

Strategic alignment of IT exists when a business organization's goals, activities, and processes are in harmony with the information systems that support them [1]. Effective strategic alignment positively influences IT effectiveness [2-4] and leads to superior business performance [5-7]. It is thus not surprising that CIOs and IT executives consistently rank alignment of IT with business strategy as a top priority [8-12]. Despite this, issues of business strategy and strategic alignment are all but ignored in requirements engineering research literature.

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 thus includes both the rationale for and the means by which a business organization competes with industry rivals [14, 15]. Various aspects of business analysis have been addressed in the requirements engineering research literature, including organizational structure and dependency relationships among actors in a system [16, 17], economic and business value analysis [18], organizational goal-driven business process modeling [19, 20], and elicitation of organizational goals from which to derive requirements [21]. Other research takes an enterprise modeling view in requirements analysis [22-24]. However, none of the requirements engineering approaches cited above include explicit analysis of an organization's competitive business strategy or strategic alignment.

To address alignment of requirements with competitive business strategy, Bleistein *et al.* present a requirements analysis approach for verification and validation of requirements in terms of alignment with and support for business strategy [25, 26]. This approach combines use of business strategy analytical tools and requirements engineering techniques [25, 26]. Goal modeling is used to represent business strategy as requirements, and Jackson context diagrams [27] to represent business and system model context. The strategy and context parts are integrated using a problem diagram framework [27]. Strategy is first elicited using *VMOST* [28], an organizational alignment analysis technique. Then, an *i** goal model [16] is constructed using goal modeling rules for organizational motivation proposed by the Business Rules group [29].

Building upon [25, 26], we thus propose *B-SCP,* a requirements analysis framework based upon the three themes of *business strategy*, *context*, and *process,* whose purpose is to enable verification and validation of requirements in terms of alignment with and support for business strategy, and the business processes that support that strategy. For each of the themes a requirements analysis technique is used, *i** goal modeling [16] for *strategy*, Jackson context diagrams (part of Jackson problem diagrams) [27] for *context*, and role-activity diagrams (RADs) [30] for *process*. A means of connecting each technique with the other two in order to form an integrated model is demonstrated. Feasibility of B-SCP is demonstrated via a case study.

In this paper we extend and refine previous work [25, 26, 31, 32] in two ways: (1) we demonstrate how to scope the context of a strategic organizational IT requirements problem using a strategic business modeling framework proposed in [33]; and (2) we introducing a means of explicitly cross-referencing business processes with an organizational goal model and context diagrams in order validate alignment of requirements with business strategy and the business processes that support that business strategy.

The rest of this paper is organized as follows. Section 2 provides an overview of the requirements analysis techniques used in B-SCP. Section 3 presents the B-SCP framework. Section 4 describes how to build a requirements analysis model based on B-SCP. Section 5 presents a proof-of-concept example applying B-SCP using the case of Seven-Eleven Japan. Section 6 discusses and evaluates B-SCP. Section 7 concludes the paper.

2 Overview of Goal Modeling, Jackson Problem Diagrams, and role activity diagrams

In this section, we provide a brief overview of these techniques and how they are used in B-SCP. Section 2.1 discusses organizational goal modeling, Section 2.2 discusses Jackson context diagrams and problem diagrams, and Section 2.3 discusses role activity diagrams (RADs).

2.1 Organizational Goal Modeling Notation

In requirements engineering research, there are several goal modeling notations used to represent an organization's intent, and to link these to functional and non-functional requirements [16, 34-37]. Typically these goal modeling notations contain a concrete goal, meaning a goal whose achievement or *satisfaction* can be quantifiably measured, and an abstract goal, whose achievement is not directly, quantifiably measurable. There are also entities used to represent activities, such as tasks and processes. Tasks may be decomposed into sub tasks via *decomposition links*, and contribute to achievement of a goal via a *means-end contribution link*. Both goals and tasks are decomposed, or *refined*, into sub goals according to formal refinement patterns based on temporal logic, such as AND/OR trees of goal assertions [38].

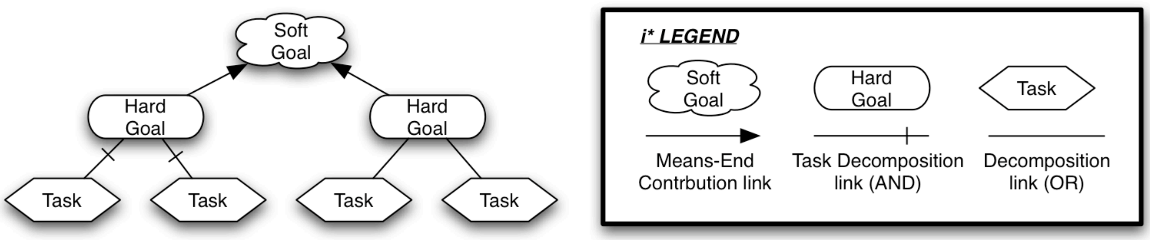

Figure 1. Simple Example of a Goal Model Structure

Figure 1 presents a simple illustration of a goal model structure using *i** notation [16], in which *soft goal* refers to an abstract goal and *hard goal* refers to a concrete goal. As shown in Figure 1, two hard goals contribute to the achievement of a soft goal. Satisfaction of the hard goal on the left is achieved by two tasks as indicated by the AND decomposition link, whereas the hard goal on the left can be satisfied by either of the tasks, indicated by the OR decomposition link.

2.2 Overview of Problem Diagrams

Problem Frames are used to capture, structure and classify recurring software development problems with a problem diagram framework [27, 39]. Problem diagrams provide an analytical framework for requirements based on real-world physical entities and their observable interactions and behaviors. Because of their focus on describing software problems in real-world, physicals terms, we make extensive use of problem diagrams in our requirements analysis approach for strategic alignment. A problem diagram describes properties of a software problem according to two "moods" [27, 39] to represent the way the world is now and the way we would like the world to be. *Indicative* mood represents everything in the problem that is given and will remain unaffected by the software system, including physical domain entities such as people, organizations, departments and devices, and their shared phenomena, such as activities, processes, events, states, commands, and information. *Optative* mood represents the way we would like everything to be, given the construction of the software system, and thus represents the requirements [27]. Requirements include business goals, objectives, processes, and all other business and system requirements whose purpose is to alter the 'As Is' view of the world in some way. As a requirement can only be understood in the context in which it occurs, a problem diagram thus consists of two major components: a requirements part and a domain context diagram [27]. Context diagrams contain real-world *physical* domain entities called *domains of interest*. The phenomena that two or more domains of interest share is indicated by an *interface* connecting the domains of interest. *Shared phenomena* consist of observable behavioral phenomena that occur between entities in a context diagram. Context diagrams always contain one special domain of interest, the *machine*, which is a general-purpose computer that is programmed. The requirements part of a problem diagram describes the effects in the real world that the *machine* should guarantee.

Figure 2 illustrates some essential elements of a problem diagram. The requirements are enclosed in a dotted-line oval. The context diagram contains several domains of interest and the *machine*. For most software problems there will be multiple requirements ovals, domain context diagrams, and numerous domains of interest.

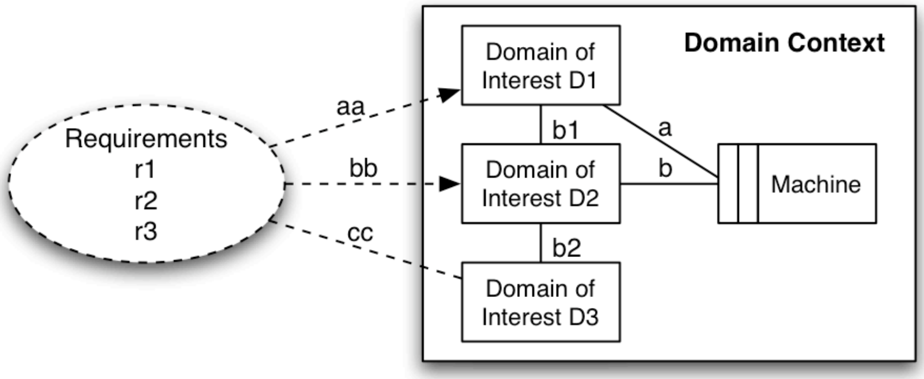

Figure 2. Anatomy of a Problem Diagram

In the context diagram, the *machine* and three domains of interest, *D1, D2,* and *D3*, are interconnected with solid line *interfaces*, labeled *a, b, b1,* and *b2,* representing shared phenomena. Shared phenomena between domains are described through the following syntax:

b : Domain of Interest D2 ! { *Shared phenomenon description* }

meaning "at *b*, Domain of Interest *D2* is responsible '!' for the *shared phenomenon description*."

Requirements either *reference* or *constrain* domains of interest in the context diagram. A *requirement constraint* indicates that "the *machine* must ensure that the state or behavior of that domain satisfies the requirement" (p. 370) [27]. A *requirement reference* indicates the domain provides a description of phenomena in the domain context. Requirements *constraints* and *references* are indicated by dotted lines from the requirements to domains of interest in the context diagram. An arrowhead indicates that the domain is constrained by the requirement, such as constraints *aa* and *bb* on domains of interest *D1* and *D2* respectively. A requirement reference, with no arrowhead such as reference *cc* on *D3*, indicates that the requirement *refers* to some phenomena in that domain. Constraints and references are described using syntax similar to that of shared phenomena:

bb : Domain of Interest D2 ! { Requirement *r2* }

meaning "at *bb,* Requirement *r2*, for which Domain of Interest *D2* is responsible, constrains Domain of Interest *D2*."

Domains of interest may appear a number of times in the problem diagrams and problem frames through the principle of *projection*. *Projection* refers to the ability to describe domain context according to various viewpoints, levels of abstraction, and degree of detail [27]. A requirement might concern only certain phenomena or certain behavior of a domain, given the particular sub-problem addressed. In a different projection, the other domain phenomena might be of interest to the requirement for that particular problem. As projection is also a means of decomposing domain context into increasingly finer degrees of detail, projection is particularly useful when managing requirements at multiple levels of abstraction.

2.3 Role Activity Diagrams (RADs)

A Role Activity Diagram (RAD) [30] is a process modeling notation, widely used and well-regarded in industry. A RAD is used to describe business processes that can involve actions and interactions among roles. Roles can be humans as well as software and hardware systems. A RAD provides an excellent means of describing dependencies between roles in organizations that work discretely and in unison to achieve a goal. A RAD has various components, the most common of which are illustrated in Figure 3.

All roles start in an initial state. For example, *Colleague A* starts in some initial state and then an external event occurs, in this case a project has been started. *Colleague A* proceeds with an *action*, 'do work.' Note that an *action* is independent of other roles. On completion of work, *Colleague A* would be said to have moved to a new state of work completed. Although *state descriptions* are often omitted in RADs, a formal view would be that the event, and action of role A, has a pre-state of 'initial' and post state of 'activity completed'. Where it is necessary to show an activity has been completed within a process, it is shown explicitly in the RAD using a *state description*.

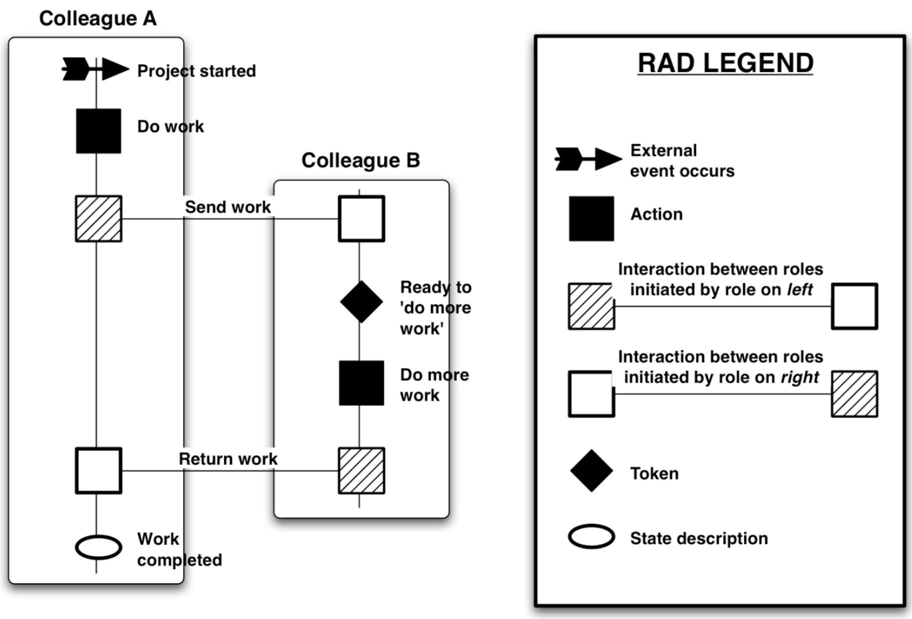

Figure 3. Some elements of a Role Activity Diagram

Continuing with the example in Figure 3, *Colleague A* sends work to *Colleague B*. This is a shared event, or *interaction*. Although the mechanism of delegation is typically immaterial, the result is that both roles involved move to the state of work delegated. While there is no sender and receiver as such, *Colleague A* is said to *initiate* whereas *Colleague B* is passive in this interaction. The *initiator* of an interaction is denoted by the hatched box and the *recipient* by the clear box. *Colleague B* is then in a state to independently 'do more work,' which is denoted by a *token,* indicating a point at which a condition is satisfied in order to continue. *Colleague B* then returns work to *Colleague A*. The oval at the end of Role A 'Work completed' indicates a *state description* has been reached, and this can equate to the realization of a goal or requirement.

Thus, a RAD captures activities, such as actions a role takes on its own, and interactions, in which multiple roles participate, that when combined, represent a process within a department, across an organization, or out into the marketplace. In addition, each process achieves a number business goal or requirements, which should be made explicit [30].

3 The B-SCP Framework for Strategic Alignment

Haglind and Cheong [40], encountering a number of obstacles in a case study of an industrial IT project, proposed a modeling framework for strategic alignment of enterprise software architectures according to three themes, *strategic content*, *business context,* and *business process,* originally proposed by Walsham [41]. However, Haglind and Cheong provide no means of operationalization of the framework [40].

We therefore propose *business strategy, context, and process (B-SCP),* a requirements engineering framework based on the three themes presented in [40, 41], as summarized in Table 1. The *strategy* theme refers to how an organization intends to use IT to compete within its market or industry. The *context* theme refers to the business and organizational environment in which an organization operates. It includes both internal and external organizational structures. The *process* theme refers to business activities, their support systems and other organizational resources, roles, entities, and the interactions among all of these.

Table 1. B-SCP Requirements Engineering Operationalization

Themes	Description	RE Notations and Techniques
Strategy	How the organization intends to use IT to compete within its market or industry	*i** Goal Model
Context	The business and organizational environment in which the organization operates	Jackson Context Diagram
Process	Business activities, their support systems and other organizational resources, roles, entities, and the interactions among all of these	Role Activity Diagram (RAD)

The requirements engineering notations and techniques used in B-SCP may be used in isolation, but using any one of these requirements techniques alone results in neglecting critical aspects of IT requirements analysis for a business organization. Organizational goals describe only objectives and intention, but say little about the context in which these occur. It is important to verify that goals are in alignment with the organizational context in which they occur, and thus the relationship between the goal model and the context diagram must be clear. While we could make use of any of a number requirements engineering goal modeling notations [16, 34-37], we use *i** [16] for convenience, as it is a widely recognized notation in the requirements research community. However, we only use modeling entities of *soft goals, hard goals, tasks,* and their contribution relationships from the *i** notation. We avoid using *resources*, *agents,* and *roles* [16], as we prefer to represent these entities as part of *context* as this reduces goal model clutter and avoids obscuring organizational context modeling, which we discuss later in Section 6.1. In addition, even though a number of goal modeling notations [16, 34-37] include entities to represent processes or tasks, these only describe the *activity* aspect of processes. They lack details of interaction among roles, the order in which activities are carried out, and concurrent and alternative process paths. Process models describe these aspects in detail and the goal that the process achieves. However, process models describe neither how the goals they achieve fit into the greater objectives of the organization or its business strategy that might be better understood as a goal model. Processes also do not describe how roles and resources are positioned in an organizational context that might be better understood in a context diagram. In order to verify alignment of a process, it is critical to understand both the larger organizational context in which a process occurs [30], and the manner in which a process's output, i.e. its goal [30], contributes to the strategic objectives of the organization [42]. Thus, to truly understand the IT requirements of a business organization in a comprehensive manner, it is necessary to use goal modeling, context diagrams, and process models together.

At the same time, each requirements notation used in B-SCP must have a means for connecting and integrating with the other two notations, as this is critical if the integrated framework is to be used to verify alignment. We therefore use a number of techniques. We integrate an *i** goal model with Jackson context diagrams using a problem diagram framework by treating goals as the requirements part of a problem diagram, a technique previously demonstrated in [25, 26]. We integrate Jackson context diagrams with RADs by maintaining equivalence between the roles in RADs and the domains of interest in context diagrams, which implies equivalence between shared phenomena and interactions, as demonstrated in [43]. We cross-reference activities and state descriptions in RADs with *task, soft goal,* and *hard goal* entities in *i**. Also, as the output of a process should be the achievement a goal [30], processes are linked to goals in the goal model. In this way, a RAD is connected to a goal model at multiple points of reference.

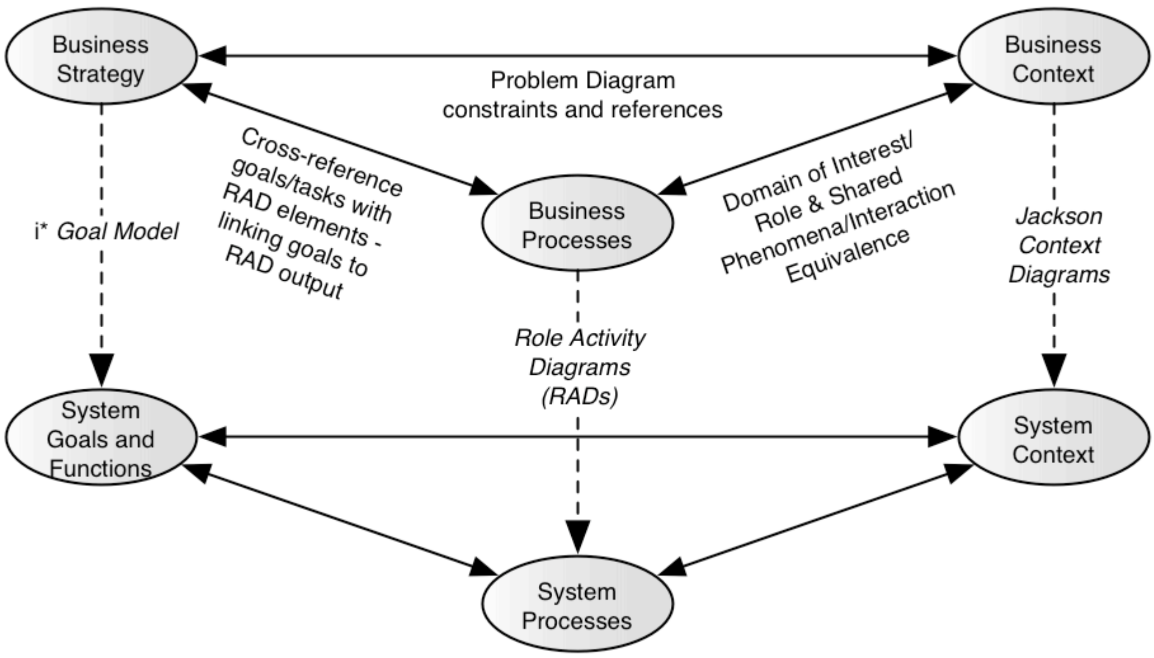

Figure 4. The B-SCP Framework

In addition, in order to validate lower-level requirements against higher-level strategic objectives, a means of top-down refinement and bottom-up traceability is critical. This is accomplished using the goal model, as goals can be refined from high-level strategic concerns to low-level technical ones [44], and there exist standard reasoning approaches for refining goals into sub goals [34, 38]. As goals are refined, they refer to context at lower levels of abstraction. Jackson context diagrams accommodate context decomposition using *projection* [27]. Similarly, RADs are capable of representing processes at multiple levels of abstraction and detail by describing or hiding process detail using respectively "black box" or "white box" representation of *roles* [30].

Figure 4 illustrates the B-SCP framework in the shape of a prism standing on end. The top-level business model is represented by the three themes of *business strategy, business context,* and *business process*. *Strategy* and *context* are integrated via a problem diagram framework using requirements constraints and references. *Context* and *process* are connected via equivalence between domains of interest and roles, and between shared phenomena and interactions. *Strategy* and *process* are connected via cross-referencing goals and tasks with elements in the RAD, in addition to linking of goals to RAD outputs. The dashed lines depict refinement or decomposition/projection from the business strategy level down to the system level of the requirements model. Strategy is refined down to system goals and requirements via an *i** goal model. System goals and requirements can also be validated against the objectives of business strategy via upward, traceable, contribution links in the goal model. Business context can be decomposed down to system context using *domain decomposition* and *projection* [27]. RADs, with their remarkable flexibility, can model both business and machine processes, at multiple levels of abstraction and detail.

4 Building a Requirements Model using the B-SCP Framework

In this section, we present the details of building a model for requirements analysis based on the B-SCP framework. Section 4.1 presents the *strategy* theme, represented as a goal model. Section 4.2 presents the *context* theme, represented by Jackson context diagrams and illustrates the integration of *strategy* and *context* using a problem diagram framework. Section 4.3 discusses refining strategy and context in parallel from business model level to system requirements level using a progression of problem diagrams. Section 4.4 demonstrates linking of process models to goal models and context diagrams. Section 4.5 presents a summary of B-SCP.

4.1 Strategy*: the Goal Model*

A recognized difficulty in goal modeling in requirements engineering is discovering what an organization's goals are and where they fit into the overall structure of a goal model [35]. Attempting to model an organization's business strategy, in which *soft* goals may be the norm rather than the exception and goal refinement is many layers deep, poses additional challenges for a requirements engineering goal modeler. Requirements engineers using requirements goal modeling notations and techniques whose goal types are limited to either *soft* and *hard*, whose goal relationships are based on simple, temporal logic formalisms, and which offer only one type of *task*, are ill equipped for the job of eliciting and modeling an organization's business strategy and linking that strategy to system requirements.

Some business analysis approaches, such as that of the Business Rules Group [29], suggest that not all goals are the same when used to model business motivation or business strategy. The Business Rules Group's Model for Organizational Motivation (BRG-Model) [29] distinguishes goal types as *vision, goal,* and *objective*. *Task* types similarly possess qualities that provide an understanding of the type of goal to which it contributes, and include *mission, strategy,* and *tactic*. The definitions of the BRG-Model goal and task types are summarized in Table 2.

The BRG-Model describes rules by which goal and task entities must relate to each other in a goal model for business strategy [29], as illustrated in Figure 5. A BRG-Model goal thus not only expresses *hardness* or *softness* but also possesses an associated quality according to its type that provides an understanding of where it is situated in an overall model of strategy and how it relates to other goals. BRG-Model tasks similarly describe not only activities and processes, but indicate to which type of goal in the strategic hierarchy the process or activity is intended to achieve. As each goal and task type has a requirements engineering goal modeling equivalent, albeit a less descriptive one, it is possible to use requirements engineering goal modeling notations to model an organization's business strategy according to the BRG-Model framework and associated rules. A recasting of the BRG-Model Framework in Figure 5 into *i** notation is illustrated in Figure 6.

Table 2. Organizational Business Strategy Modeling - Goals and Tasks

Goal Types	RE Equivalent	Definition	Activity Types	RE Equivalent	Definition
Vision	*Soft Goal*	*An end-state toward which the organization strives*	***Mission***	*Task*	*The primary activity of the organization that achieves the vision*
Goal	*Soft Goal*	*An abstract statement of intent whose achievement supports the vision*	***Strategy***	*Task*	*A long-term activity designed to achieve a goal*
Objective	*Hard Goal*	*A specific and measurable statement of intent whose achievement supports a goal*	***Tactic***	*Task*	*A short-term action designed to achieve an objective*

While we could make use of any of these goal modeling approach in B-SCP, we use *i** [16] for convenience, as it is a widely recognized notation in the requirements research community. However, because the B-SCP *strategy* theme deals only with goals and tasks, we only use modeling entities of *soft goals, hard goals, tasks,* and their contribution relationships from the *i** notation. We avoid using *resources*, *agents,* and *roles* [16], as we prefer to represent these entities as part of *context*.

Figure 5. BRG-Model, adapted from [29]

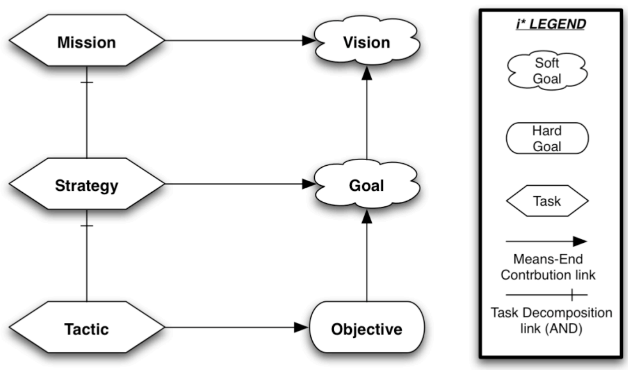

Figure 6. BRG-Model operationalized

Using the BRG-Model definitions enables a goal modeler to take advantage of business strategy analysis tools that use the same analysis concepts, i.e., *vison, mission, goal, strategy, objective,* and *tactic*, for the elicitation of the organizational goals that comprise business strategy. One such tool is VMOST (standing for vision, mission, objective, strategy, tactic [28]), which provides a means for deconstructing business strategy by answering a number of *key questions*. Following VMOST analysis, the BRG-Model provides a framework to guide a requirements engineer in constructing a goal model. We do not illustrate the process of using VMOST analysis in conjunction with the BRG-Model to develop a requirements engineering goal model of organizational business strategy in this paper, as it is discussed in detail in [25, 26].

4.2 Context*: Jackson Context Diagrams and Integrating* Strategy

Jackson Context Diagrams [27, 39] are a means of scoping the context of a problem, i.e., where the problem is located and what parts of the real world it concerns, answering the question, "What parts of the context are relevant to the problem?" Relevant contextual elements can appear at differing levels of abstraction, making this question more difficult to answer convincingly. The key is to represent what is *useful* and *necessary* to describe the problem being addressed. Problem context may also be decomposed into smaller, sub problem contexts at lower levels of abstraction, and Jackson proposes heuristics for this [27]. However, these heuristics are useful only once the problem scope and context have been determined and the problem is recognized as being close to the machine. Problems whose scope encompasses business strategy, i.e., the type of problem that is the subject of this paper, are not typically close to the machine, implying a need for an alternative approach to identify the problem context from which to begin decomposition into sub problems. For this purpose, we apply Weill and Vitale's business modeling framework [33].

Weill and Vitale define a business model as "a description of the roles and relationships among an organization's consumers, customers, allies, and suppliers that identifies the major flows of product, information, and money, and the major benefits to the participants" [33]. Based on their definition, Weill and Vitale develop a graphical modeling notation whose entities include the *organization of interest, suppliers, allies, customers, consumers,* and the *relationships* among these entities. Each relationship describes *flows* of money, product, and/or information. A number of key business requirements are associated with each graphical model of business context so that the model might be successful. These requirements outline the business strategy aspect of the model. While Weill and Vitale use this definition specifically for e-business models, the definition is general enough to apply to any kind of business in a value network selling any type of product or service. Note that Weill and Vitale also propose *recurring* e-business models [33], which we do not address in this paper. A detailed discussion on leveraging Weill and Vitale's recurring models for e-business in requirements analysis appears in [45].

The Weill and Vitale business model framework thus fits nicely into a problem diagram framework, as it provides the basis for the separation of concerns, between business model context and the requirements of the business strategy. The contextual part of the business model, which Weill and Vitale model graphically, can be modeled as a Jackson context diagram. Physical domains of interest represent business model participants, i.e., the *organization of interest, suppliers, allies, customers,* and *consumers*. The relationships among the participants are indicated as *interfaces,* whose flows of *money, products,* and *information* are described as *shared phenomena*. The *machine* domain of interest can be used to represent Weill and Vitale's IT-enabled *organization of interest.* While Weill and Vitale simply list the strategic requirements associated with each business model, we provide structure to these by representing them in a goal model in the *requirement* part of the problem diagram.

Figure 7 illustrates an example of a Weill and Vitale business model framework as a problem diagram. Please note that this is only a simple example for the purpose of illustration, and does not represent a model for a sophisticated, competitive business strategy. The context model describes a simple model of a wholesale business. The organization of interest as indicated by the double line on the left side of the entity. The wholesale business takes orders from customers and arranges for their manufacture and just-in-time delivery to customers using an allied third-party logistics partner. The *Wholesale Business, Customer, Supplier,* and *Logistics Partner* are described in the context diagram. The shared phenomena are indicated and labeled with single letters, or a letter and number in the case of an interface between domains of interest not serving as the *machine.* The interfaces *a, b, b1, b2,* and *c* in the context diagram are explained in the box below the problem diagram. The requirement part of the problem diagram describes the strategy of the wholesale business in the form of a goal model. Each goal is treated as a discrete requirement and labeled *G1-4,* and *O1-4*. Entities in the goal model are labeled according to a convention first presented in [25] indicating BRG-Model type, in this case *G* for *Goal* and *O* for *Objective*, plus a number. Similarly, *S* is used for *Strategy* and *T* for *Tactic,* although these entity types do not appear in the example in Figure 7. Requirements constraints and references are labeled with double letters *aa, bb,* and *cc* and detailed in the box below the problem diagram. In this way, we can use a problem diagram framework to integrate *strategy*, represented as a goal model, and *context*, represented as a context diagram, according to the business modeling framework suggested by Weill and Vitale [33].

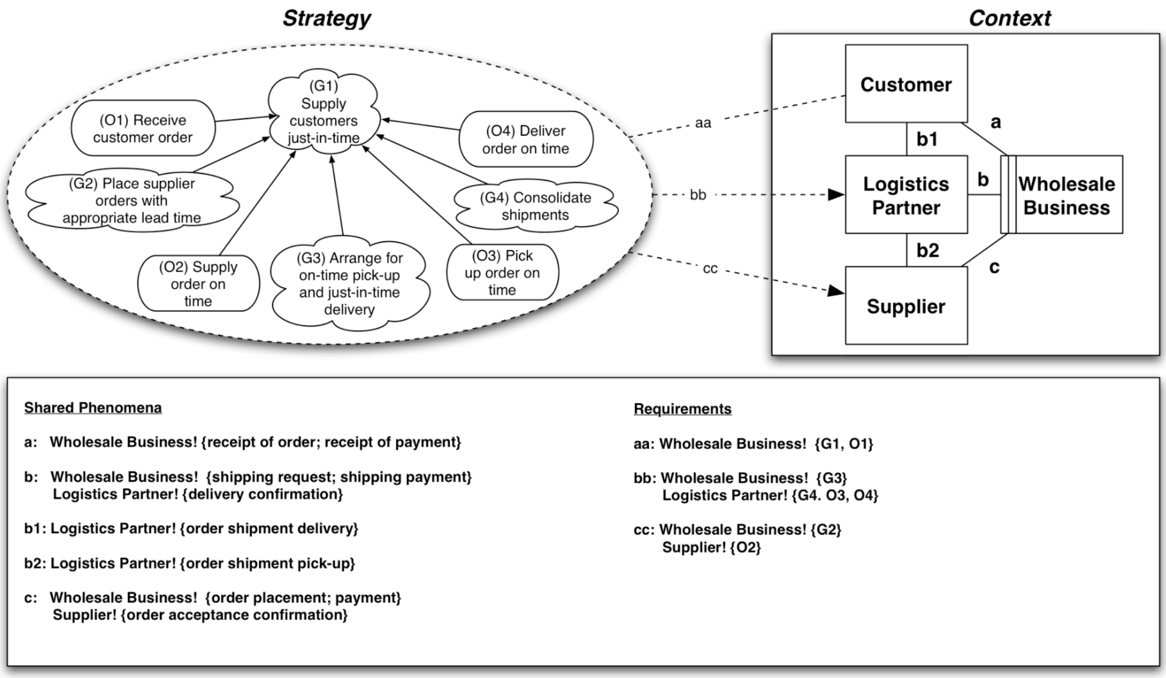

Figure 7. Business Strategy and Context as a Problem Diagram: Wholesale Business Example

To develop the Weill and Vitale business model problem diagram, we propose the following steps:

1) **Identify the business model participants.** Who are the *organization of interest, suppliers, allies, customers,* and *consumers* in the model? These become the *domains of interest* and the *machine* in the context diagram.

2) **Identify the relationships among the participants.** What are the flows of *money, product,* and *information,* and between which participants do these flows occur? The flows represent *shared phenomena,* and the relationships between participants represent *interfaces* between *domains of interest.*

3) **Identify the strategic requirements of the business model and represent these as a goal model.** As mentioned previously, we recommend performing this process of analysis by combining VMOST analysis [28] with goal model construction according to BRG-Model rules [29], which we do not show in this paper but demonstrate in detail in [25, 26].

Identifying the Weill and Vitale business model and representing it as a problem diagram with an integrated goal model is a critical step in requirements analysis using B-SCP. The reason for this is that the top-level problem diagram defines both the scope of the business problem to be solved and the critical strategic, business objectives that are to be met. Without this, understanding the lower-level requirements of the system is murky at best, and strategic alignment will remain elusive. It has been shown that defining project scope not only leads to good requirements [46, 47], but is more importantly critical to organizational IT project success [48].

4.3 Refining Strategy and Context

A problem diagram at the Weill and Vitale business model level, however, is very distant from system-level requirements, and is likely to be too abstract to begin designing and implementing a solution consisting only of hardware, software, data, network resources, and individual people. To refine requirements from a high-level problem diagram down to the machine, the concept of a *progression of problems* discussed in [27] is particularly useful.

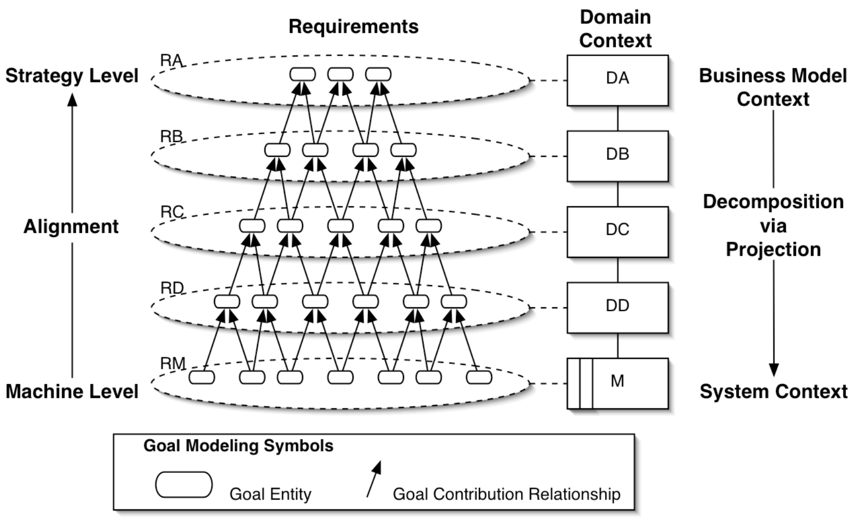

Figure 8. Progression of Problems adapted from [27] p. 103

Figure 8 illustrates a progression of problem diagrams. Requirement ovals *RA, RB, RC, RD,* and *RM* each refer to domain context diagrams *DA, DB, DC, DD,* and *M* respectively. The domain context *DA* represents business context and strategy at the level of a Weill and Vitale business model. Requirement *RA* represents the requirements of business strategy, in the form of a goal model, associated with the Weill and Vitale business model. Through analysis of *DA*, it is possible to decompose the domain context into a more refined context diagram *DB*. Then through an analysis of *DA* and *RA*, it is possible to find a requirement *RB* that refers only to *DB* while satisfying *RA*. Similarly, through analysis of *DB*, it is possible to decompose the domain context into context diagram *DC*. Then through an analysis of *DB* and *RB*, it is possible to find a requirement *RC* that refers only to *DC* while satisfying *RC*, and so on. Through this process of domain context decomposition, analysis, problem projection, and refinement, ultimately the requirement refers just to the *machine*, yielding the system specification [27].

As goals refer to *optative* properties a system is intended to ensure [49], we treat goals as requirements in problem diagrams. Note that in Figure 8 a single, large goal model, used to model the *strategy* theme of the B-SCP framework, is partitioned at multiple levels of refinement by the requirement ovals *RA, RB, RC, RD,* and *RM.* A goal model representation is particularly useful in this way when performing a progression of problem diagram analysis, as the goal model furnishes a means to link one level of refinement of *strategy* with adjacent levels. For example, Figure 8 shows how *RB* contributes upward to *RA* and is refined downward to *RC* via the goal model contribution relationships. Lower-level requirements can thus be validated against higher-level goals, enabling validation of requirements alignment against objectives of business strategy. At the same time, requirements can also be understood within the domain context to which they refer. The problem diagram framework thus also enables validation of requirements alignment within organizational context. Bottom-up traceability of requirements in the goal model in conjunction with verifying requirements within the context in which they occur at appropriate levels of refinement are both essential to validating overall strategic alignment of requirements.

4.4 Linking Process Models to Goal Models and Context Diagrams

A process consists of tasks, activities, and roles according to procedural constraints or rules in order to achieve a desired output or goal [50]. Many companies use process models to describe their activities and systems. There is a close relationship among process models, context diagrams, and goals. Indeed, moving from process models to context diagrams is a recommended approach [51], and as mentioned previously, connecting RADs to Jackson context diagrams and problem frames has been demonstrated in [43]. In addition, Ould recommends performing business process analysis according to the organizational goals that the process is intended to achieve [30], providing a link between business processes described by RADs and the goals appearing in an organizational goal model.

Table 3. Mapping Role Activity Diagram to Goal Models and Context diagrams

RAD	Goal Model ("Closest to" mapping)	Jackson Context Diagram (Equivalence)
Role	---	*Domain of Interest*
Action	*Hard Goal / Soft Goal / Task*	---
Interaction	*Hard Goal / Soft Goal / Task*	*Shared Phenomena / interface*
States	*Hard Goal / Soft Goal*	---

A mapping of the elements in RADs to goal models and context diagrams is described in Table 3. A *role* is equivalent to a *domain of interest* in a Jackson context diagram, but has no equivalent in a goal model consisting of goals and tasks. An *action* maps to a goal or task in a goal model, but has no equivalent in a context diagram. An *interaction* is always between two roles or among several, and thus an *interaction* maps directly to *shared phenomena* as identified by an *interface* in a context diagram. An *interaction* also maps to a goal or task in a goal model. *State descriptions* map to goals in a goal model but not to tasks, and have no equivalent in context diagrams. Between RADs and Jackson context diagrams, there is a one-to-one equivalence between *roles* and *domains of interest*, and between *interactions* and *shared phenomena.* However, the goal model cross-reference does not necessarily correspond in an exact one-to-one manner, as process models describe a degree of detail that is awkward in a goal model [32]. Thus, the goal model cross-reference refers to the goal entity that is *closest* to the activity described in the process model.

For a simple demonstration of mapping a RAD to a goal model and context diagram, let us return to the wholesale business example discussed in Section 4.2. The RAD in Figure 9 describes the processes of the wholesale business, and is linked to the goal model and context diagrams in Figure 7. Please note for each interaction, activity, and goal appearing in the RADs, a cross-reference appears at the beginning of each label in parentheses. Interactions cross-reference both an entity in the goal model and a shared phenomenon in the context diagram, for example *(G2, c)*. The purpose of the cross-reference of an interaction is to enable an analyst to situate the interaction in both the goal model and the context diagram in order to validate organizational alignment between the process model and the goal model, and between the process model and the context diagram. As mentioned above, the shared phenomenon interface label cross-reference represents a one-to-one correspondence. The interaction between roles in a RAD corresponds precisely to shared phenomena between domains of interest in a context diagram.

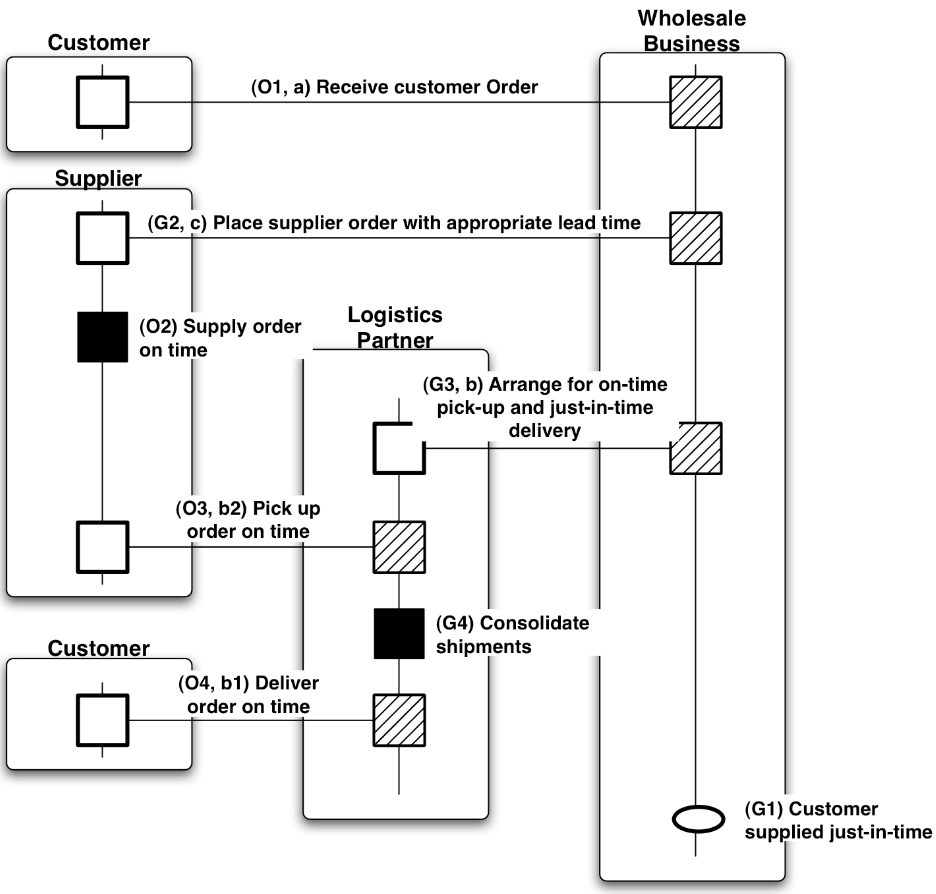

Figure 9. Wholesale Business RAD

The wholesale business RAD supports organizational goal (G1), supply customers just-in-time, in Figure 7. The *Wholesale Business* receives a customer order (O1, a), places an order to the *Supplier* with the appropriate lead time (G2, c), and arranges with the *Logistics Partner* for on-time pick-up and just-in-time delivery (G3, b). The

Supplier in the meantime endeavors to supply the order on time (O2), which the *Logistics Partner* must pick up on time (O3, b2), consolidate with other incoming shipments (G4), and then deliver to the *Customer* just-in-time. At this point, the customer has been supplied just-in-time, achieving G1 for the *Wholesale Business*.

4.5 B-SCP Summary

B-SCP thus provides and integrated framework for *i** goal modeling [16], Jackson context diagrams [27], and RADs [30]. A goal model is integrated with context diagrams via a problem diagram framework [27], in a progression of problems [27] from the strategic-business level problem to the system level problem. The Weill and Vitale business model framework [33] is used to scope the strategic-level business problem diagram, a critical step in B-SCP. RADs are connected to the goal model and context diagrams by cross-referencing according to mapping rules.

5 Case Example: Seven-Eleven Japan

For initial validation of a new technique, such as B-SCP, it should applied to an appropriate requirements engineering exemplar that demonstrates its usefulness, rather than a generic example, such as the elevator problem, the ATM problem, the meeting scheduler, etc., that does not exemplify the type of problem the new technique is intended to solve [52]. We found that the requirements engineering research literature is devoid of well-documented examples of organizational IT that encompass business strategy. To address this problem, we developed a requirements engineering example suited to demonstrating a capability of verifying and validating requirements in terms of alignment with business strategy. We based the example on the rich research literature on the Seven-Eleven Japan case appearing in both management and information systems literature [33, 53-58]. We use the case of Seven-Eleven Japan (SEJ) and its IT system to demonstrate B-SCP. Our objectives in using the SEJ example are to demonstrate (1) validation of system requirements against strategic business objectives via traceable links, and (2) cross-referencing between process models against both a goal model and context diagram as a means of better understanding the processes supporting business strategy. We present a partial view, or a projection [27] of the requirements problem, sufficient to meet these objectives. Section 5.1 presents an overview of the SEJ case. Section 5.2 describes the progression of problem diagrams integrating SEJ's goals and context. Section 5.3 discusses validating alignment of lower-level requirements against SEJ's business strategy, and explores processes that support the strategy.

5.1 SEJ Case Overview

SEJ manages a national franchise of independently-owned convenience stores, whose Chief Executive Officer (CEO) described his vision to create a chain of convenience stores "where you can find a solution for any of your daily life problems" at hours when needed [53]. The CEO's plan was to use IT to help realize his vision, enabling SEJ to leverage information to coordinate a supply chain of business partners to ensure that stores were stocked with precisely the products that consumers want when they want them [53, 56]. These business partners include product *suppliers*, who either make or distribute the products for sale in the stores, *combined delivery centers*, companies with warehouses and fleets of trucks that provide logistics support, and the *franchise stores* themselves, SEJ's direct customers, that sell to the individual end-*consumers* who patron the stores.

Stores in Tokyo, where land is a premium commodity, tend to be very small, and thus have little space to stock inventory. Shelf space must be filled only with products that move quickly, and stock must be replenished frequently. SEJ's business strategy thus focuses heavily on a value proposition to storeowners that addresses these requirements. To this end, SEJ needs to predict with precision what products consumers will demand, when they will demand them, and then deliver inventory just-in-time to meet that demand. This is particularly challenging for perishable goods, such as box lunches and other processed fresh foods as consumers' tastes change daily depending on the weather, holidays, and neighborhood events. Tastes also vary from store-to-store depending on neighborhood demographics. At odds with the need to limit inventory is the need for consumers to find what they want in the store. Should a consumer fail to find the product he is seeking, not only does SEJ lose the opportunity to make a sale, but SEJ has also learned through experience that the consumer often never returns [53].

5.2 SEJ Strategy: Business Model Goals and Context

In this section, we present a progression of problem diagrams integrating the goals and context of the SEJ case in Figure 10. The associated shared phenomena, requirements constraints and references are detailed in Table 4. To derive and construct the goal model in Figure 10, we used the combination of VMOST analysis to elicit goals, and applied the BRG-Model rules to construct the goal model using *i**. We do not describe this process here, as it is discussed in detail in [25, 26]. Note that a number of goal model entities in Figure 10 are shaded and/or outlined in bold. This is to highlight the entities used in demonstrating cross-referencing the process models with the goal model in Section 5.3.2. The reader may find it useful to photocopy Figure 10 and Table 4 for ease of reference.

Table 4. Phenomena of the SEJ Problem Description

INDICATIVE			OPTATIVE		
Interfaces			Requirements		
Domain Context	Responsible Domain of Interest	Shared Phenomena	Req't Set	Responsible Domain of Interest	Requirements
DA	**a:** Franchise Store!	Provision of products for purchase that consumers want when they want them}	**RA**	**aa:** Franchise Store!	G6
	b: SEJ!	{IT support enabling franchise store to maximize us of limited resources to meet consumer demand}		**bb:** SEJ!	G1-6, O3-5
	c: SEJ!	{Logistics coordination}		**cc:** SEJ!	O1, O2
	c1: Combined Delivery Center!	{Pick-up of products from suppliers}		Supplier!	O1
	c2: Combined Delivery Center!	{Delivery of products to franchise stores}		Combined Delivery Center!	O1
	Supplier!	{Ex-factory delivery}		**dd:** SEJ!	O1, O2
	d: SEJ!	{Product order}		Supplier!	O1
DB	**e:** SEJ Host Computer!	{Data collection and processing}	**RB**	**ee:** SEJ!	O6, O7, T3-T5
	f: SEJ Host Computer!	{Weather report query and receiption)		**ff:** SEJ Host Computer!	T3,T4
	g: SEJ Host Computer!	{Consumer profile, purchase, and store inventory data collection}		**gg:** SEJ Host Computer!	O8, T3
	h: SEJ Host Computer!	{Logistics services request}		**hh:** SEJ Host Computer!	T2
	i: SEJ Host Computer!	{Product order}		**ii:** SEJ Host Computer!	T1
DC	**j:** Franchise Store Computer!	{Product flow}	**RC**	**jj:** Franchise Store Computer!	T9
	k: Franchise Store Computer!	{Stock monitoring}		**kk:** Handheld Scanner!	T7
	k1: Handheld Scanner!	{Barcode scan}			
	l1: Clerk!	{Product shipment reception scanning}		**ll:** Clerk!	T6
	l2: Clerk!	{Report reading}			
	m: Franchise Store Computer!	{Display of graphical reports of sales performance data, recommended orders and actions, data entry of clerk's decisions}		**mm:** GOT!	T9-11
	n: Franchise Store Computer!	{Transmission of consumer profile, consumer purchase, and inventory flow data}		**nn:** Franchise Store Computer!	O9, T8
	o: POS!	{consumer purchase and profile data collection and transmission}		**oo:** POS!	O10, T13
DD	**p:** POS!	{Opening cash drawer}	**RD**	**pp:** POS!	O11, T14
	q: POS!	{Product barcode scanning}		**qq:** POS!	T16
	r: Clerk!	{Product barcode scanning, consumer profiling and profile data entry}		**rr:** Clerk!	T12, T16-22
	r1: Clerk!	{Product barcode scanning}		POS!	T14, T17
	r2: Clerk!	{Consumer age estimation and gender check by sight}		**ss:** Clerk!	T15, T18, T22

5.2.1 Domain DA and Requirements Set RA

Analysis of domain context and requirements at the highest level begins with the construction of a Weill and Vitale business model as a problem diagram, as discussed in Section 4.2.

Step 1: Identify the business model *participants*, i.e., the *organization of interest, suppliers, allies, customers,* and *consumers*. In reviewing the SEJ case, we identify each participant respectively as *SEJ, Supplier* (of various types)*, Combined Delivery Centers, Franchise Stores* and their individual patrons, the *Consumers,* as described in Table 5. These are the business model *participants* that appear as *domains of interest* in *DA*.

Step 2: Identify the relationships among the participants, which we treat as *shared phenomena* describing flows of money, product, and/or information among *domains of interest*, indicated by *interfaces* in the context diagram *DA*. The shared phenomena for DA are listed in Table 4. *Franchise Stores* provide products for purchase to *Consumers* (a). *SEJ* shares information with *Franchise Stores* to enable stores to maximize use of limited resources (b). *SEJ* also shares information with the *Combined Delivery Centers* to coordinate the supply chain (c). *SEJ* orders products from *Suppliers* for delivery to stores (d). The *Combined Delivery Centers* pick up product orders from *Suppliers* (c1) and deliver product orders to *Franchise Stores* (c2).

Table 5. Participants in the SEJ business model

Business Model Participant Type	**Seven-Eleven Japan (SEJ) Business Model**
Organization of Interest	*Seven-Eleven Japan (SEJ)*
Suppliers	*Suppliers* of various manufactured and fresh, processed products like box lunches
Allies	*Combined Delivery Centers* that possess fleets of trucks to transport products, and warehouse to collect products and consolidate shipments.
Customers	*Franchise Stores,* SEJ's direct customers
Consumers	The individual *Consumers* who patron the stores

Step 3: Identify the strategic requirements of the business model and represent these as a goal model. As mentioned previously, this is accomplished by a technique using VMOST analysis and goal model construction according to the BRG-Model rules presented in [25, 26]. The requirements in *RA* reference and constrain the domains of interest in *DA*. *SEJ* promises each owner of a *Franchise Store* (bb) that it will enable the store to maximize use of limited floor space (G3), shorten inventory turnover time (G4), reduce lost sales opportunities and lost store customers (G1), and minimize unsold perishable goods (G2) while guaranteeing their freshness (G5) [33]. *SEJ* must enable each *Franchise Store* to stock products that consumers want when they want them according to continuously changing consumer needs (G6) (aa, bb), ensuring that the *Combined Delivery Centers* (cc) deliver stock from *Suppliers* (dd) to stores just-in-time (O1). *SEJ* achieves this by supporting effective stock order decision-making (O3) via an ability to forecast consumer demand (O5) for each *Franchise Store* (bb). *SEJ* coordinates a supply chain via a data network (O2) linking every *Franchise Store*, *Supplier*, and *Combined Delivery Center* (cc, dd), which enables *SEJ* to control inventory in real time (O4) of each *Franchise Store* (bb).

Domain DA and requirements set RA thus describe a Weill and Vitale business model of SEJ. The requirements set RA, in the form of a goal model, describes the objectives and activities of SEJ's business strategy, whereas the domain context DA describes business model context; i.e., the business model participants and the relationships among them. The DA-RA problem diagram defines both the scope of the business problem to be solved, and the critical strategic, business objectives that are to be met for success. The context diagrams of further problem diagrams in progression discussed below are each derived from DA. Requirements, in the form of goal model entities appearing in further problem diagrams in progression, will be validated against those in RA in order to validate strategic alignment.

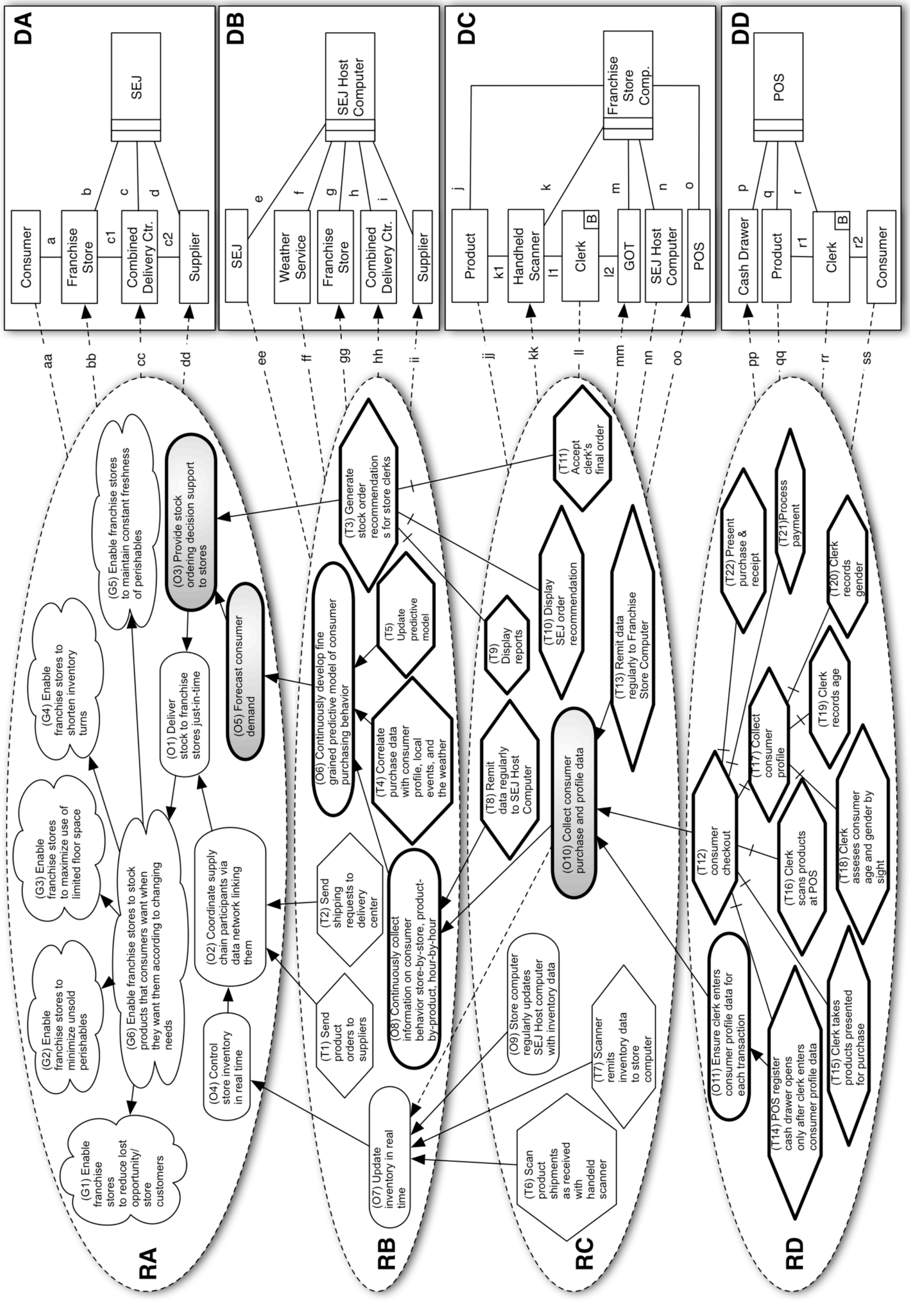

Figure 10. SEJ progression of problems: combined goal model and cotext diagrams

5.2.2 Domain DB and Requirements Set RB

While DA-RA describes what SEJ intends to achieve as a business, it provides little in terms of concrete requirements from which to begin building a system. DB represents a decomposition of DA, and a projection of the SEJ problem context. In DB, the *SEJ Host Computer*, a domain of interest within *SEJ*, serves as the *machine*. SEJ uses the *SEJ Host Computer* (e) for data collection and processing. The *SEJ Host Computer* shares phenomena with a *Weather Service* (f) for predictions for and records of the local weather conditions of the stores, each *Franchise Store* (g) for gathering consumer profile, purchase, and store inventory data, each *Combined Delivery Center* (h) for requests for logistics services, and each *Supplier* (i) for product orders.

The *SEJ Host Computer* provides stock ordering decision support (O3 in RA) by generating stock order recommendations (T3) for *Franchise Stores* (gg). The *SEJ Host Computer* coordinates a supply chain sending product orders (T1) to *Suppliers* (ii) and shipping requests (T2) to *Combined Delivery Centers* (hh), and then by maintaining up-to-date information on inventory in real time (O7) for *SEJ* (ee). To forecast consumer demand for stores, the *SEJ Host computer* continuously develops a fine-grained predictive model of consumer purchasing behavior (O6) for *SEJ (ee)* by collecting information on consumer behavior store-by-store, product-by-product, hour-by-hour (O8) from each *Franchise Store* (gg). After collecting weather data from the *Weather Service* (ff), the *SEJ Host Computer* correlates purchase data with individual consumer profiles, neighborhood events, and local weather for each store (T4), updating the predictive model continuously (T5) for *SEJ* (ee).

5.2.3 Domain DC and Requirements Set RC

The *SEJ Host Computer* relies heavily on gathering data from the franchise stores. To facilitate this, each store is equipped with a *Franchise Store Computer*. The *Franchise Store Computer* shares phenomena with the *Product* (j), to track product flow from inventory delivery intake to shelving and either purchase or scrapping, a *Handheld Scanner* (k), for monitoring inventory by scanning *Product* barcodes (k1), used by the *Clerk* particularly during reception of inventory shipments (l1), a *Graphic Order Terminal (GOT)*, used by the *Clerk* (l2) for reading of sales performance reports, stock order recommendations from SEJ, and data entry of the *Clerk's* stock order decisions (m), the *SEJ Host Computer* (n), to which consumer profile, purchase, and inventory flow data are transmitted, and a *Point-of-Sales (POS) register* (o), at which consumer purchase and profile data are collected.

The *Clerk* (ll) scans product shipments when received (T6) using *Handheld Scanners* (kk) that remit inventory data to the *Franchise Store Computer* (T7). The *Franchise Store Computer* regularly updates (O9) the *SEJ Host Computer* (nn). A store clerk may use the *GOT* (mm) to analyze real-time sales performance reports (T9) and view SEJ stock order recommendations (T10), and then either accept or make changes to a recommendation (T11). The *POS* (oo) collects consumer profile and purchase data (O10), which it regularly remits (T13) to the *Franchise Store Computer,* which in turn regularly remits this data (T8) to the *SEJ Host Computer* (nn).

5.2.4 Domain DD and Requirements Set RD

The *Franchise Store Computer* relies on the *POS* for gathering consumer profile and purchase data. DD describes the domain context of the *POS.* The *POS* shares phenomena with the *POS Cash Drawer* (p), which opens only after the clerk has entered customer profile data, the *Product* (q), whose barcode is scanned by the *POS*, and the *Clerk* (r), who performs the tasks of barcode scanning (r1) and *Consumer* profiling (r2).

Consumer profile data consists of the consumer's approximate age and gender, the products purchased, and the location, time and date of purchase. SEJ collects this data as part of the checkout process (T12, T14-22) at the *POS.* As part of this process, the *Clerk* (rr) is expected to take note of the *Consumer's* gender and approximate age by sight (T18), and then record the age (T19) and gender (T20) at the *POS*. *SEJ* relies on store clerks to enter consumer profile data, as only clerks have the direct interaction with consumers necessary to perform this data collection. In order to help ensure that clerks enter consumer profile data for each transaction (O11), the *POS (pp)* opens the *Cash Drawer* only after the clerk enters the data (T14) [53].

At this point, we have refined requirements down to a relatively low level of detail, equivalent to the starting point of many requirements engineering problem examples appearing in [27]. We do not continue with further refinement, because to do so would simply be to describe a standard software requirements specification exercise. Our aim is to illustrate an explicit link between system requirements and the objectives of business strategy, and the processes that support business strategy.

5.3 Validating Strategic Alignment

In this section, we demonstrate how it is possible to validate alignment with and support for strategic objectives of low-level systems requirements. We examine two critical, strategic-level goals appearing in RA of Figure 10, (O3) and (O5), and the requirements and processes that support them. Section 5.3.1 demonstrates validating low-level requirements against business strategy via the traceable contribution links of the goal model in Figure 10. Section 5.3.2 then illustrates cross-referencing process model detail against the goal model and context diagrams in Figure 10 using RADs.

5.3.1 Validating Requirements against Strategy via Contribution Links

In the case of the POS register, we understand the requirement *T17* in which the *Clerk* collects the consumer profile in terms of the functionality of the *POS* described by requirements *T18-20* in RD within the context of DD. At the same time, we also understand these requirements in terms of their importance to the achievement of SEJ's strategic business objectives in RA by tracing the contribution relationships up through the goal model. The function of collecting the consumer profile data in *T18-20* is to achieve *T17* and constitutes a critical part of the *checkout* task in *T12*. *T12* contributes to the achievement of *O10* in RC, *Collect consumer purchase and profile data,* which contributes to *O8* and then to *O6* in RB. *O6* contributes to *O3*, then to *O1*, and *G6*, *Enable Franchise stores to stock the products consumers want when they want them according to changing needs,* in RA. In this manner, we are able to trace how the lowest-level system requirements align with and provide support for the strategic business objectives.

The goals in RA are critical to the success of SEJ's business, as these represent SEJ's *strategic* objectives. Achievement of these is dependent upon the sub goals and tasks that contribute to them. Assuming a failure to meet *T17* in RD for example, we might also find a failure to meet requirements up through RC, RB, and RA.

Failing to meet requirements in RA, could spell disaster for SEJ, as this could mean failure to achieve SEJ's core goals in enabling the franchise stores to meet the needs of the end consumers who shop there. It is for this reason that understanding how low-level IT systems requirements ultimately support the business strategy is so critical when validating requirements.

5.3.2 Cross-referencing the Process Model

While the SEJ goal model tells us much about the activities and processes of the SEJ system and the context diagrams describe the context in which these occur, there is no process detail or notion of sequential order. To describe process aspects of the requirements for SEJ, process modeling is helpful.

In this section, we examine two interrelated SEJ processes: one process addresses achievement of consumer purchase and profile data and checkout, *O10* of Figure 10, represented as a RAD in Figure 11; the other process is related to consumer demand forecasting and decision support for stores, achieving *O3, O5* in Figure 10, represented as a RAD in Figure 12. Goals *O3, O5* and *O10* are shaded for your ease of reference in Figure 10. In addition, all goal model entities that are related to the process models are outlined in bold in Figure 10. The RADs in Figure 11 and Figure 12 also contain roles that are equivalent and refer to the domains of interest in the context diagrams in Figure 10.

Note that for each interaction, activity, and goal appearing in the RADs, a cross-reference appears at the beginning of each label in parentheses. Interactions cross-reference both an entity in the goal model and a shared phenomenon in the context diagrams, for example *(G1, c)*. The purpose of the cross-reference of an interaction is to enable an analyst to situate the interaction in both the goal model and the context diagrams in order to validate organizational alignment between the process model and the goal model, and between the process model and the context diagrams. Note that the goal model cross-reference does not necessarily correspond in an exact one-to-one manner. Recall that process models are intended to describe a degree of detail that is awkward in a goal model. Thus, the goal model cross-reference refers to the goal entity that is *closest* to the activity described in the process model. The shared phenomenon cross-reference, however, is a one-to-one correspondence. As discussed previously in Section 4.4, the interaction between roles in a RAD corresponds precisely to shared phenomena between domains of interest in a context diagram.

5.3.2.1 Collecting Consumer Purchase and Profile Data, and the Checkout Process

Figure 11 describes the RAD that achieves *O10*. The *Consumer* initiates the process by presenting products for purchase to the *Clerk* (*T15, r1*). The *Clerk* takes the *Products* presented for purchase (*T15, r1*) and scans the product barcodes at the *POS*, (*T16, r, r1*) and (*T16, q*). The *POS* keeps a running sum of payment due as products are scanned *T12*, and then presents a total payment due to the *Clerk* (*T21, r*), who informs the *Consumer* (*T21, r2*). The *POS* prompts the *Clerk* for the consumer's age (*T17, r*), the *Clerk* looks at the consumer to assess age and gender (*T18, r2*), and the *Clerk* enters the consumer's age (*T19, r*).[1] The *POS* prompts the *Clerk* for the consumer's gender (*T17, r*), and the *Clerk* enters it (*T20, r*). The black diamond in the *POS* role of the RAD in Figure 11 is a *token.* In this case, the *POS* notes that the *Clerk* has entered the required profile data, and may now continue by opening the *Cash Drawer* in order to accept payment (*T14, p*).

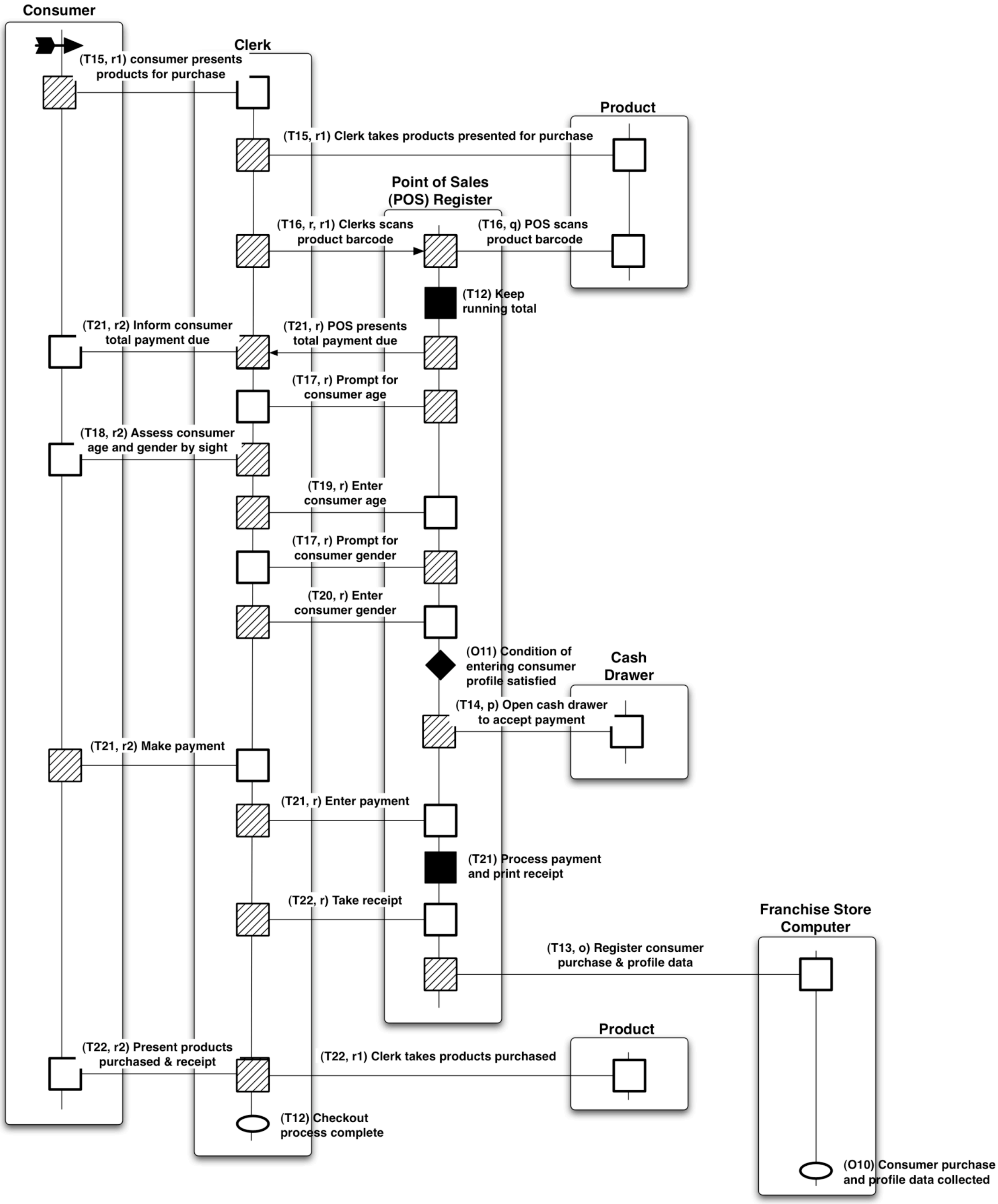

Figure 11. Collecting Consumer Purchase and Profile Data, and Checkout Achieving O10 in Figure 10

---

[1] The SEJ case literature does not elaborate on how age is entered, but similar POS systems in operation today offer four or five age ranges from which the clerk is to select one as an approximate age.

The *token* in the process, effectively satisfies *O11*, to ensure that the clerk enters consumer profile data for each transaction. The checkout process continues with the *Consumer* making payment to the *Clerk* (*T21, r2*), who enters the payment into the *POS* (*T21, r*). The *POS* processes the payment and prints a receipt *T21*, which the *Clerk* takes (*T22, r*), and the *POS* registers the consumer purchase and profile data on the *Franchise Store Computer* (T13, o), the first interaction in the decision support and forecasting process in Figure 12 discussed above. The *Clerk* takes the *Products* (*T22, r1*), and presents them to the *Consumer* along with the receipt (*T22, r2*), completing the checkout task *T12* and achieving *O10* as consumer profile and purchase data has been collected.

5.3.2.2 Decision Support and Demand Forecasting Process

The process to achieve *O3* forecasting consumer demand and *O5* provide stock ordering decision support, described Figure 12, relies heavily on a achievement of *O10*, collecting consumer purchase and profile data, which occurs largely during the checkout process and starts where the RAD in Figure 11 ends. It encompasses a number of roles, including the *Clerk,* the *POS,* the *GOT,* the *Franchise Store Computer,* the *SEJ Host Computer,* and the *Weather Service.* The *POS* remits customer purchase and profile data to the *Franchise Store Computer* (*T13, o*), which then remits aggregate purchase and profile data to the *SEJ Host Computer* (*T8, n*), achieving *O10* and *O8* as consumer purchase and profile data is collected for both the *Franchise Store Computer* and the *SEJ Host Computer.* The *SEJ Host Computer* retrieves weather data associated with store locations and times (*T4, f*), correlates purchase data with consumer profile, local events, and weather (*T4*), and updates the consumer behavior prediction model *T5*, satisfying *O6*. The *SEJ Host Computer* then retrieves a current weather forecast from the *Weather Service* (*T4, f*), generates a stock order recommendation for stores (T3), achieving (O5) as consumer demand has been forecast. The *SEJ Host Computer* transmits the recommendation to the stores via the *Franchise Store Computer* (*T3, g*). The *Clerk* queries the *GOT* for store performance reports and SEJ's stock order recommendations (*T9, T10, l2*). The *GOT* relays the query to the *Franchise Store Computer* (T9, T10, m), which sends reports and the stock order recommendations for display on the *GOT* (T9, T10, m). The *Clerk* examines the reports and recommendations (*T9, T10, l2*), and then enters changes to the order recommendations, or simply accepts the recommendation with no changes (*T11, l2*), which the *GOT* remits to the *Franchise Store Computer* (*T11, n*), which in turn remits to the *SEJ Host Computer* (*T11, n*). The *SEJ Host Computer* updates the order *T11*, achieving *O3*, as stock ordering decision support has been provided.

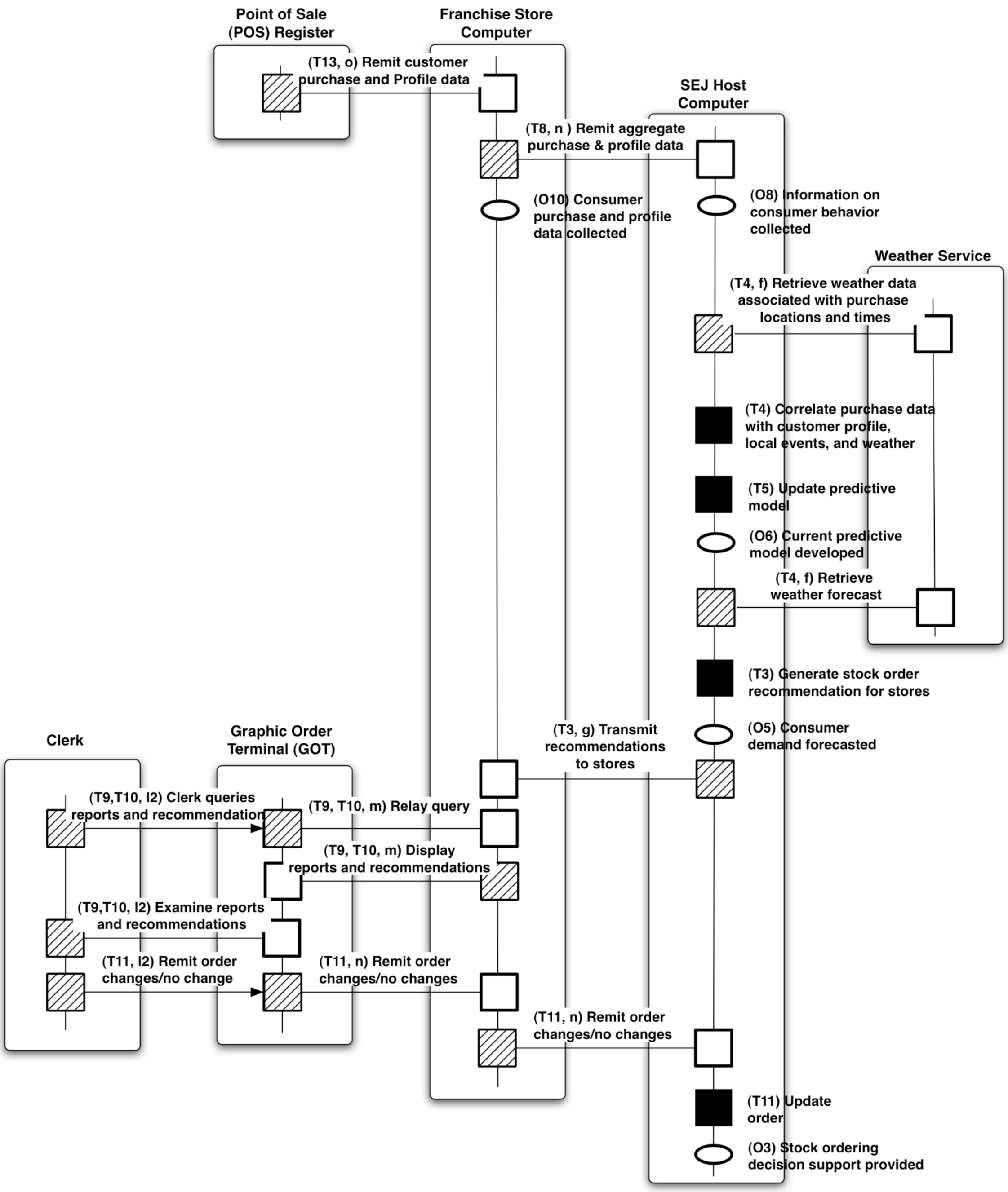

Figure 12. Decision Support and Demand Forecasting Process Achieving (O3, O5) in Figure 10

5.3.3 Summary of Process Models

The business processes presented describe sequential activities and a degree of process detail simply not possible to represent using a goal model and/or context diagrams alone. The walkthrough of the two RADs in Figure 11 and Figure 12 demonstrates cross-referencing of business processes with a goal model of strategic organizational intent and organizational context. This cross-referencing enables explicit traceability from process activities to the organizational goals they achieve. In addition, the cross-referencing also identifies explicitly where a business process sits in the overall strategic intent of the organization, and its organizational business structure. As processes detail aspects of system requirements, the cross-referenced process models also help ensure alignment of requirements with business strategy and the processes that support that strategy.

6 Discussion and Evaluation

The integration of goal modeling with problem diagrams has a number of advantages over using either technique in isolation. Each goal entity refers to specific shared phenomena between domains of interest within the referred domain context. This explicit referencing of *shared phenomena* enables verification of requirements in the context within which they occur at an appropriate level of contextual abstraction. The goal model provides a mechanism for verifying alignment as it enables explicit connections to requirements at adjacent levels in terms of super goals and sub goals. Each partition of requirements as part of a larger goal model represents a smaller and more manageable portion of that goal model, situated clearly within its domain context. Thus, context diagrams help ensure that requirements are consistent with business and system context, while the goal model helps ensure that system requirements achieve business objectives. Integrating goal modeling with problem diagrams thus helps improve manageability of requirements analysis of complex systems.

The Jackson problem diagram framework also offers a number of advantages when modeling requirements for business strategy. First, interfaces, constraints, and references, which effectively separate requirements from shared phenomena, provide a superb mechanism for helping confirm consistency, completeness, and correctness [59] of business requirements. For complex systems, such as those of strategic organizational IT, physical entities share many phenomena and are constrained or referenced by, in most cases, an even greater number of requirements. For example, two domains of interest, a *clerk* and a *consumer,* might share a "checkout" phenomenon, but the requirements constraining or referencing the domains might include *scan products, ask customer for residence postal code, estimate customer age,* among others. This separation forces a requirements analyst to do some cross-checking; i.e., *have all the requirements for this shared phenomenon been included; has whom or what is responsible for the requirement been correctly identified; does this requirement make sense in the domain context?* Indeed, the separation of shared phenomena from requirements constraints and references is somewhat akin to the double-entry system of accounting [60], the primary difference being that you cannot simply do the sums to ensure balance, at least not for requirements at the high-level of abstraction in modeling business strategy. However, a problem diagram does break the problem down for the analyst, and simplifies the confirmation of the rationale for the requirement from *two* perspectives. In our experience in performing the analysis of the SEJ case, this was helpful, as we were able to catch requirements inconsistencies that we might have missed, had we used goal modeling alone or other more conventional requirements analysis techniques. Second, the progression of problem diagrams from the strategy level toward the system level enables a partitioning of requirements according to level of abstraction with reference to appropriate domain context. The progression of problems approach to systems with multiple stakeholder needs, from strategic and tactical managers to direct, operational users, is one possible solution to the obsequious problem in requirements analysis of managing "high-level" and "low-level" requirements, or what Davis calls the "what versus how" problem [59, 61]. Third, the problem diagram framework enables application of the Weill and Vitale business model framework [33]. Using the Weill and Vitale business model framework as a problem diagram is a means to systematically define the scope of the organizational IT problem in a requirements analysis context. As mentioned previously, good scope analysis has been identified as leading both to good requirements analysis and IT project success [47, 48].

In addition, the cross-referencing of RADs [30] to both the goal model and context diagrams enables explicit alignment of business process models with business strategy. Understanding the organizational goals that drive business processes has long been recognized as critical to effective and successful business process modeling [30, 62]. In the requirements engineering domain, some research has attempted to integrate goal modeling with business process modeling techniques [19]. However, B-SCP represents the first significant attempt to link business process modeling to an explicit model of organizational business strategy so that a process might be clearly situated within the strategic context of the business. We consider this a major contribution to the field of requirements engineering.

However, applying our approach to the SEJ case study was not without difficulties. First, we did not find the process of decomposing context from the Weill and Vitale model, represented as DA in Figure 10, into lower level problem context projections to be entirely straightforward. Problem decomposition depends upon projections that include the parts of the contextual elements that are relevant, i.e., useful and necessary, to describe the particular problem being addressed [27]. Those contextual elements can exist at multiple levels of abstraction, making decisions of which ones to include and where to include them even more difficult. Jackson provides some heuristic advice on how to decompose problems, but rightly states this is not an exact science and there are no hard and fast rules that will always work [27]. Jackson recommends identification of the core problem, any ancillary problems, and advises that there are non-standard sub-problem decompositions as well as standard ones. However, Jackson's heuristics are most useful after the problem context has been decomposed to a level "close to the machine." We thus found ourselves having to perform context analysis iteratively, modeling and then remodeling, until we felt we had a set of context diagrams that convincingly represented the SEJ problem context.

In addition, we recognize some potential limitations to the analytical approach we describe. First, we treat strategic alignment as a state of being. Requirements are determined to be in or out of alignment according to a snapshot in time of an organization's strategy, context, and processes, as does most research in strategic alignment [63]. In recent years, it has been recognized that many organizations learn their strategies as they go, and develop them through experience and trial and error over time [64]. This implies that an organization may not know what its strategy is or ought to be at a given moment in time. Some recent strategic alignment research recognizes this problem, and proposes a framework for organizational IT infrastructure investment decisions that maximizes an organization's flexibility and agility in executing options on future IT projects [65]. We have successfully applied B-SCP in this context as a requirements engineering framework for organizational IT infrastructure using the case of a large e-business initiative of a major Australian bank [66]. The case study indicates that B-SCP could be used to identify and describe IT infrastructure requirements that support an "emergent" organizational business strategy [64].

6.1 Comparison with Other Approaches

Much requirements engineering research has focused on use of *i** [16] to model requirements of organizational information systems [67-71], particularly as part of the "early requirements analysis phase" of the TROPOS methodology [17]. While we make use of *i** goal modeling in B-SCP, we depart from standard *i** in a number of ways. First, we separate requirements from domain context in accordance with Jackson's problem diagram framework [27]. This separation of concerns enables us to leverage the existing strategy analysis techniques of VMOST analysis [28] and the BRG-Model [29], which similarly treat an organization's intent and motivation separately from the context in which these occur, as demonstrated in [25, 26]. In contrast, *i** [16] and TROPOS [17] mix physical entities, i.e., actors, agents, and IT resources, directly into the goal model. B-SCP treats physical entities as *context*, and represents them in Jackson context diagrams. One of the reasons *i** mixes contextual and goal entities is to highlight requirements dependencies between agents [16]. In B-SCP, these dependencies are understood as shared phenomena, possibly making them less apparent in comparison to *i**. At the same time, this can be an advantage when modeling complex systems with multiple dependencies between agents. However, the mixing of contextual entities into the goal model in *i** has drawbacks. First, it obscures how organizational entities relate with each other independently of organizational goals, tasks, and other requirements, thus also obscuring how entities in a business model, an organizational structure, or an IS infrastructure relate to each other independently of requirements. Essentially, *i** obscures the picture of the 'as is' model of context. In addition, our experience of using *i** has shown that modeling multiple dependencies of even moderately complex systems tends to hopelessly muddle a goal model [31, 32]. Indeed, in a study of a large-scale industrial project in which *i** was used for requirements analysis [70], practitioners were unable to understand *i** models well enough to validate the requirements of the system they were building. The academic researchers providing requirements engineering consulting for the project, exasperated, ultimately resorted to transposing the *i** models back into standard English text, with no goal modeling [72]. While the study did not explicitly conclude that this was necessary because of the problems of clutter we have experienced, the *i** diagrams presented in [70, 72] appear so medusa-like that we suspect clutter may have been the cause. Thus, separation of concerns and using projection may help make *i** goal models more comprehensible and easier to use. Incidentally, while much of the *i** literature refers to the "dependency relationships" between agents in a systems as "strategic", e.g., [71, 73] the use of the word has nothing to do with *business strategy* [13, 15, 64]. Indeed, it is unclear why the word "strategic" is used to describe *i** dependency relationships at all. Second, *i** has no means of modeling processes other than as representing them as *task* entities in a goal model. *Tasks* may be decomposed into *sub tasks* in order to show more detail. However, the *i** approach to process modeling has two major shortcomings: (1) there is no means of showing sequential order, parallel processes, or changing processes dependent on conditions; and (2) as *tasks* are detailed by decomposing them into *sub tasks*, a goal model bloats rapidly to the point where it becomes unusable. It is for these reasons that in B-SCP, we prefer to represent process details using RADs [30], and cross-referencing RADs with both the goal model and context diagrams.

There also exist software and systems development methodologies for organizational IT that include business modeling that incorporates the scope of business strategy, such as *Information Engineering* [74] and more recently *Business modeling with UML* [75]. Each of these methodologies propose frameworks according to the themes of *strategic content*, *business context,* and *business process*, although each refers to these themes by different names and use differing means of representation.

In *Information Engineering,* Finkelstein details a methodology for information systems development that incorporates a means for eliciting and modeling business strategy [74]. *Information Engineering* also presents a variation on the themes of strategy, context, and process with "strategic model", "organization structure", and "process modeling" [74]. Strategy is represented as a set of goals. Finkelstein describes an interviewing process along with template questionnaires in order to help analysts elicit business strategy from management stakeholders. Goals are represented according to a note card style template describing the "Goal" and "Concerns and Issues" in text [74]. Context is represented as an organizational chart. Processes are modeled using *procedure maps*, a means to illustrate "programming logic," representing the "processing associated with one (or several) business events" [74].

One of the most significant features in *Information Engineering* is the support it provides for business analysts to elicit and document business strategy from executive stakeholders in structured interviews [74]. For this purpose, B-SCP proposes VMOST analysis [28] in [25, 26]. However, Finkelstein provides more detailed questions and structured interview documents [74] than what is provided in VMOST analysis [28]. Despite this, *Information Engineering* suffers from a number of weaknesses. First*, Information Engineering* was developed during a time when IT was viewed primarily as an internal support function for organizations, which is no longer the case today [76]. With the exception of the strategy analysis part, the business analysis focuses heavily on what is internal to the organization. Organizational context is represented as an organizational chart, without reference to organizational entities external to the firm. External entities might include customers, suppliers, and business partners that are important in e-business systems for example, which are relatively common types of organizational IT systems in use today. Similarly, procedure maps focus on routine, internal operational procedures represented in a notation developed for logical sequence programming, in which even moderately complex business processes are extremely difficult to represent, if not impossible. In addition, while *Information Engineering* uses goals to represent strategy, it offers no goal modeling notation, and thus there is no linking of goals in contribution hierarchies through which a coherent strategy can be understood and traced. Overall, *Information Engineering* is a development methodology for traditional management information systems, with an inward-facing approach to business modeling that emphasizes data modeling and database transaction programming over requirements analysis.

In *Business Modeling with UML* [75]*,* Eriksson and Penker propose an extended version of UML for the purpose of business modeling, which has been applied to modeling business strategy in research [77-79]. *Business modeling with UML* presents a variation on the themes of strategy, context, and process with business model views of business vision, business structure, business process, and attached to business process, business behavior. To support these views, Eriksson and Penker propose a set of UML extensions for business modeling, including simple goal modeling, context modeling of organizational structure, and process modeling as an "assembly line" [75]. The extension for goal modeling consists of only one goal type and one contribution relationship type. No extension is offered for domain context modeling, which is performed as UML class diagrams representing organizational structure. The UML extension for process modeling treats processes as a set of ordered activities that add value to deliver an output to a market or customer, like an "assembly line process" [75].

However, business modeling with UML has a number of shortcomings. First, the UML goal modeling extension lacks the richness of other established goal modeling notations and frameworks such as *KAOS* [34], *i** [16], *GBRAM* [35], and the BRG-Model [29], which B-SCP leverages in its *strategy* theme. Indeed, the UML goal modeling extension is an inferior substitute for each of these. Also, the "assembly line process" modeling proposed in [75] focuses on process flow, which unfortunately makes it an awkward notation for understanding business processes that do not match the "assembly line" pattern of bundles of work passing through manufacturing-type processes to build a product. In addition, modeling domain context using UML class diagrams is awkward. In the real world of an organization, do all departments of a firm inherit attributes from some common super class of departments? Are the attributes of a clerk, such as name, employee ID, date hired, relevant, i.e., useful and necessary, to the contextual domain of all problems in which the *clerk* appears? In a "real" ATM system, does a "cash note" know its attribute of being either "on hand" or "dispensed" as proposed in [80]? Such concepts, which are common to object-oriented software design, bear little resemblance to the context of the real world and are at times, as in the case of [80] mentioned above, absurd. Class diagrams were originally developed for the purpose of modeling object-oriented software design [81], and that is what they should be used for. They are simply not an appropriate notation for modeling domain context of the real world [27, 39, 82, 83].

The advantage of *Business modeling with UML*, as proclaimed by its authors, is that it is UML, and easy to pick up for those already familiar with UML [75]. Unfortunately, *Business modeling with UML* also has three major weaknesses in comparison with B-SCP when used for business-IT alignment in requirements analysis. First, Eriksson and Penker never demonstrate how the "business views" connect and integrate with each other [75]. Business vision, structure, and process are modeled independently of each other and have no explicit cross-referencing mechanism to help verify alignment of views with each other. For example, if there is a change in the organizational structure, there is no mechanism for tracing how that change might impact organizational goals or business processes. Second, *Business modeling with UML* appears to restrict its "business structure" view to what is internal to the enterprise. Indeed, the examples presented in [75, 79] focus heavily on internal, operational concerns, with little reference to the external environment. These are simply not examples that address the scope of business strategy. Business strategy is primarily about what is outside the enterprise [13-15, 64, 84, 85], such as customers, suppliers, and business partners, and thus the external environment is critical to strategic problem scope. Indeed, in the SEJ example presented in Section 5, much of the focus is on SEJ's external environment. SEJ's suppliers, logistics allies, and customers, i.e., the franchise stores, are separate and independent companies, external to SEJ, yet fundamental to SEJ's requirements problem context. *Business modeling with UML* severely limits its scope to the peril of those who intend to perform business analysis for systems of strategic import. Third, as the authors state, the UML software architecture model is distinct from the UML business model [75]. This separation of business model from system model severely limits the capacity of using *Business modeling with UML* to provide explicit requirements traceability to business strategy, which is one of the key advantages of B-SCP.

6.2 A Note on Complexity and the Relevance of RE Research

In *Domain-driven design: tackling complexity in the heart of software,* Evans writes, "Technical people enjoy quantifiable problems that exercise their technical skills. Domain work is messy and demands a lot of complicated new knowledge that doesn't seem to add to a computer scientist's capabilities. Instead, the technical talent goes to work on elaborate [technical] frameworks, trying to solve domain problems with technology. Learning about and modeling the domain is left to others" [83].

Indeed, B-SCP is about modeling the real-world domain of an organization's competitive business strategy as part of requirements analysis, as executive management stakeholders expect strategic IT systems to meet the requirements of their business strategy. Business strategy is messy, complicated, and certainly demands a lot of knowledge that has little direct relationship with computer science or software engineering. Understanding business strategy is nonetheless critical to getting the requirements right for organizational IT systems, and is thus within the purview of the requirements engineer.

Evans continues, "Complexity in the heart of software has to be tackled head-on. To do otherwise is to risk irrelevance" [83]. Indeed, business strategy is part of the complexity of engineering of organizational IT. For requirements engineering research to be relevant to strategic organizational IT problems, business strategy must be addressed head-on.

7 Conclusion

Despite the recognized importance of alignment of IT with business strategy, requirements engineering research has yet to focus much attention on this issue. We have presented B-SCP, a requirements engineering framework based on connecting and integrating the themes of strategy, context, and process to help enable modeling organizational IT requirements for the purpose of validating requirements against business strategy, extending a framework originally presented in [25, 26]. We represent each theme using a requirements engineering technique: goal modeling, Jackson context diagrams, and role activity diagrams (RADs) respectively. We integrate strategy and context using a Jackson problem diagram framework. We connect RADs to the goal model and context diagrams via explicit cross-referencing of elements. We leverage the Weill and Vitale framework for business modeling to scope the strategic-level context and requirements of the organizational IT problem. We demonstrate initial validity of B-SCP via application to an exemplar, developed from multiple sources in the literature on Seven-Eleven Japan (SEJ). We find that B-SCP offers promise as a requirements analysis framework for aligning organizational IT requirements with business strategy as it enables explicity traceability between organizational business stategy and IT system requirements.

B-SCP makes the following contributions to requirements engineering research: (1) it presents a requirements engineering framework that integrates business process models explicitly with a model of business strategy; (2) it provides a framework for scoping context and requirements of strategic, organizational IT problems; and (3) it demonstrates a means of validating system requirements against business strategy and the business processes that support that strategy via explicit and traceable links.

At the time of writing, the authors have obtained approval to apply B-SCP in a strategic IT initiative of a major organization. Our intention is to use the project to evaluate, further develop and refine B-SCP, while reporting our experience and results back to the research community.

References

[1] J. D. McKeen and H. Smith, *Making IT happen: critical issues in IT management*. Chichester; Hoboken, NJ: Wiley, 2003.

[2] C. U. Ciborra, "De profundis? Deconstructing the concept of strategic alignment," *Scandinavian Journal of Information Systems*, vol. 9, pp. 67-82, 1997.

[3] R. D. Galliers, "Strategic information systems planning: myths, reality and guidelines for successful implementation," *European Journal of Information Systems*, vol. 1, pp. 55-64, 1991.

[4] M. Porter, "From competitive advantage to corporate strategy," *Harvard Business Review*, vol. 65, pp. 43-59, 1987.

[5] Y. Chan, S. Huff, D. Barclay, and D. Copeland, "Business Strategic Orientation: Information Systems Strategic Orientation and Strategic Alignment," *Information Systems Research*, vol. 8, pp. 125-150, 1997.

[6] A.-M. Croteau and F. Bergeron, "An Information Technology Trilogy: Business Strategy, Technological Deployment and Organizational Performance," *Journal of Strategic Information Systems*, vol. 10, pp. 77-99, 2001.

[7] P. P. Tallon and K. L. Kraemer, "Executives' Perspectives on IT: Unraveling the Link between Business Strategy, Management Practices and IT Business Value," presented at Americas Conference on Information Systems (ACIS2002), Dallas, TX, USA, 2002.

[8] J. Luftman and E. R. MacLean, "Key issues for IT executives," *MIS Quarterly Executive*, vol. 3, pp. 89-104, 2004.

[9] B. H. Reich and K. M. Nelson, "In Their Own Words: CIO Visions About the Future of In-House IT Organizations," *The DATA BASE for Advances in Information Systems*, vol. 34, pp. 28-44, 2003.

[10] P. P. Tallon, K. L. Kraemer, and V. Gurbaxani, "Executives' perceptions of the business value of information technology: A process-oriented approach," *Journal of Management Information Systems*, vol. 16, pp. 145-173, 2000.

[11] R. T. Watson, G. G. Kelly, R. D. Galliers, and J. Brancheau, "Key Issues in Information Systems Management: an International Perspective," *Journal of Management Information Systems*, vol. 13, pp. 91-115, 1997.

[12] J. C. Brancheau, B. D. Janz, and J. C. Wetherbe, "Key issues in information systems management: 1994-95 SIM delphi results," *Mis Quarterly*, vol. 20, pp. 225-242, 1996.

[13] R. W. Oliver, "What is Strategy, Anyway?" *Journal of Business Strategy*, pp. 7-10, 2001.

[14] M. Porter, "What is Strategy?" *Harvard Business Review*, vol. 74, pp. 61-78, 1996.

[15] M. Porter, *Competitive Advantage: Creating and Sustaining Superior Performance*. New York: Free Press, 1985.

[16] E. Yu, "Modeling organizations for information systems requirements engineering," presented at IEEE International Symposium on Requirements Engineering, 1993.

[17] P. Giorgini, J. Mylopoulos, and M. Pistore, "The Tropos methdology: an overview," in *Methodologies and software engineering for agent systems: the agent-oriented software engineering handbook*, F. Bergenti, M.-P. Gleizes, and F. Zambonelli, Eds. Boston, MA; London: Kluwer Academic, 2004, pp. xxx, 505 p.

[18] J. Gordijn and J. Akkermans, "Value-based requirements engineering: exploring innovative ecommerce ideas," *Requirements Engineering Journal*, vol. 8, pp. 114-135, 2003.

[19] V. Kavakli and P. Loucopoulos, "Goal-driven business process analysis application in electricity deregulation," *Information Systems*, vol. 24, pp. 187-207, 1999.

[20] P. Loucopoulos, "The S3 (Strategy-Service-Support) Framework for Business Process Modelling," presented at Workshop on Requirements Engineering for Business Process Support (REBPS'03), Klagenfurt/Velden, Austria, 2003.

[21] C. Rolland, C. Souveyet, and C. Ben Achour, "Guiding Goal Modeling Using Scenarios," *IEEE TRANSACTIONS ON SOFTWARE ENGINEERING*, vol. 24, pp. 1055-71, 1998.

[22] J. A. Bubenko and M. Kirikova, "Worlds in requirements acquisition and modelling," presented at 4th European-Japanese Seminar on Information Modelling and Knowledge Bases, Kista, Sweden, 1994.

[23] P. Loucopoulos and E. Kavakli, "Enterprise Modeling and the Teleological Approach to Requirements Engineering," *International Journal of Cooperative Information Systems*, vol. 4, pp. 45-79, 1995.

[24] R. E. M. Champion and T. T. Moore, "Exploiting an enterprise model during systems' requirements capture and analysis," presented at International Conference on Requirements Engineering (ICRE'96), Colorado Springs, CO, 1996.

[25] S. J. Bleistein, K. Cox, and J. Verner, "Validating Strategic Alignment of Organizational IT Requirements Using Goal Modeling and Problem Diagrams," *Journal of Systems and Software*, 2005 (Accepted for publication).

[26] S. J. Bleistein, K. Cox, and J. Verner, "Strategic Alignment in Requirements Analysis for Organizational IT: an Integrated Approach," presented at 20th ACM Symposium on Applied Computing (SAC'05), Santa Fe, NM, USA, 2005.

[27] M. Jackson, *Problem Frames: Analyzing and Structuring Software Development Problem*, 1st ed: Addison-Wesley Publishing Company, 2001.
[28] R. Sondhi, *Total Strategy*: Airworthy Publications International Ltd., 1999.
[29] A. B. Kolber, C. Estep, D. Hay, D. Struck, G. Lam, J. Healy, J. Hall, J. A. Zachman, K. Healy, M. Eulenberg, N. Fishman, R. Ross, T. Moriarty, and W. Selkow, "Organizing Business Plans: The Standard Model for Business Rule Motivation," The Business Rule Group November 15 2000.
[30] M. A. Ould, *Business processes: modelling and analysis for re-engineering and improvement*. Chichester; New York: Wiley, 1995.
[31] S. J. Bleistein, K. Cox, and J. Verner, "Requirements Engineering for e-Business Systems: Integrating Jackson Problem Diagrams with Goal Modeling and BPM," presented at 11th Asia Pacific Software Engineering Conference, Busan, Korea, 2004.
[32] S. J. Bleistein, K. Cox, J. Verner, and K. T. Phalp, "Requirements Engineering for e-Business Advantage," *Requirements Engineering Journal*, 2005 (Accepted for publication).
[33] P. Weill and M. Vitale, *Place to Space: Moving to eBusiness Models*. Boston: Harvard Business School Publishing Corporation, 2001.
[34] A. Dardenne, A. van Lamsweerde, and S. Fickas, "Goal-Directed Requirements Acquisition," *Science and Computer Programming*, vol. 20, pp. 3-50, 1993.
[35] A. I. Antón, "Goal-based requirements analysis," presented at International Conference on Requirements Engineering (ICRE'96), Colorado Springs, CO, USA, 1996.
[36] L. E. Chung, B. Nixon, E. Yu, and J. Mylopoulos, *Non-Functional Requirements in Software Engineering*, vol. 5, 1st ed: Kluwer Academic Publishers, 1999.
[37] L. Liu and E. Yu, "From Requirements to Architectural Design - Using Goals and Scenarios," presented at ICSE-2001 (STRAW 2001), Toronto, Canada, 2001.
[38] R. Darimont and A. van Lamsweerde, "Formal Refinement Patterns for Goal-Driven Requirements Elaboration," presented at Fourth ACM SIGSOFT Symposium on Foundations of Software Engineering (SIGSOFT'96), San Francisco, 1996.
[39] M. J. Jackson, *Software requirements & specifications: a lexicon of practice, principles, and prejudices*. New York; Wokingham, England; Reading, Mass.: ACM Press; Addison-Wesley Pub. Co., 1995.
[40] M. Haglind and K. H. Cheong, "Striving towards strategic it & business alignment in small and medium-sized electric utilities - a pragmatic approach," presented at 16th International Conference and Exhibition on Electricity and Distribution (CIRED), Amsterdam, The Netherlands, 2001.
[41] G. Walsham, *Interpreting information systems in organizations*. Chichester, West Sussex, England; New York: Wiley, 1993.
[42] G. A. Rummler and A. P. Brache, *Improving performance: how to manage the white space on the organization chart*, 2nd ed. San Francisco, CA: Jossey-Bass, 1995.
[43] K. Cox, K. Phalp, S. J. Bleistein, and J. Verner, "Deriving Requirements from Process Models via the Problem Frames Approach," *Information and Software Technology*, vol. 47, pp. 319-337, 2005.
[44] A. van Lamsweerde, "Goal-Oriented Requirements Engineering: A Guided Tour," presented at 5th IEEE International Symposium on Requirements Engineering, Toronto, 2001.
[45] S. J. Bleistein, A. Aurum, K. Cox, and P. Ray, "Strategy-Oriented Alignment in Requirements Engineering: Linking Business Strategy to Requirements of e-Business Systems using the SOARE Approach," *Journal of Research and Practice in Information Technology*, vol. 36, pp. 259-276, 2004.
[46] J. M. Verner, K. Cox, S. Bleistein, and N. Cerpa, "What requirements engineering practices predict software project success?" presented at 3rd REK-SWDC International Conference on Software Development, Reykjavik, Iceland, 2005.
[47] J. Verner, K. Cox, S. J. Bleistein, and N. Cerpa, "Requirements engineering and software project success: an industrial survey in Australia and the US," *Australian Journal of Information Systems*, 2005 (In press).
[48] J. Verner and W. Evanco, "In house software development: what software management practices are needed for project success," *IEEE Software*, vol. 22, pp. 86-93, 2005.
[49] P. Zave and M. Jackson, "Four Dark Corners of Requirements Engineering," *ACM Transactions on Software Engineering and Methodology*, vol. 6, pp. 1-30, 1997.
[50] N. Melao and M. Pidd, "A conceptual framework for understanding business processes and business process modelling," *Information Systems Journal*, vol. 10, pp. 105-129, 2000.
[51] C. Britton and J. Doake, *Software System Development: a Gentle Introduction*: McGraw-Hill, 1993.
[52] M. S. Feather, S. Fickas, A. Finkelstein, and A. van Lamsweerde, "Requirements and specification exemplars," *Automated Software Engineering*, vol. 4, pp. 419-438, 1997.
[53] M. Bensaou, "Seven-Eleven Japan: Managing a Networked Organization," INSEAD Euro-Asia Centre, Case Study 1997.
[54] S. Whang, C. Koshijima, H. Saito, T. Ueda, and S. V. Horne, "Seven Eleven Japan (GS18)," Stanford University Graduate School of Business 1997.
[55] W. V. Rapp, "Retailing: Ito-Yokado Seven-Eleven Japan," in *Information technology strategies: how leading firms use IT to gain an advantage*. New York: Oxford University Press, 2002, pp. 163-186.
[56] N. Makino and T. Suzuki, "Convenience Stores and the Information Revolution," *Japan Echo*, vol. 44, pp. 44-49, 1997.
[57] R. Kunitomo, "Seven-Eleven is Revolutionising Grocery Distribution in Japan," *Long Range Planning*, vol. 30, pp. 887-89, 1997.
[58] The Economist Newspaper Limited, "Over the Counter e-Commerce; E-Strategy Brief: Seven-Eleven; Japan's Seven-Eleven," *Economist (US)*, vol. 359, pp. 77, 2001.

[59] A. M. Davis, *Software requirements: analysis and specification*. Englewood Cliffs, N.J.: Prentice Hall, 1990.
[60] J. B. Geijsbeek and L. Pacioli, *Ancient double-entry bookkeeping: Lucas Pacioli's treatise (A.D. 1494, the earliest known writer on bookkeeping) reproduced and translated with reproductions, notes, and abstracts from Manzoni, Pietra, Mainardi, Ympyn, Stevin, and Dafforne*. Houston, Tex.: Scholars Book Co., 1974.
[61] A. M. Davis, *Just enough requirements management: where software development meets marketing*. New York, NY: Dorset House Publ., 2005.
[62] P. Keung and P. Kawalek, "Goal-based business process models: creation and evaluation," *Business Process Management Journal*, vol. 3, pp. 17-38.
[63] D. Avison, J. Jones, P. Powell, and D. Wilson, "Using and validating the strategic alignment model," *Journal of Strategic Information Systems*, vol. 13, pp. 223-246, 2004.
[64] H. Mintzberg, B. W. Ahlstrand, and J. Lampel, *Strategy safari: a guided tour through the wilds of strategic management*. New York: Free Press, 1998.
[65] A. F. Thorogood and P. Yetton, "Reducing the technical complexity of business reisk of major systems projects," presented at 37th Annual Hawaii International Conference on System Sciences, Hawaii, 2004.
[66] P. Reynolds, A. Thorogood, S. J. Bleistein, and K. Cox, "A contingency view of requirements enigneering: towards a new research agenda," presented at 1st Workshop on Requirements Engineering for Business Need and IT Alignment (REBNITA'05) held at the 13th IEEE International Conference on Requirements Engineering, Paris, France, 2005 (Submitted).
[67] J. Castro, M. Kolp, and J. Mylopoulos, "Towards Requirements-Driven Information Systems Engineering: the Tropos Project," *Information Systems Journal*, vol. 27, pp. 365-89, 2002.
[68] M. Kolp, P. Giorgini, and J. Mylopoulos, "A goal-based organizational perspective on multi-agent architectures," *Intelligent Agents Viii: Agent Theories, Architectures, and Languages*, vol. 2333, pp. 128-140, 2002.
[69] M. Garzetti, P. Giorgini, J. Mylopoulos, and F. Sannicolo, "Applying Tropos methdology to a real case study," presented at Joint Workshop 'From Objects to Agents': From Information to Knowledge (WOA 2002), Milan, Italy, 2002.
[70] N. A. M. Maiden, S. V. Jones, S. Manning, J. Greenwood, and L. Renou, "Model-driven requirements engineering: Synchronising models in an air traffic management case study," presented at 16th International Conference on Advanced Information Systems Engineering, 2004.
[71] E. Yu and L. Liu, "Modelling Strategic Actor Relationships to Support Intellectual Property Management," presented at 20th International Conference on Conceptual Modelling, ER-2001, Yokohama, Japan, 2001.
[72] N. A. M. Maiden, S. Manning, S. Jones, and J. Greenwood, "Towards Pattern-Based Generation of Requirements from System Models," presented at 10th Anniversary International Workshop on Requirements Enginering: Foundation of Software Quality (REFSQ), Riga, Latvia, 2004.
[73] E. S. K. Yu and J. Mylopoulos, "From E-R to a-R - Modeling Strategic Actor Relationships for Business Process Reengineering," *International Journal of Cooperative Information Systems*, vol. 4, pp. 125-144, 1995.
[74] C. Finkelstein, *Information engineering: strategic systems development*. Sydney; Reading, Mass.: Addison-Wesley, 1992.
[75] H.-E. Eriksson and M. Penker, *Business modeling with UML: business patterns at work*. New York: John Wiley & Sons, 2000.
[76] M. Earl, Khan B., "E-Commerce is Changing the Face of IT," *MIT Sloan Management Review*, pp. 64-72, 2001.
[77] R. Mendes, A. Vasconcelos, A. Caetano, J. Neves, P. Sinogas, and J. Tribolet, "Understanding strategy: a goal modeling methodology," presented at 7th International Conference on Object Oriented Information Systems (OOIS 2001), Calgary, Canada, 2001.
[78] R. Mendes, A. Vasconcelos, A. Caetano, J. Neves, P. Sinogas, and J. Tribolet, "Representing Business Strategy through Goal Modeling," presented at International Conference on Enterprise Information Systems (ICEIS 2001), Setubal, Portugal, 2001.
[79] A. Vasconcelos, A. Caetano, J. Neves, P. Sinogas, R. Mendes, and J. M. Tribolet, "A Framework for Modeling Strategy, Business Processes and Information Systems," presented at 5th International Enterprise Distributed Object Computing Conference (EDOC 2001), Seattle, WA, USA, 2001.
[80] H. Kaindl, "Is object-oriented requirements engineering of interest?" *Requirements Engineering Journal*, vol. 10, pp. 81-84, 2005.
[81] L. Favre, *UML and the unified process*. Hershey, Pa.: IRM Press, 2003.
[82] B. L. Kovitz, *Practical software requirements: a manual of content and style*. Greenwich, Conn.: Manning, 1999.
[83] E. Evans, *Domain-driven design: tackling complexity in the heart of software*. Boston: Addison-Wesley, 2004.
[84] J. B. Quinn, H. Mintzberg, and R. M. James, *The strategy process: concepts, contexts, and cases*. Englewood Cliffs, N.J.: Prentice-Hall, 1988.
[85] G. Hamel and C. K. Prahalad, *Competing for the future*. Boston, Mass.: Harvard Business School Press, 1994.