e-Business cases assessment: from business value to system feasibility

Ziv Baida, Hans de Bruin and Jaap Gordijn

Vrije Universiteit Amsterdam, Mathematics and Computer Science Department, De Boelelaan 1081a, 1081 HV Amsterdam,
The Netherlands
E-mail: zsb@wanadoo.nl
E-mail: brui0006@mail.hzeeland.nl
E-mail: gordijn@cs.vu.nl

Abstract: Innovative e-commerce initiatives present a new way of doing business. Since short time to market is a major requirement in e-commerce, and typically multiple parties (enterprises) are involved, multiple stakeholders have to be convinced quickly that the new way of doing business is technically feasible and economically profitable for them. Consequently, a lightweight approach is required for defining, deriving and analysing multi-enterprise relationships, business cases and requirements. The e^3-*value* framework is a multi-viewpoint requirements engineering method that offers such an approach. It is based on analysing e-commerce initiatives through three stakeholder-based viewpoints. To ensure that these viewpoints keep correlated, to facilitate traceability of requirement decisions, and to support trade-off analysis between requirements we present Feature–Solution (FS) graphs as an extension to the e^3-*value* framework. An FS-graph captures architectural knowledge in the form of desired features and solutions that realize these features. By combining the two methods, we ensure viewpoint integration, which makes it possible to conduct a systematic exploration of design alternatives in an e-commerce initiative. A business value driven information systems architecture for implementing an e-commerce system is generated as a by-product of our method.

Keywords: e-commerce; stakeholders; feasibility; requirements enginee-ring; business cases; viewpoints; architecture.

Reference to this paper should be made as follows: Baida, Z., de Bruin, H. and Gordijn, J. (2003) 'e-Business cases assessment: from business value to system feasibility', *International Journal of Web Engineering Technology*, Vol. 1, No. 1, pp. 127–144.

Biographical notes: Ziv Baida completed his Master degree in computer science at the Vrije Universiteit in Amsterdam, the Netherlands. His Masters thesis discusses the use of architecture visualizations as a tool to support the decision-making process of business managers.

Hans de Bruin is a Lecturer and Researcher at the Hogeschool Zeeland. He received his Masters degree from the Technical University of Delft in 1987 and his Ph.D. from the Erasmus University Rotterdam in 1995. Before joining the Hogeschool Zeeland, he worked as a software engineer and consultant in industry, and he was affiliated as a lecturer and researcher at the Vrije Universiteit Amsterdam. Currently, he teaches computer science at the Hogeschool Zeeland. His research interests include software engineering, in particular software architecture and component-based development, and programming language design.

Jaap Gordijn has a PhD degree from the Free University in Amsterdam and is an Assistant Professor at the same university. He is one of the key developers of the e3value methodology for e-business development and previously held positions in management consultancy firms and industry.

1 Introduction

Over the past few years, a vast amount of e-commerce ideas have been considered. During 1998–1999, the e-commerce hype reached its top. Recently, it became clear that many e-commerce ideas are not successful [1]. Many enterprises doing e-commerce have not been able to create profit with their e-commerce ideas. Some of these companies who relied entirely on future e-commerce profits have gone bankrupt.

An important reason for the failure of e-commerce ideas is the lack of a technical feasible and economically sound value proposition to customers. One explanation for the absence of such a proposition is that stakeholders did not understand the idea very well, and consequently were not able to assess the idea for economical and technological feasibility sufficiently. This lack of understanding was largely caused by the newness of the innovative ideas.

Despite the many failures of e-commerce initiatives, we still believe that many potential successful e-commerce ideas exist, which utilize enabling internet related technical innovations in a profitable way. In addition to new meaningful uses of technology in general, this is also motivated by many industries which are *forced* to find new value propositions. For instance, the digital content industry is facing challenges with respect to new value propositions utilizing internet technology, e.g. how to earn money by streaming music to an end-consumer's device. Consequently, what first is needed is to understand such ideas well, and to assess feasibility.

Additionally, we experienced in the e-commerce tracks we have carried out (in the realm of news provisioning, internet service provisioning, music industry, insurance, and energy [2]), that such tracks are characterized by a wide range of stakeholders (e.g. business and technical responsibles) representing different companies. Business oriented people are concerned with the new, technology enabled, value proposition. Additionally, since business processes need often to be created or changed, stakeholders responsible for intra- and inter-organizational business processes are involved. Also, because e-commerce ideas are so heavily dependent on information technology, information technologists are important stakeholders too during the assessment of the e-commerce idea. Many e-commerce ideas would simply never have been thought of, without an in-depth understanding of their enabling information technology. In conclusion, what is needed is a methodology to explore an e-commerce idea, not only from the economical perspective, but also from the business process and information technology perspective. This makes e-commerce idea exploration and assessment a truly trans-disciplinary activity.

To our knowledge, a trans-disciplinary methodology for e-commerce idea exploration does not yet exist. On the one hand, business oriented approaches such as value chain theory [3] or value web theory [4], but also pure economical approaches which stem from investment theory [5] neglect the importance of information technology. Moreover, these approaches do not have a clear explorative design perspective. We advocate that finding and detailing an e-commerce idea is mainly a design task and therefore should be treated in this way. On the other hand, information technology approaches (especially in the field of requirements engineering and conceptual modelling) lack the notion of *economic value* and how this is created, exchanged and consumed in a multi-actor network. For instance, the Unified Modelling Language (UML) [6] focuses mainly on technical aspects of an e-commerce idea, and at best at business processes needed. Goal-oriented requirements engineering techniques [7] focus on *why* an information system is needed, but do not emphasize on the value proposition perspective. In sum, what is needed is an approach with a focus on business value and technological perspectives, and which can be used as a way of working to design innovative, technology intensive e-commerce ideas.

To do so, we have proposed the e^3*-value* methodology [8,9]. This methodology conceptualizes and formalizes an e-commerce idea so that all stakeholders have a common understanding of the idea, and such that the idea can be analyzed and evaluated, e.g. for potential profitability. In fact, we utilize a conceptual model-based way of working, known from the realm of requirements engineering and conceptual modelling, and we use terminology from economics, marketing and axiology [10]. Additionally, the methodology is lightweight, because e-commerce exploration tracks need to be carried out in a limited timeframe (typically a few weeks or a month).

The e^3*-value* methodology uses multiple viewpoints (borrowed from the realm of requirements engineering, see e.g. [11]), to represent interests of various stakeholders. We separate different foci of stakeholders in different, relatively self-contained viewpoints, so that each stakeholder group can decide on its own on requirement expressions. Seperation of these foci is important because otherwise discussions between stakeholders are confusing and not efficient.

A drawback of using such self-contained viewpoints is that expressions on different viewpoints about the same phenomena tend to diverge. Consequently, we need a mechanism to relate and integrate expressions on different viewpoints in a lightweight fashion, to ensure that all stakeholders are discussing the same idea. At present, the e^3*-value* framework does not provide a means to relate and integrate viewpoints. To this end, we propose to combine the e^3*-value* framework with feature–solution (FS) graphs to fill in this void. An FS-graph establishes relations between features (e.g. requirements) and solutions that realize these features [12,13]. They can be used to define a design space not only within a viewpoint, but also across viewpoints. In addition, viewpoint integration can also be used to trace design decisions and to explore design alternatives. The contributions of this paper are twofold:

- It presents a way to perform a thorough but fast exploration of e-commerce initiatives by combining two methods: the e^3*-value* framework that offers the necessary lightweight, multi-viewpoint approach, and FS-graphs that ensure systematic relating requirements to solutions, on business level as well as technical level. In this way, viewpoints are integrated and their effects on each other can be analysed.
- The proposed method introduces as a by-product a business-value driven information systems architecture for implementing the e-commerce initiative, if found feasible. The early exploration of supporting information systems is important for e-commerce tracks because information systems are a critical success factor in e-commerce.

In sum, the focus on this paper is how to deal with the wide range of stakeholders involved during the exploration of an e-commerce idea.

The remainder of this paper is organized as follows. We start with a short description of the project we consider in Section 2. The project concerns provisioning online news. In Section 3 we present the background for our method. Section 4 discusses the method using the online news project as an example. In Section 5 we discuss related work, and finally in Section 6 we draw conclusions and discuss future work.

2 Project: Online news

We exemplify our method by a project we carried out in the realm of provisioning of a value-added news service, which is discussed in [8]. A regular newspaper called (say) the *Amsterdam Times* wants to offer subscribers the service to read articles online using the internet, but with making hardly any costs. Therefore, the business idea is to finance this by the telephone connection revenues, which are paid by the reader who has to set up a telephone connection for internet connectivity. Multiple parties are involved in this e-commerce initiative. For brevity, we will discuss only two of them: the *Amsterdam Times* (newspaper) and the *Last Mile* (the local operator exploiting the last mile of copper wire between a subscriber's home and a telephone switch). The examples in the following sections are mostly related to the *Amsterdam Times*, unless otherwise mentioned, since it is the major party involved in this e-commerce initiative. The business idea can be achieved by two different value models: the *terminating* model and the *originating* model. In the terminating model, the reader pays the local operator a fee for a telephone connection. This fee is used to pay other parties; the local operator sets the price. In the originating model, the reader pays the newspaper directly, who pays other parties;the newspaper sets the price. An analysis of strategic pros and cons of both models for the various parties involved is given in [8].

3 Background for our method

Our method rests on two pillars: the *e*^{3}*-value* framework and the FS-graph. We first present them separately, and in the next section we explain how both methods are combined.

3.1 The e^3-value *framework*

Earlier work [8,9] presented the *e*^{3}*-value* framework, which is a lightweight approach for defining, deriving and analysing multi-enterprise relationships, business cases and requirements. The framework presents three stakeholder based viewpoints for e-commerce:

1. the *business value* viewpoint focuses on the (new) way of economic value creation, distribution and consumption in a multi-actor network
2. the *business process* viewpoint focuses on a way to put the value viewpoint into operation in terms of business processes

3 the *information system* viewpoint focuses on the information systems that enable and support e-commerce processes

The term *viewpoint* refers to a mechanism for dealing with multi-perspective problems. For a formal definition of this term, see the IEEE standard 1471-2000. The multi-perspective attitude is required since e-commerce tracks typically involve a wide range of stakeholders, from top management (e.g., CEO, CFO, and CIO) to IT personnel. The latter are important to bring in knowledge on application of new technological possibilities. By using a multi-viewpoint approach, we separate concerns of stakeholders to clarify discussions. As its name suggests, the business value viewpoint is of main interest for the top management stakeholders. It enables setting up a prediction of revenues and expenses, based on exchanges of valuable goods and services between multiple actors. The business process viewpoint focuses on operational fulfilment of business processes that enable the solutions for the requirements set on the business value viewpoint. The information system viewpoint is typically a concern of an IT department, but is crucial for our method since information systems are a critical success factor in e-commerce, and because information systems may require major investments and operational expenses. We will further use the term "three viewpoints", referring to the business value viewpoint, the business process viewpoint and the information system viewpoint.

3.2 The Feature–Solution graph

The Feature–Solution (FS) graph presented in [12,13] captures evolving knowledge about quality requirements and solution fragments, and is used to guide an iterative architecture development process. The FS-graph was introduced within the information system viewpoint, to connect quality requirements with solution fragments. The structure of this FS-graph resembles that of a goal hierarchy in goal-oriented requirements engineering (see for instance [14]). The FS-graph consists of a feature space and a solution space, and relations between elements of one space with elements of the other space. In this paper, the FS-graph is applied to all three viewpoints recognized in the e^3-*value* framework and it is applied across viewpoints as well to support convergence of viewpoints. We define the feature space and the solution space in the context of this work as follows:

- The *feature space* describes desired properties of an architectural viewpoint, as expressed by the stakeholders involved in this viewpoint.
- The *solution space* contains the internal decomposition of solutions for the requirements set by the stakeholders. In addition, the solution space may also contain general applicable solutions (e.g., business and architectural patterns).

Features as well as solutions can be decomposed in AND–(EX)OR decomposition trees. An AND decomposition of a node means that all its constituents must be available; an OR requires an arbitrary (0 or more) number of constituents; an EXOR requires precisely one constituent. Two types of relationships between elements of the feature space and elements of the solution space were presented in [12,13]:

1 A feature in the feature space can select a solution in the solution space, meaning that the selected solution is required (necessary) in order to meet the selecting requirement. We will further call this type of relationship a *required* selection.

2 A feature in the feature space can explicitly rule out a particular solution, meaning that if the selected solution is indeed implemented, the selecting requirement cannot be met. We will further call this type of relationship a *forbidden* selection.

The required and forbidden selection relations establish rather strong relations between features and solutions: if situation X is encountered, then (do not) apply solution Y. In many situations, the relationships between features and solutions are not that clearcut. In this study, we add two new relationships to the FS-graph to establish weaker relations:

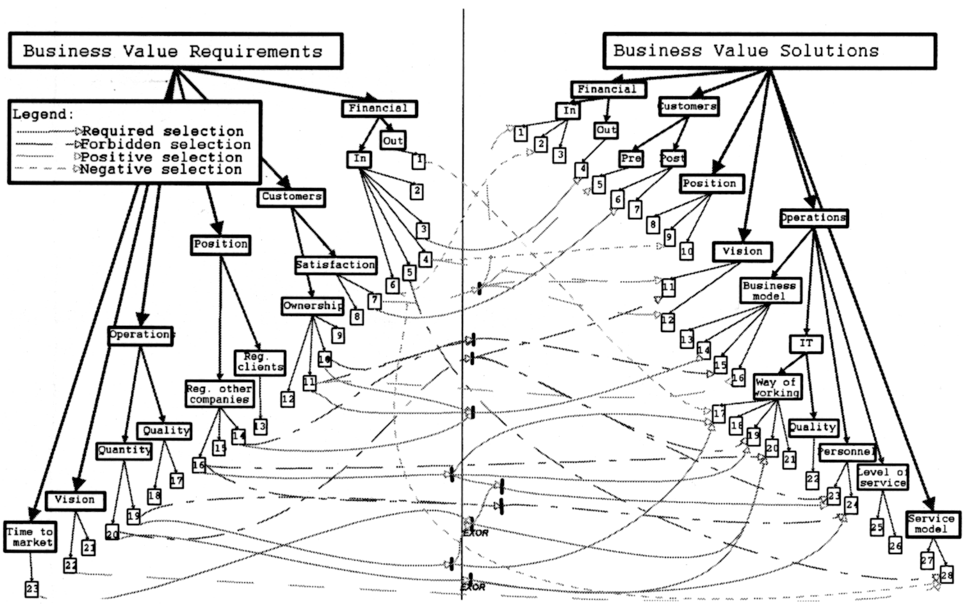
1 A solution in the solution space has a positive influence on meeting the selecting requirement in the feature space, but is not absolutely necessary in order to meet the requirement. We will further call this type of relationship a *positive* selection.

2 A solution in the solution space has a negative influence on meeting the selecting requirement in the feature space, but the requirement can still be met if this solution is implemented. We will further call this type of relationship a *negative* selection.

In order to make the reader familiar with the FS-graph terminology, Figure 1 presents a simple FS-graph. We use the following conventions in the graph:

- the various types of relationships between elements of the feature space and elements of the solution space are represented by different line styles, as shown in the figure
- a decomposition of a node is always of the type OR, unless otherwise mentioned with the labels AND or EXOR

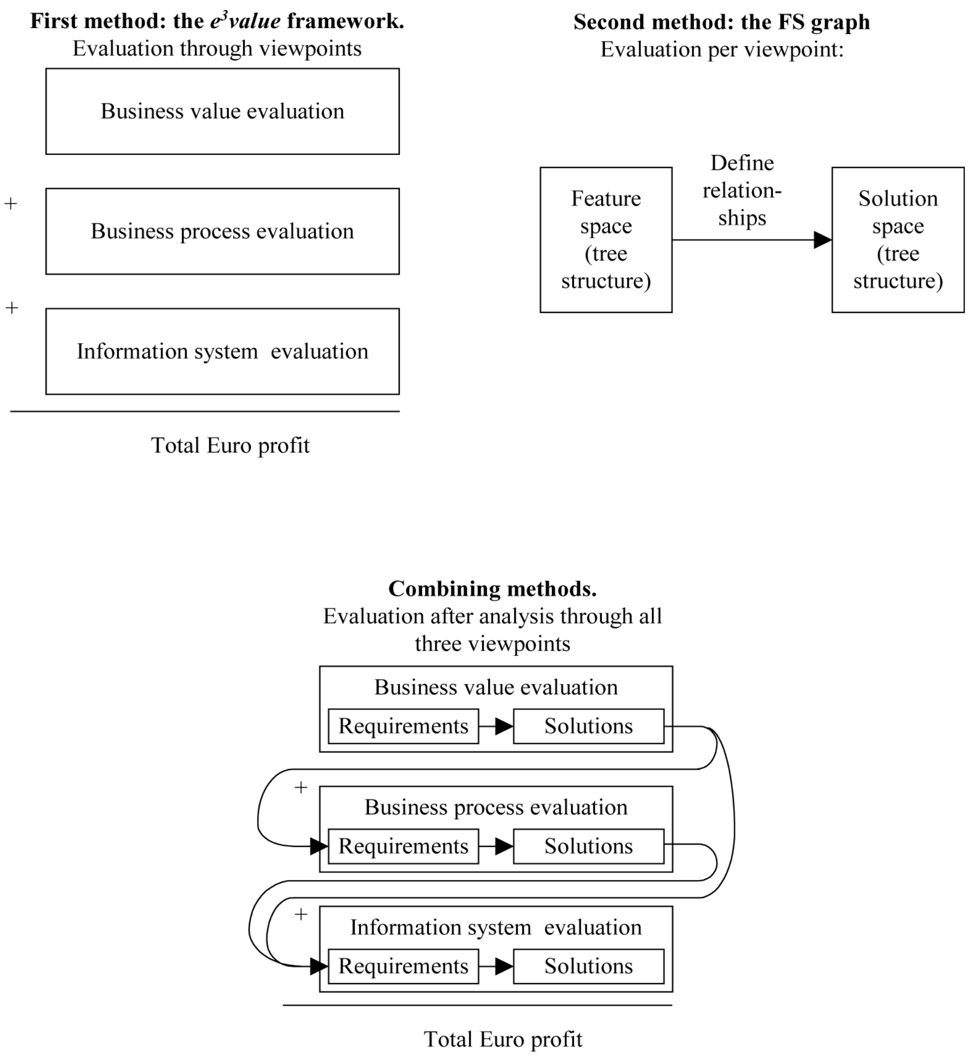
Figure 1 A partial FS-graph modeling the business value viewpoint



4 The method: business cases assessment

In this section we present our method, and demonstrate how to use it on a real-life project from the telecom sector, where the e^3-*value* framework was implemented successfully. Our method combines the e^3-*value* framework with FS-graphs; we define a feature space and a solution space and therefore an FS-graph for each viewpoint recognized in the e^3-*value* framework. We then interrelate the three FS-graphs, again using FS-graphs. The idea of combining both methods is presented in Figure 2. The term business case is central in this paper. A business case is defined as a subset of the three viewpoints FS-graphs, and it always includes both feature and solution fragments. Business cases therefore refer to possible ways to carry out a business idea (solutions), as well as to their justifications (requirements).

Figure 2 Outline of combining the e^3-*value* framework with FS-graphs. (a) The e^3-*value* framework. (b) The FS-graph. (c) The combination of the methods



As presented in [12,13], the FS-graph encapsulates knowledge about trade-offs between conflicting requirements. Generally speaking, conflicts arise from multiple viewpoints and concerns [15]. We use the knowledge that the FS-graph encapsulates to identify conflicts in stakeholders' concerns and analyse business cases. We spot the implications of conflicting demands, possibly adapt business cases and eventually choose the preferred business case to carry out the business idea. Given the e^3-*value* framework and FS-graphs, we propose a method that consists of five steps to analyse business cases. These steps are listed in a logical order, but sometimes iterations may be required.

1 defining three sets of feature and solution spaces, one per viewpoint

2 relating features and solutions within every viewpoint

3 interrelating features and solution between viewpoints

4 defining business cases through all viewpoints

Evaluating business cases using the e^3-*value* framework.

4.1 Step one: defining three sets of feature and solution spaces, one per viewpoint

4.1.1 Methodology

Step one includes defining requirements and solutions per viewpoint. Identifying, formulating and validating requirements is the traditional process of requirements engineering. The requirements are stated in the feature space of an FS-graph. Solutions are concrete measures to be taken and they are represented in an FS-graph's solution space. The feature space and the solution space can be split with AND–(EX)OR relationships into domains, which group logically related requirements and solutions. Examples of domains include financial, personnel, and operations.

4.1.2 Project example

The feature and solution spaces are split into requirement and solution domains. The domains, the requirements and the solutions apply for both the *Amsterdam Times* and the *Last Mile*. However, not both of these parties would set all of the requirements, and some requirements require different solutions when applied by various parties. We will later use the feature space domains of the business value viewpoint as criteria for defining business cases. These domains are therefore of prime importance in the whole process. Some important requirements and solutions are listed in Table 1. A more comprehensive view of the business value domains can be found in Figure 1. Other feature and solution spaces are discussed generally.

Note that one party can make decisions about certain solutions that are invisible for the other party. For example, whether one party hires more personnel to carry out a task, or outsources this task, has no influence on the second party, unless the second party is involved in the process.

Table 1 Important requirements and solutions on all three viewpoints

Business value viewpoint feature space	
Financial	Examples for requirements are 'low investment' (quantifiable) and 'generate as much phone traffic as possible'.
Position in the market	The two main types of entities with which a company interacts are the clients and other companies. We therefore split this domain into two sub domains: 'position regarding the client' and 'dependence on other companies'. The latter includes the requirement that the *Amsterdam Times* should minimize dependence on other entities.
Time-to-market	Time-to-market refers to the period of time between the beginning of designing the product or service, until the product or service is available for the customer. The underlying requirement is 'the service must be operational within three months'.
Business value viewpoint solution space	
Customers	Serve as many customers as possible (quantifiable); assure clients' demands are met.
Operations	Use the originating business model; use the terminating business model; the *Amsterdam Times* should outsource whole IT-aspect; hire substantial new personnel; use only existing personnel (plus at the most three new employees).
Business process viewpoint feature space	
Resources	Minimal number of persons required to carry out processes; clients can read articles at any time (24 hours a day, 7 days a week, 365 days a year).
Business process viewpoint solution space	
Personnel	Hire more than three new IT-specialists.
Operations	Link offline and online activities.
Information system viewpoint feature space	
Quality of service	The newspaper's offline activities create input for the online activities as well; serving a high number of customers simultaneously (quantifiable).
Information system viewpoint solution space	
Architecture	Use a three-tier architecture; client access to layers is concurrent.
Performance	Server farm with load balancing.
Security	Firewall, encryption, authentication server and out-sourcing the payment handling (e.g., to www.ibill.com).

4.2 Step two: relating features and solutions within every viewpoint

4.2.1 Methodology

In step two we link requirements to solutions on every viewpoint of the three FS-graphs. This process requires expert knowledge on the three viewpoints: representatives of the business have the knowledge to link business value and business process requirements to solutions; information systems experts have the knowledge required to link requirements to solutions on the information system viewpoint. This process has to be performed for every involved party.

4.2.2 Project example

The business value requirement 'generate as much phone traffic as possible' has a positive selection on the following solutions: 'serve as many customers as possible' or 'assure client demands are met'. The business process requirement that clients can read articles at any time has a positive selection on 'link offline and online solutions'. The performance-related information system requirement of serving a high number of customers simultaneously has a positive selection on the solution to integrate the BL and DM layers of the three-tier architecture [16] into one layer. The same requirement also selects other solutions, including the use of a server farm.

4.3 Step three: interrelating features and solutions between viewpoints

4.3.1 Methodology

In this step we introduce a new dimension, unique for the combination of the e^3-*value* framework and the FS-graph: viewpoints integration through interrelating viewpoints. In step two we linked the requirements of every viewpoint to solutions on the same viewpoint. Now we identify the implications of business value viewpoint solutions on requirements on both underlying viewpoints (business process and information system viewpoints). We also identify the implications of business process solutions on the underlying information system viewpoint. By doing this we express and enact relationships between viewpoints, and resolve conflicts when necessary. This inter-viewpoint communication is a key factor in achieving integration [17]. The same relationships that are used to link requirements and solutions within one viewpoint are used to link solutions of one viewpoint to requirements of underlying viewpoints: the required selection, the forbidden selection, the positive selection and the negative selection. Note that in this case solutions select requirements, instead of the other way around. This is because a solution identified at a particular viewpoint (abstraction level) may act as a requirement for subordinate viewpoints. Step three ensures that business principles guide the operational and technical aspects of a feasibility study. The product of this step is knowledge that enables analysing feasibility of business cases and generating a business value driven architecture for information systems that corresponds with a desired business case.

4.3.2 Project example

Examples of interrelating viewpoints are given in Table 2.

Table 2 Examples of interrelating viewpoints

Solution	*Selection type*	*Requirement*
Business value: serve as many customers as possible	Positive selection	Business process: clients can read articles at any time
Business value: assure clients' demands are met	Required selection	Information system: serving a high number of customers simultaneously
Business process: link offline and online activities	Positive selection	Information system: the newspaper's offline activities create input for the online activities

We also investigated the impact of a viewpoint on upper viewpoints, instead of underlying viewpoints, but no significant implications and relationships were identified for the project at hand.

4.4 Step four: defining business cases through all viewpoints

4.4.1 Methodology

In step four we define business cases. A business case is a subset of the FS-graphs corresponding with each viewpoint. As such, it refers to a possible way to carry out a business idea. This is where the value of interrelating viewpoint becomes visible. We define business cases and use the FS-graph to choose requirements and solutions, meanwhile identifying and handling conflicts. This step can be divided into two substeps: defining business cases on the business value viewpoint, and then working them out in all viewpoints.

4.4.1.1 Substep 1: defining business cases on the business value viewpoint

Our starting point is the product of steps one, two and three, which can be understood through Figure 2; the architectural space has been divided into chunks, which are interlinked. Business managers of the enterprises involved have to define business cases for their enterprises. Possible conflicts must be identified and handled throughout this process, and business cases of multiple enterprises must then be combined. Business cases are defined by prioritising the feature space (requirements) of the business value viewpoint, since business cases are driven by what the business wants to achieve. This process ends with multiple sets of requirements and requirement-domains of high priority. Each set forms the start of a business case. It implies choosing certain requirements on the business value viewpoint, which in their turn select certain solutions on the same viewpoint. This process may introduce conflicts, situations in which two or more requirements have contradicting relationships with the same solution. We handle conflicts based on their type:

- A *major conflict* involves a *required selection* and a *forbidden selection*. This business case is not feasible, and is either discarded or returned to the business for redefinition.
- A *minor conflict* involves a *positive selection* and a *negative selection*. This business case is feasible, but requires compromises. It is therefore returned to the business for redefinition.
- In conflicts where a *required selection* or a *forbidden selection* is involved, but not both of them, we analyse its impact, and classify it as a major one or as a minor one.

This way of handling conflicts is mentioned in [15]: conflicts are identified at the requirements level and characterized as differences at goal level; such differences are resolved (by business representatives) and then down propagated to the requirements level. This process is repeated iteratively, until no more business cases are returned to the business. At this point, we have identified for each enterprise a set of business cases that are feasible from the enterprise’s point of view. We proceed with combining business cases of multiple enterprises. For every combination we identify conflicts as done earlier, and determine whether a combination is feasible or not. In this way, we can quickly discard infeasible business cases and only elaborate the promising ones. This enables a fast exploration, a necessity in exploring e-commerce initiatives.

4.4.1.2 Substep 2: working business cases out through all three viewpoints

After having created a set of business cases on the business value viewpoint, we now use intra- and inter-viewpoint links to identify the requirements and the solutions that every business case implies on the two remaining viewpoints of the *e*^{3}*-value* framework. The business cases we defined in the business value viewpoint serve as a starting point for this process. This process is repeated for every business case, which has not yet been declared as not feasible. The product of this activity is a set of information system viewpoint requirements and solutions that fit into the various business cases. This set does not necessarily include all requirements and solutions to devise an information system architecture. Additional requirements and solutions are added to a business case as follows. Firstly, define information system viewpoint scenarios. This can be done in the same way business cases were defined in substep 1. Secondly, compare these scenarios with the sets of information system requirements that fit into the various business cases. Thirdly, decide which information system viewpoint scenario can be combined with which business case. Finally, add such a scenario to the business case. The solutions of an information system viewpoint scenario form fragments of an information systems architecture. If conflicts rise throughout this process, they should be handled as explained before. Step four forms the core of our method, and results in a set of business cases, worked out through all three viewpoints. Important remarks regarding this step are:

- While working out a business case through the various viewpoints, we may find out that although a business case was feasible on the business value viewpoint, it is not feasible on an underlying viewpoint (due to major conflicts). Such a business case is then being discarded.
- We start the process at the business value viewpoint, and work it out all the way down to the information system viewpoint, thereby ensuring that the choices made on the information system viewpoint and on the business process viewpoint are business value driven. The suggested information systems architecture, being a product of this process, is consequently business value driven as well.

4.4.2 Project example

4.4.2.1 Substep 1: defining business cases on the business value viewpoint

Short time to market, a fundamental requirement in e-commerce activities, is included in almost all business cases for the *Amsterdam Times*. Other criteria for business cases are: hiring new staff or insisting on not doing so, outsourcing services, customer ownership, maintaining little dependence on other parties in the market and the business model (originating/terminating). We defined seven business cases for the *Amsterdam Times*, with combinations of above domains and requirements, and two more for the *Last Mile*. Checking feasibility of the business cases yielded two interesting observations:

1. A short time to market and little dependence on other parties in the market cannot be achieved together.
2. The *Amsterdam Times* cannot assure a high level of customer ownership and yet be flexible on the business model, enabling not only the originating model but also the terminating one. This conflict rises since the terminating model means that other entities have contact with the customer, and not only the *Amsterdam Times*.

The domains and requirements relevant for business cases for the *Last Mile* are: business model (originating or terminating), customer ownership, and generating as many phone ticks as possible. These yield two business cases for the *Last Mile*: one that requires the terminating model, and one that enables both business models. The first one ensures customer ownership, the latter does not. Combining the scenarios of the *Amsterdam Times* and the *Last Mile* revealed that quite a few combinations are not feasible. Our feasibility control is sketched in Box 1.

Box 1 Combining business cases of the *Amsterdam Times* and the *Last Mile*.

At the beginning of this step we defined 14 combinations of business cases: seven business cases of the *Amsterdam Times*, with two of the *Last Mile*. After identifying conflicts, we divided the combinations to groups:

1. Five combinations of business cases were not feasible due to major conflicts; we discarded them. The conflict was: when the *Amsterdam Times* requires customer ownership, the originating business model must be applied, whereas the terminating model is required to assure the *Last Mile* customer ownership.
2. two combinations introduced no conflicts at all.
3. Seven combinations introduced conflicts that required differing levels of compromises and trade-offs.

One of the combinations in group two included a business case which was not desired but yet defined for the sake of comparison, leaving us with one combination that introduced no conflicts what so ever. Group three included three combinations with minor conflicts, of which one was discarded since it matched the business strategy to a lesser degree, and four combinations with more serious conflicts, which were discarded as well.

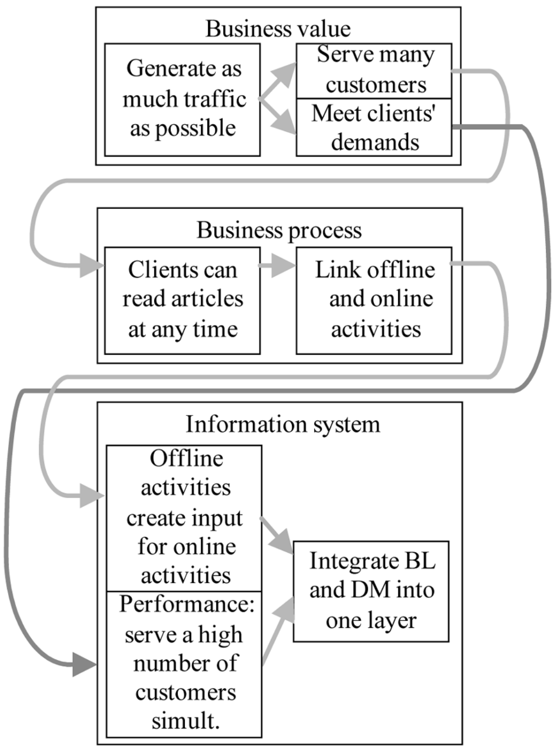
We ended this substep with three combinations of business cases, after having started with 14. Eleven business cases will not even be considered on underlying viewpoints, since they are not feasible on the business value viewpoint anyhow. This results in major saving of time, making our method a lightweight one.

4.4.2.2 Substep 2: working business cases out through all three viewpoints

The three remaining business cases form the input for substep 2. On the business process viewpoint it appeared that the *Last Mile* could choose any of the two scenarios, while this remains invisible for the *Amsterdam Times*. The information system viewpoint is relevant for the *Amsterdam Times* only, and not the *Last Mile*, which has a specific, limited involvement in the initiative. We defined three scenarios for the *Amsterdam Times* on the information system viewpoint: the first scenario requires normal performance and high availability; the second scenario requires high performance and normal availability and ensures no more than normal flexibility and low privacy; the third scenario is a “minimum scenario”: all non-functional requirements are set to ‘normal’, and not to 'high'. This substep revealed that all three combinations of business cases introduce conflicts on the information system viewpoint. Figure 3 presents such a conflict. In the given conflict, it was decided not to choose for the specified information system viewpoint solution, and to ensure high performance by other, stronger means. If conflicts cannot be resolved, the business case is discarded. This was not the case in our project. If all business cases are discarded, the e-commerce initiative is not feasible.

The information system solutions that are eventually chosen form fragments of an information system architecture. Requirements and solutions on the information system viewpoint are defined in levels, e.g., normal or high performance. By using the FS-graph, we can analyse whether it is possible to change the levels. In our project, a business case required normal performance, which implied two solutions: single server and medium bandwidth. It was also possible to require a high performance level. This would require two different solutions: a server farm and high bandwidth. Both requirements (normal or high performance) fit into the business case. However, their financial implications became clear only after having performed step five.

Figure 3 An example of a conflict on the information system viewpoint



4.5 Step five: evaluating business cases using the e[3]-value framework – methodology

In steps one through four, business cases were defined, put into operation and translated to fragments of a system level architecture. Those business cases that proved not to be feasible on any of the viewpoints were discarded, and trade-offs were made wherever necessary. The business cases that ‘survived’ the four steps must now be tested for economic feasibility. This is done using the e^3-*value* framework, as explained in [8,9]. The financial implications of changing the degree of non-functional requirements become obvious in this step: a server farm costs more than a single server. Since this step is an implementation of the e^3-*value* framework, and it presents no features related to combining this method with the FS-graph, we do not present here the calculations of economic feasibility for our project. The method and example calculations can be found in [8,9].

5 Related work

Empirical studies showed that poor understanding of the domain is a primary cause of project failure [18]. A deeper understanding about a domain requires understanding the interests, priorities and abilities of the various actors involved. We believe that by interrelating the three viewpoints of the e^3-*value* framework, the domain knowledge which businessmen have is directly related to technical implementation issues and vice versa. This way, systems are related to business, and business issues are translated to implementation in terms of information systems. Understanding the reasons that underlie established work patterns and practices is important when developing information systems. In [19], the work patterns are considered from a business process point of view. In contrast, we suggest focusing on the business value as the organizational environment of information systems, since business value stands above business processes. A Design Rationale (DR) [20,21] is a representation of the reasoning behind the design of an artefact. It is concerned with methods and representations for capturing why designers have made the decisions they have made. A well-known approach to representing design rationale is Design Space Analysis, whose notation is called QOC (Questions, Options and Criteria) notation [22]. Questions in QOC are key design issues, and Options are possible answers to the Questions. The solution fragments of the FS-graph contain both Questions and Options. Criteria in QOC are used to choose between Options. They resemble the requirements as captured in the FS-graph. Options and Criteria in QOC are linked by Arguments such as *supports* or *objects to* which resemble the links between the requirements and solution part of the FS-graph. DR is most often used as a tool in the design process, especially the user-interface design process, to augment design reasoning, and to help in formulating and communicating arguments. It usually pertains to one particular set of choices, not to a complete space of design options, as we try to capture in the FS-graph. In [23], design spaces are used to map requirements to components. The approach is geared towards the reuse of components within a domain, and does not handle conflicts. In Feature-Oriented Domain Analysis (FODA) [24,25], and variants thereof, a family or product line is represented in a feature tree. Features can be mandatory, alternative, or optional. A specific product is then composed by choosing a set of alternative and optional features; these express the variabilities within the product line. The feature tree may span the current set of products, or a design space of possible products, or a mixture. The feature tree thus resembles the feature space of the FS-graph. Usually though, features of a product line are units of functionality rather than quality concerns. In [14], goal-oriented requirements engineering techniques are discussed that complement and strengthen traditional requirements analysis techniques by offering a means for capturing and evaluating alternative ways of meeting business goals. The goals presented in this work are functional goals. In our work, we concentrate on goals derived from the business strategy. These are strategic goals, and not functional (operational) ones. We then present a value-driven analysis instead of a goal-driven one: achieving the business goals adds business value. The added value of one business within a value constellation is the justification for the business' existence. For that reason, we concentrate on business value. In [14], goals are decomposed based on their type: those that can be delineated clearly (goals), and those that cannot (soft goals). We suggest a stakeholder-based distinction between requirements (or goals). The stakeholder centric *e*^{3}*-value* framework ensures that the needs of multiple stakeholders are addressed. Kavakli *et al.*, focus on the relationship between enterprise goals and information system requirements [26]. They consider operational issues as part of Enterprise Modeling, which is also concerned with the business goals. We separate these two issues into two viewpoints: business value and business process. Different stakeholders may have differing concerns in those two viewpoints. Like Kavakli *et al.*, also we do not use business cases (or scenarios, in their terms) only for goal identification, but also for considering operational aspects.

6 Conclusions and future work

We have presented a method for a fast and thorough exploration of feasibility and commercial viability of e-commerce initiatives. This method combines two earlier presented methods. To this end, we have proposed first investigating multiple stakeholder viewpoints, thereby supporting separation of concerns, and then integrating these viewpoints. Our method recognizes the fact that exploring an e-commerce initiative involves strategic business issues, operational business issues and technical issues, and even more important, that these issues must remain related to each other. The method can be used to define and investigate business cases. Economic viability is checked only after a business case has proven to be value adding for all involved parties, and feasible, both operationally and technically. This is already checked at an early stage, thereby avoiding unnecessary work and enabling a fast process. We recognize two main areas of further research, aimed at easing and improving the use of our method: automating processes and visualizing information. Some parts of the method can be automated, thereby saving valuable time during the exploration of initiatives. This includes identifying conflicts, i.e., when two or more requirements have contradicting relationships with the same solution. Conflict handling is much harder to automate, and often requires human intervention. A drawback of the three-viewpoints FS-graph is its size and complexity, which makes it practically impossible to grasp the whole picture at a glance. We are currently investigating abstraction mechanisms for hiding complexity in order to focus on those parts of an FS-graph that require attention.

References and Notes

1 Shama, A. (2001) 'Dot-coms' coma', *The Journal of Systems and Software*, Vol. 56, No. 1, pp. 101–104.

2 Gordijn, J. (2002) *Value-based Requirements Engineering – Exploring Innovative e-Commerce Ideas*, Ph.D. Thesis, Vrije Universiteit, Amsterdam, NL, Also available from http://www.cs.vu.nl/˜gordijn/.

3 Porter, M.E. (1985) Competitive Advantage – Creating and Sustaining Superior Performance, Free Press, New York, NY.

4 Tapscott, D., Ticoll, D. and Lowy, A. (2000) *Digital Capital – Harnessing the Power of Business Webs*, Nicholas Brealy Publishing, London, UK.

5 Horngren, C.T. and Foster, G. (1987) *Cost Accounting: A Managerial Emphasis*, 6th ed., Prentice-Hall, Englewood Cliffs, NJ.

6 Rumbaugh, J., Jacobson, I. and Booch, G. (1999) *The Unified Modelling Language Reference Manual*, Addison Wesley Longmann, Inc., Reading, MA.

7 Yu, E.S.K. and Mylopoulos, J. (1998) 'Why goal-oriented requirements engineering', in Dubois, E., Opdahl, A.L. and Pohl, K. (Eds.): *Proceedings of the 4th International Workshop on Requirements Engineering: Foundation for Software Quality (RESFQ 1998)*, Namur, B. Presses Universitaires de Namur.

8 Gordijn, J. and Akkermans, J.M. (2001) 'Designing and evaluating e-business models', *IEEE Intelligent Systems – Intelligent e-Business*, Vol. 16, No. 4, pp. 11–17.

9 Gordijn, J., Akkermans, J.M. and van Vliet, J.C. (2000) 'What's in an electronic business model', in Dieng, R. and Corb, O. (Eds.): *Knowledge Engineering and Knowledge Management – Methods, Models, and Tools, 12th International Conference (EKAW 2000)*, LNAI, Vol. 1937, pp. 257–273, Berlin, D. Springer Verlag, Also available from http://www.cs.vu.nl/~gordijn/.

10 Holbrook, M.B. (1999) Consumer Value: A Framework for Analysis and Research, Routledge, New York, NY.

11 Sommerville, I., Sawyer, P. and Viller, S. (1998) 'Viewpoints for requirements elicitation: A practical approach', *Proceedings of the Third IEEE Internation Conference on Requirements Engineering (ICRE 98)*, Colorado Springs, IEEE CS Press, Los Alamitos, CA, pp. 74–81.

12 de Bruin, H. and van Vliet, H. (2001) 'Scenario-based generation and evaluation of software architectures', in Bosch, J. (Ed.): *Proceedings of the Third Symposium on Generative and Component-Based Software Engineering (GCSE'2001)*, Erfurt, Germany, Lecture Notes in Computer Science (LNCS), Vol. 2186, pp. 128–139, Springer-Verlag, Berlin, Germany.

13 de Bruin, H. and van Vliet, H. (2002) 'Top-down composition of software architectures', in Runeson, P. (Ed.): *Proceedings of 9th International Conference and Workshop on the Engineering of Computer-Based Systems (ECBS'2002)*, Lund, Sweden, pp. 1–10.

14 Mylopoulos, J., Chung, L., Liao, S., Wang, H. and Yu, E. (2001) 'Exploring alternatives during requirements analysis', *IEEE Software*, Vol. 18, No. 1, pp. 92–96.

15 Lamsweerde, A.V. (2000) 'Requirements engineering in the year '00: A research perspective', *Conference Proceedings ICSE'00*, pp. 5–19, Limerick, Ireland, ACM.

16 The three-tier architecture includes three layers: the User Interface layer (UI), the Business Logic layer (BL) and the Data Management layer (DM).

17 Nuseibeh, B., Kramer, J. and Finkelstein, A. (1996) 'A framework for expressing the relationships between multiple views in requirements specifications', *IEEE Transactions on Software Engineering*, Vol. 20, No. 10, pp. 760–773.

18 Yu, E. (1997) 'Towards modelling and reasoning support for early-phase requirements engineering', *Proceedings of 3rd IEEE International Symposium of Requirements Engineering (RE'97)*.

19 Yu, E. and Mylopoulos, J. (1993) 'An actor dependency model of organizational work; with application to business process engineering', *Proceedings of the Conference on Organizational Computing Systems (COOCS'93)*.

20 Buckingham Shum, S. and Hammond, N. (1994) 'Argumentation-based design rationale: What use at what cost?', *International Journal of Man–Machine Studies*, Vol. 40, No. 4, pp. 603–652.

21 Moran, T. and Carroll, J. (Eds.) (1994) *Design Rationale: Concepts, Techniques, and Use*, Lawrence Erlbaum Associates, Hillsdale, New Jersey.

22 MacLean, A., Young, R.M., Bellotti, V.M. and Moran, T.P. (1991) 'Questions, options and criteria: Elements of design space analysis', *Human–Computer Interaction*, Vol. 6, No. 3&4, pp. 201–250.

23 Baum, L., Becker, M., Geyer, L. and Molter, G. (2000) 'Mapping requirements to reusable components using design spaces', *Proceedings 4th International Conference on Requirements Engineering*, IEEE, pp. 159–167.

24 Czarnecki, K. and Eisenecker, U.W. (2000) *Generative Programming: Methods, Tools, and Applications*, Addison-Wesley, Reading, Massachusetts.

25 Kang, K., Cohen, S., Hess, J., Novak, W. and Peterson, S. (1990) *Feature-Oriented Domain Analysis (FODA) Feasibility Study*, Technical Report, Software Engineering Institute.

26 Kavalki, E., Loucopoulos, P. and Filippidou, D. (1996) 'Using scenarios to systematically support goal-directed elaboration for information system requirements', *Proceedings of 9th International Conference and Workshop on the Engineering of Computer-Based Systems (ECBS'96)*, Friedrichshafen, Germany.