# Scenario Methods for Viewpoint Integration in e-Business Requirements Engineering


**Jaap Gordijn**
Vrije Universiteit - Vuture.net
De Boelelaan 1081a
1081 HV Amsterdam
The Netherlands
gordijn@cs.vu.nl

**Hans de Bruin**
Vrije Universiteit
Faculty of Sciences
IMSE Department
hansdb@cs.vu.nl

**Hans Akkermans**
Vrije Universiteit - Vuture.net
IMSE Department
AKMC Knowledge Management
HansAkkermans@cs.vu.nl



## Abstract

*Before it makes sense to embark upon e-commerce systems development, first the commercial and technical feasibility of an e-business idea must be established. To this end, we describe how needs and interests of various types of stakeholders can be expressed by different viewpoint models. We propose an extension of so-called use case maps (UCMs) to e-business requirements engineering, as a scenario method to achieve the necessary viewpoint integration. Furthermore, we show how this scenario method is employed in an iteratively progressing 'spiral' process of e-business requirements elicitation and analysis. Our approach is practically illustrated by an e-business case study in electronic advertising.*


## 1 Introduction

Innovative e-business projects are characterized by short but intensive system development efforts, due to the required quick time to market. In a short time frame, not only a system design, but also a new e-business idea has to be designed. To enable a fast and focused development track, it is important for stakeholders to build up confidence in the feasibility of the business idea as soon as possible. In particular, the following questions should be answered:

1. Is the e-business idea at hand expected to be profitable for each actor involved?
2. Are the supporting e-business information systems technically feasible?

If, and only if, these questions can be answered positively, it is worthwhile to explore the requirements of a prospective system more in depth.

To answer these questions, a quick and broad overview of the whole e-system, that is, the business idea in terms of services offered to customers *and* the supporting information systems, must be created. Therefore, in the first phase of an innovative e-business development project, business as well as system-oriented stakeholders should be involved. To build and jusitify confidence in an e-business idea for this wide range of stakeholders, we propose an approach based on multi-viewpoint requirements engineering [9, 12, 8] in order to quickly assess the technical and the business feasibility.

This approach exploits the mechanism of *separation of concerns*. This is an aspect of special importance in e-business projects, because requirement discussions with business-oriented stakeholders (typically CxOs) and system-oriented stakeholders (typically IT departments) tend to interfere with each other, leading to unclear, troubled, and time-consuming decisions. Separation of concerns is a well-known and valid principle in IT development, but it can easily lead to lacking integration between the viewpoints of various stakeholders. Therefore, it is important that relations between viewpoints are made visible and traceable.

In this paper, we present an approach to achieve this. We propose a scenario method, based on an extension of use case maps (UCMs) [1], to express, relate and integrate different stakeholder viewpoints in an iterative requirements process. For each viewpoint we develop the *same* set of scenarios, expressed by *different* UCMs tied to that particular viewpoint. By developing the same scenarios for each viewpoint, different requirement viewpoint models emerge as a single integrated and traceable set of requirements. This scenario-based multi-viewpoint approach is part of our $e^3$-*value*$^{TM}$ method for developing innovative e-business information systems [5, 4, 3].

The motivation to develop scenarios is to build better understanding of and confidence in the viability of an e-business idea. We do so by presenting *'profit sheets'* and *'cost sheets'* for each actor that are derived from the mentioned scenarios. The major objective of these sheets is to justify stakeholder confidence in the commercial and technical feasibility of a business idea, and not so much to obtain precise estimates of expected benefits. In fact, in the early requirements stages of innovative e-business projects, the former is much more important (and realistic) than the latter.

This paper introduces in Sec. 2 our $e^3$-*value* framework, presents three important e-business viewpoints, and gives an outline of the UCM scenario method. In Sec. 3, we describe our approach and illustrate it by one of the practical e-business projects we have been carrying out. Sec. 4 presents conclusions and lessons learnt.

# 2 The $e^3$-*value* Framework

## 2.1 Viewpoints in Innovative e-Business Requirements Engineering

To facilitate separation of concerns, we distinguish three requirement viewpoints, which are based on important groups of stakeholders that participate in the requirements engineering process. We only briefly discuss these viewpoints, as a more detailed report has appeared in [5].

**Business Value Viewpoint.** The business value viewpoint models a new, *innovative* way of doing business in terms of values exchanged between actors. To represent this viewpoint, in previous work we have proposed a *value-oriented modelling* technique, see [5, 4, 3] and Sec. 3.3.1.

**Business Process Viewpoint.** The business process viewpoint shows the operational fullfillment of the business value viewpoint by means of processes and workflows. To represent the business process viewpoint a number of existing techniques are suitable; in this paper we use Ould's role-based process-modelling technique [10].

**System Architecture Viewpoint.** The system architecture viewpoint shows how the requirements captured in the business-value and business-process viewpoints can actually be realized in an information system. For our purposes, a system architecture will at this stage be primarily developed to assess both the technical and commercial feasibility of a business idea. Once feasibility has been demonstrated from a business and system architecture viewpoint, the system architecture can be further elaborated.

## 2.2 Iterations in e-Business Requirements Engineering

A potential pitfall in innovative e-business projects is to follow a strict sequential 'waterfall' track of developing the requirement viewpoints in detail. This may result in a complete fine-grained business value model that however is impossible to realize in the given business process and system architecture environment. What is therefore needed initially, is a coherent set of *global* requirements for an e-business system in which stakeholders have justifiable confidence that it is feasible to implement it, and that it has an underlying business value model showing benefits to all actors involved. Only if stakeholders agree on such a set of requirements, it makes sense to further detail the requirement viewpoints so that implementation work can commence. For all the aforementioned viewpoints we develop (an extended version of) so-called use case maps based on the same set of scenarios. If these scenarios are all supported by the considered viewpoint, the business case is considered to be technologically viable with respect to *that* viewpoint. At the same time, these scenarios are used to enhance our insight in the profitability for each actor, by drafting a profit or cost sheet per viewpoint for each actor. Such a first requirements engineering cycle should result in confidence that the innovative e-business idea is technologically feasible, and that the idea is commercially interesting for each actor in the business value model.

## 2.3 Scenarios and Use Case Maps

A UCM is a visual notation to be used by humans to understand the behavior of a system at a high level of abstraction [1]. It is a scenario-based approach intended to explicate cause-effect relationships by travelling over paths through a system.

The basic UCM notation is very simple, and consists of three basic elements: responsibilities, paths and components. The term component should be interpreted in a broad sense: it may be a software component, but it can also represent a human actor or a hardware system. A simple UCM exemplifying the basic elements is shown in Fig. 1. A path is executed as a result of the receipt of an external stimulus. Imagine that an execution pointer is now placed on the start position (bullet at the top). Next, the pointer moves along the indicated scenario path, thereby entering and leaving components, and touching responsibility points. A responsibility point represents a place where the state of a system is affected or interrogated. The effect of touching a responsibility point is not defined in the UCM itself since the concept of state is not part of a UCM; typically, this effect is described in natural language. Finally, the end position is reached (stroke perpendicular to the scenario path) and the

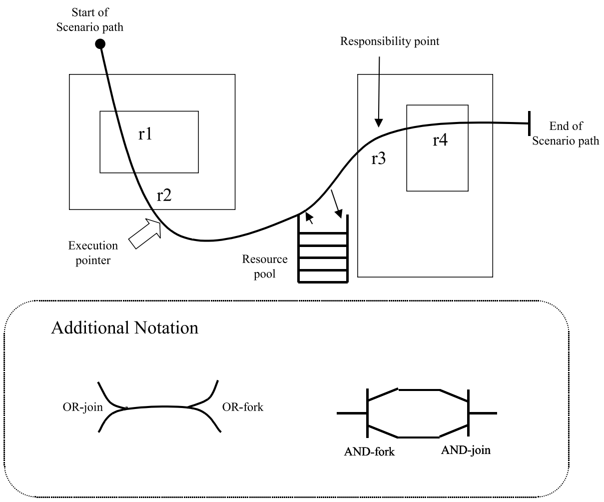


**Figure 1. UCM constructs.**

pointer is removed from the diagram.

In the same Fig. 1, two frequently used UCM constructs are shown. The AND construct is used to spawn (AND-fork) and synchronize (AND-join) multiple activities along parallel scenario paths. The OR construct is a means to express multiple independent scenario paths, which share identical sub-paths, within a single diagram.

Furthermore, the UCM notation supports resources by means of pools. A scenario in progress can only remove a resource from a pool if one or more resources are in it, otherwise the scenario will wait until a resource has been stored into the pool by another scenario.

# 3 Viewpoints and Scenarios Illustrated by an e-Business Case Study

## 3.1 The Initial e-Business Idea

The Ad Association is a company that co-ordinates more than 150 local, world-wide located, free ad papers called FAPs. FAPs independently produce (non-electronic) papers with ads and serve a geographical region. The handling of ads is as follows. A customer submits an ad to a FAP. The FAP checks the ad (e.g. for absence of dirty language and for style) and places the ad in its next issue. It is possible to place an international ad. In this case, the FAP to which the ad was submitted distributes the ad to the Ad Association, which redistributes the ad to other FAPs (serving different geographical regions). These other papers publish the ad as soon as possible. In a new e-business idea, the Ad Association and FAPs want to exploit their local established brand names to develop an internationally, Internet based, contact ad service. The following sections show for this business idea the first iteration in exploring the aforementioned viewpoints to build confidence in commercial and technical feasibility. We construct one business value model and a corresponding business process model. Subsequently, we discuss two software architectures that both realize the given business value and process model.

## 3.2 Scenarios

The first step after a statement of a business idea is to outline the value-added services to be offered. This step can lead to multiple, alternative, sets of services. These services are explored with the help of scenarios, which we use throughout the design of our viewpoints. A possible set of scenarios for the business idea at hand is:

- A contact searcher submits an ad to a FAP, and gets a possible contact in return. The latter means that an ad submission increases the chance for a contact searcher to find a contact s/he likes.
- A contact searcher queries for an ad on a website of a FAP, reads an ad, and pays a fee to the FAP.
- The Ad Association redistributes ads from FAPs to other FAPs, pays the originating FAP a fee, and gets paid by the FAPs who receive the ad.

Many other sets of services are possible, for instance a set where FAPs exchange ads on a bilateral basis (without the Ad Association). However, due to lack of space we only conside the former set of offerings.

## 3.3 Business Value Viewpoint

After an initial identification of scenarios, the next step is to design the business value model. Besides reaching understanding between stakeholders on a first draft of the way of doing business, a goal of developing the business value viewpoint is to discover services, which could not be discovered by making a first set of scenarios.

### 3.3.1 Business Value Model Constructs

Below, we briefly discuss the constructs which should be present in a business value model; an extensive discussion is given in [5, 6]. A general reference model for value-oriented e-business models is depicted in Fig. 2 and the main concepts that occur in it are summarized below.

An **actor** is perceived by its environment as an *independent economic* (and often also legal) entity. By performing *value activities* (see below) actors add value, either for themselves or for others. In a sound, viable, business model *every* actor is capable of adding value.

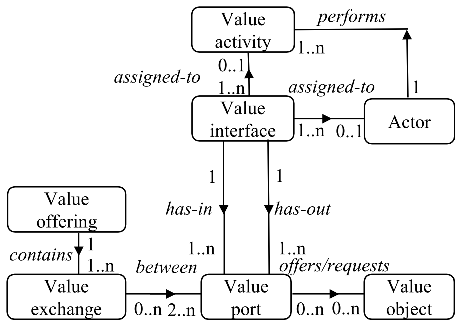


**Figure 2. Value-based reference model for e-business models.**

A **value activity** is *performed by* an actor to produce objects of value (outputs) by adding value to other objects of value (inputs).

A **value object** is a service, thing, or consumer experience that is of value to one or more actors.

A **value port** is a connector, which interconnects actors or value activities in a component-based way. Value ports *offer* or *request* value objects.

A **value interface** is made up of one or more value ports, and models the offering of an actor or value activity to its environment. It shows the value objects an actor is willing to *exchange* in return for other value objects via its ports. Its working is based on the principle "one good turn deserves another": a value object always has to be exchanged in return for another value object.

A **value exchange** represents the trade of a value object between value ports. It shows which actors are willing to exchange objects of value with what other actors.

A **value offering** consists of a set of value exchanges. Whereas value exchanges connect value ports, value offerings connect value interfaces of actors. The principle of "one good turn deserves another" is also valid for the value offering. For each value exchange, an 'inverse' value exchange should also be present, modelling the counterpart of an offering by the connected actor.

#### 3.3.2 Business Value Model

Using the forementioned concepts, we present a business value model (Fig. 3) for the business idea presented in Sec. 3.1. For brevity, the model also shows the value scenarios (see Sec. 3.3.3). Note that in this viewpoint we only model the exchange of objects that are of *value* to someone and not objects that result as I/O from a work activity (e.g. sending an invoice), which are modelled in the business process view. Also note that the business value model introduces a new service, which is expected to be commercially viable: *checking an ad*. We mention in passing that this service was not present in the first set of scenarios. It was identified by stakeholders later in the project, because they were forced to think about value adding activities.

#### 3.3.3 Business Value Scenarios

Fig. 3 shows the UCM paths for each of the three scenarios identified above, plus the checking scenario. A UCM path is constructed by concatenating value offerings that cause each other.

Our value scenarios differ from the UCM method specified in [1]. The main difference is that Buhr supposes a time-ordering on a scenario path, while we only assume causal relations. A business value model only states what is exchanged for what; no time-ordering is assumed. Our experience is that time-ordering issues tend to give a wrong focus in the discussion of the e-business model; stakeholders are primarily interested in *who* is doing *what* for *whom* and in the resulting revenues and costs. Because our scenario paths do not indicate time-ordering we have no scenario start- and endpoint as is normally the case in UCMs, but we have scenario *delimiters* instead.

For the scenario *submit ad*, paths 1 and 2 model a submitted ad, which is accepted and published on one or more website of FAPs, while paths 3 and 4 model a submitted ad which is rejected. Note that a rejected ad does not result in *possible-contact* nor *ad-placement* value offerings. The only offering that occurs is the *check ad* offering.

On scenario paths, *responsibility points* are superimposed. We use these points to model changes in the *profit sheet* of an actor as a result of executing a scenario path. Changes in a profit sheet are caused by exchanges of values between actors via their value interfaces. Therefore, value interfaces are responsibility points by definition in a value model. If we can estimate the number of times a scenario path is executed, and we have all possible paths, we have a basic idea about the profitability of the business idea for a specific actor.

#### 3.3.4 Profitability

For the scenario *submit ad* we derive a profit sheet for $FAP_i$ (Table 1), which is constructed as follows.

1. **Make a list of values entering and leaving the actor.** By following the scenario paths of a scenario, a list is constructed consisting of all objects of value entering or leaving the actor.

2. **Remove the value-neutral in and out values.** Some sets of value objects, which enter or leave an actor, and which are hard to quantify, are *value-neutral* for an actor. This means that after the execution of that path,

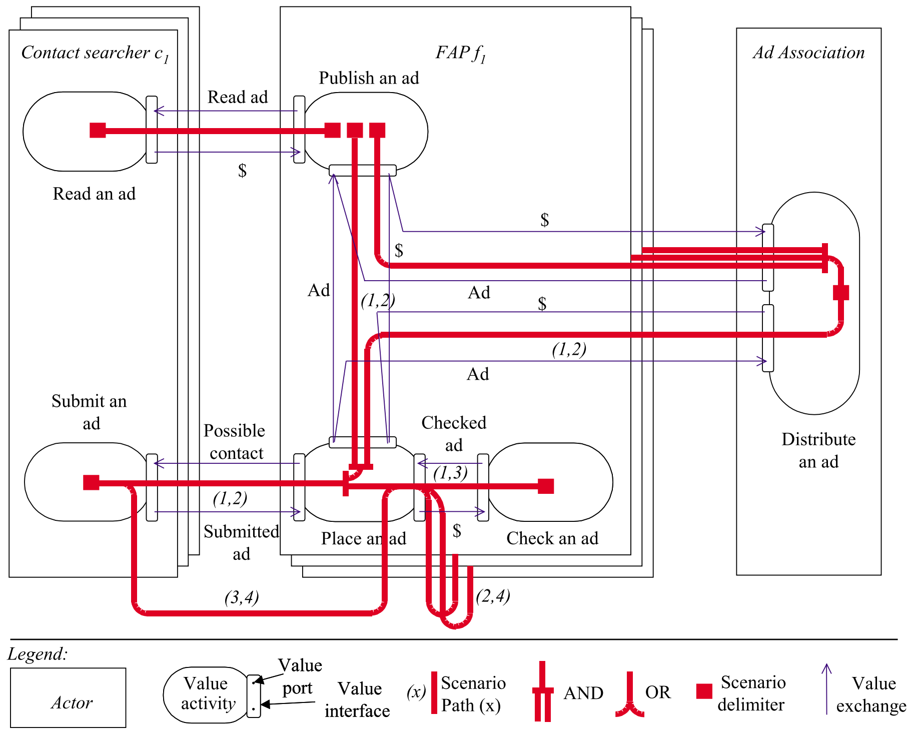


**Figure 3. Business value model with UCMs for the Ad Association business idea.**

the total value of objects in such a set is zero for an actor. This is for example the case if a value object enters the actor and leaves the actor in the same scenario path. For example, consider the first scenario path. The *submitted ad*, *checked ad*, *possible contact*, $ad_{to-own-publish-activity}$, and $ad_{to-Ad-Association}$ are value-neutral for actor $FAP_i$ because when the scenario has been executed, the actor has fulfilled its obligations, that is ensuring that an ad is published, and this actor cannot use the ad anymore (e.g. resell it in another scenario path), because it has been already sold to all possible parties.

3. **Estimate the value of the remaining value objects.** The value of the remaining objects is expressed in monetary units. For value objects representing a money-based payment, this is trivial, but valuing other objects (such as the value of a possible contact for a contact searcher) is more complicated, and involves different qualities and value dimensions. In [7], a general approach is described for valuing such objects; in [4], we show an application of this approach to digital content in a real-life e-business situation.

4. **Calculate the profitability of scenario paths.** Then, the profitability for each scenario path is calculated by totalizing the values of all objects entering the actor and subtracting the values of objects leaving the actor in that scenario path.

5. **Estimate the likelihood of scenario paths.** Based on estimations or previous experiences, the likelyhood of occurence for each scenario path of a specific scenario is specified.

6. **Calculate the expected profitability of a scenario path.** The expected value of each scenario path is calculated by multiplying the likelyhood of the occurence of the path with the profitability of the path.

7. **Calculate the expected profitability of a scenario.** Finally, we totalize the expected profitability of each scenario path in the scenario.

If we fill-in the fees in Table 1, we get a first impression of the profitability of the business-idea. Moreover, Table 1 can be used to perform a sensitivity analysis of the profitability, for instance during a workshop with actors about the value model. However, for a more overall view on the profitability, additional cost sheets for the business process and information system requirements have to be developed.

### 3.4 Business Process Viewpoint

The e-business process viewpoint illustrates processes to be carried out by actors, and messages interchanged between those actors, on a conceptual level. Because we gain

**Table 1. Profit sheet for $FAP_i$ for the scenario submit ad (Business Value viewpoint)**

| | |
|---|---|
| *Viewpoint* | Business value |
| *Actor* | $FAP_i$ |
| *Scenario* | Submit ad |
| *Scenario path* | *Profit* |
| 1 (60%) | $s_1 = fee_{distr_{AdAssocation}}$ |
| 2 (20%) | $s_2 = fee_{distr_{AdAssociation}} - fee_{check_{FAP_{other}}}$ |
| 3 (15%) | $s_3 = 0$ |
| 4 (5%) | $s_4 = -fee_{check_{FAP_{other}}}$ |
| Expected profit | $p = 0.6 * s_1 + 0.2 * s_2 + 0.15 * s_3 + 0.05 * s_4$ |

more insight in *how* processes, necessary to create value, are carried out, it is possible to identify major operational costs such as costs caused by persons carrying out work. Responsibility points indicate such costs.

#### 3.4.1 Business Process Model

A number of techniques have been developed to model business processes, such as UML activity diagrams with swimlanes to represent actors [11], or role-based process modelling techniques [10]. In this paper, we choose for the latter. Ould defines a *role* as a set of activities that are carried out by an actor in an organization. An *activity* is what actors do in their roles. Between activities and thus between roles *interactions* can occur.

Fig. 4 shows a process model which explains *how* the business value model is carried out by actors. We do not show the interactions explicitly to prevent unneccessary cluttering of the diagram. Interactions are implicitly shown by the UCMs.

There are no strict rules to map the business value model onto a process model because they express different viewpoints. There are, however, informal guidelines to derive a process model from an business value model. Value activities are mapped onto roles. Value exchanges are candidates for interactions between roles. However, value exchanges are not the same as interactions. Value exchanges denote things of value to (other) actors which do not always result in interactions between actors directly. Also, new interactions may be introduced that do not have a counterpart in value exchanges. For example, value exchanges regarding payments between value activities performed by the same FAP need not have a counterpart in interactions. The query asked by a contact searcher to a FAP, and the ad to be checked, are new interactions, which are not represented in the business value model.

#### 3.4.2 Business Process Scenarios

In a business process model, an UCM scenario path shows the time-sequence of messages and activities performed for a specific scenario. The same scenarios as in the business value model are shown, however the paths now show a sequence of interactions between roles. Note the synchronization bar (with the N:1 indication) in the *distribute ad*, the *place ad* and the *publish ad* role. Such a bar 'collects' a number of ads, say 100, and then continues the scenario with one payment for all these 100 ads This refers to the mechanism of aggregate payment [2]; it is much cheaper to handle one big payment rather than a large number of small ones.

Responsibility points indicate substantial operational costs, for instance caused by personnel. Selecting a capable checker, checking an ad, and administrating payments (received payments and payments done) are all tasks where humans are involved. These points are used to fill in the cost sheet for $FAP_i$ (Table 2). Based on estimations of the occurence of the scenario paths, the expected costs for the entire scenario is calculated, analogue to the process described in Sec. 3.3.4.

Note that, although we use the *same* scenarios in all our viewpoints, the scenario paths may *differ* in structure as well as in number for each viewpoint. This is caused by the different modelling perspectives of requirement viewpoints. In the business process viewpoint, scenario paths show similarities with paths for the business value model and are defined by concatenating interactions, which are numbered in Figure 4.

**Table 2. Cost sheet for $FAP_i$ for the scenario submit ad (Business Process viewpoint)**

| | |
|---|---|
| *Viewpoint* | Business process |
| *Actor* | $FAP_i$ |
| *Scenario* | Submit ad |
| *Scenario path* | *Costs* |
| 1 (60%) | $c_1 =$ *select-costs* $+$ *check-costs* $+$ *admin-costs*$/N$ |
| 2 (20%) | $c_2 =$ *select-costs* $+ (2 *$ *admin-costs*$)/N$ |
| 3 (15%) | $c_3 =$ *select-costs* $+$ *check-costs* |
| 4 (5%) | $c_4 =$ *select-costs* $+$ *admin-costs*$/N$ |
| Expected costs | $c_{business-process} = 0.6 * c_1 + 0.2 * c_2 + 0.15 * c_3 + 0.05 * c_4$ |

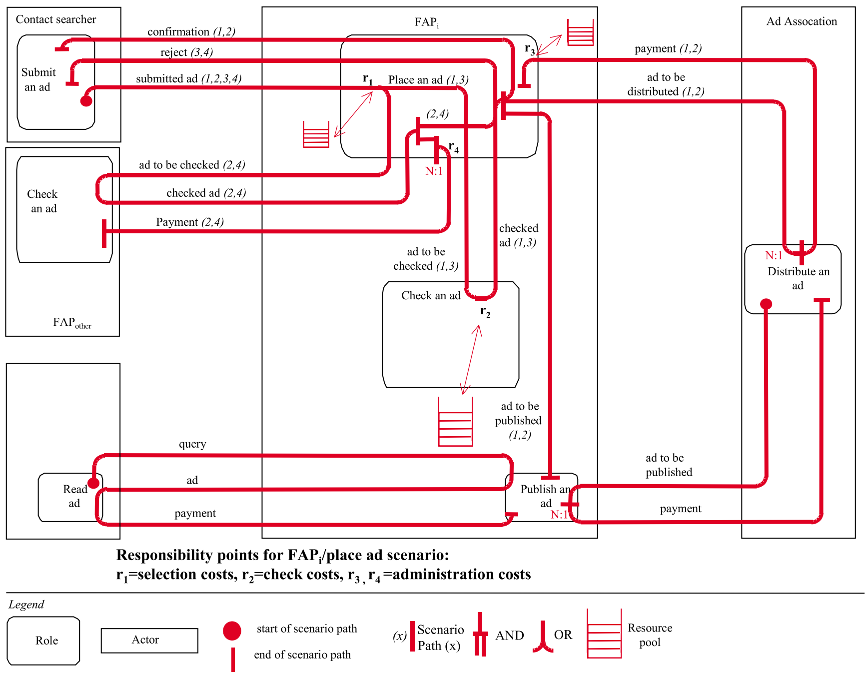


**Figure 4. Business process model for the Ad Association business idea.**

## 3.5 System Architecture Viewpoint

### 3.5.1 System Architecture

Two candidate system architectures will be presented for the Ad Association case. Both are based on a 3-tier architectural style in which a system is decomposed into three components: (1) the database, (2) the business logic, and (3) the user interface. This division reflects the principle of separation of concerns: a component should be responsible for one task only. Adhering to this principle minimizes the impact of change of one component on other ones. In addition, a 3-tier architectural style caters for distributedness and scalability. All of these are important quality attributes in e-business systems.

The following two architectural variations have been designed: (1) a decentralized database (Fig. 5) and (2) a centralized database (Fig. 6). In the first alternative, each FAP maintains its own database of ads that are offered to its readers, and sends its ads to the Ad Association for further distribution. In the second alternative, the Ad Association maintains the database of all ads for all readers centrally. A reader sends a request for an ad via the website of a local FAP, but this FAP will forward the request to the Ad Association.

### 3.5.2 System Architecture Scenarios

We will now evaluate the two achitectural solutions *with respect to the variation centralized versus decentralized database*. We neglect network costs, because the current tendency is that these are much cheaper then database or message server costs. The database server costs comprise all costs for having a local or central database server. For the decentralised scenario, we assume a message server (e.g., an SMTP server), which incurs costs. All these costs are accounted for on a per scenario basis. This means that no fixed costs exists, as these are allocated to each individual execution of a scenario, based on the expected number of executions per time-frame. Note that the database server(s) and the message server are not part of the business value and process model, so their impact on the costs cannot be assessed by evaluating a business value or process model in isolation.

Table 3 and Table 4 show costs for the scenario *submit ad* for all actors (except contact searchers). For the *submit ad* scenario, 4 paths can be identified (paths 1,2 for ads which are published and locally or remotely checked, paths 3,4 for ads which are rejected and locally or remotely checked).

## 3.6 Profitability and Feasibility

We have performed the first cycle of our proposed requirements engineering process for innovative e-business

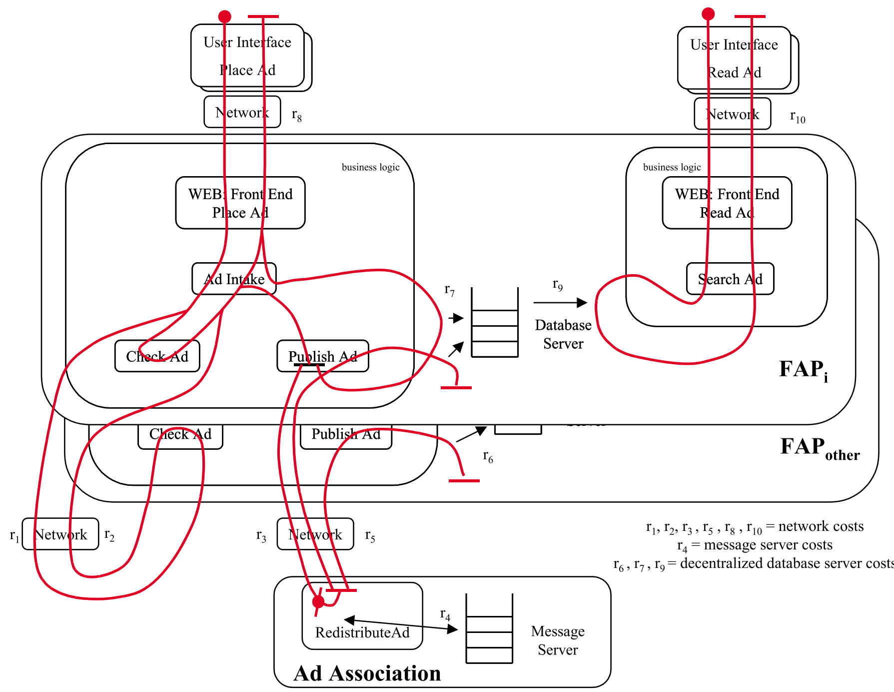


**Figure 5. A decentralized architecture.**

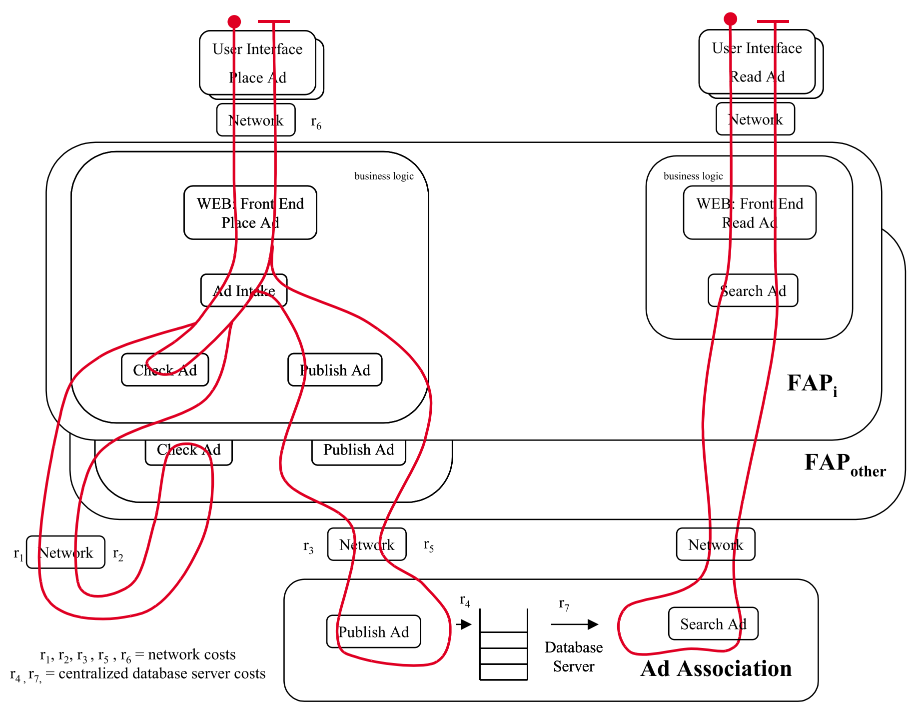


**Figure 6. A centralized architecture.**

**Table 3. Cost sheet for the decentralized variant (System Architecture viewpoint)**

| | | | |
|---|---|---|---|
| *Viewpoint* | System architecture (decentralized) | | |
| *Actor* | $FAP_i$ | $FAP_{other}$ | Ad Association |
| *Scenario* | Submit ad (Costs) | | |
| 1 (60%) | *decentr-dbase* | 0 | 0 |
| 2 (20%) | *decentr-dbase* | 0 | 0 |
| 3 (15%) | 0 | 0 | 0 |
| 4 (5%) | 0 | 0 | 0 |
| *Scenario* | Distribute ad (Costs) | | |
| 1 (100%) | 0 | *decentr-dbase* | *message server* |
| *Scenario* | Read ad (Costs) | | |
| 1 (100%) | 0 | *decentr-dbase* | 0 |

**Table 4. Cost sheet for the centralized variant (System Architecture viewpoint)**

| | | | |
|---|---|---|---|
| *Viewpoint* | System architecture (centralized) | | |
| *Actor* | $FAP_i$ | $FAP_{other}$ | Ad Association |
| *Scenario* | Submit ad (Costs) | | |
| 1 (60%) | 0 | 0 | *central dbase* |
| 2 (20%) | 0 | 0 | *central dbase* |
| 3 (15%) | 0 | 0 | 0 |
| 4 (5%) | 0 | 0 | 0 |
| *Scenario* | Distribute ad (Costs) | | |
| 1 (100%) | 0 | 0 | 0 |
| *Scenario* | Read ad (Costs) | | |
| 1 (100%) | 0 | 0 | *central dbase* |

information systems to be able to answer the questions whether the business idea at hand is profitable and feasible.

A first positive idea about the technical feasilibility has been given by outlining a business process model and two system architectures for the service-based scenario set, which support all the identified scenarios.

Confidence building in commercial viabilibility is much more complicated. This process is performed as follows.

1. **Estimate scenario frequency.** For each scenario, estimate the number of times a scenario occurs in a given period (say a month).

2. **Identify requirement sets.** Requirement sets are formed by all meaningful combinations of system architectures, which support the corresponding business processes and business value models. In this paper, we have two sets: (1) the set comprised by the business value model, the process model, and the centralized system architecture, and (2) the set consisting of the same business value and process models, and the decentralized system architecture.

3. **Estimate overall profit of a requirement set for each actor.**

   (a) *Estimate profit/costs per viewpoint per scenario occurrence.* Estimate, per scenario and per viewpoint , the profit/costs of an actor. This process is explained in Secs. 3.3.4, 3.4.2, and 3.5.2. The overall estimate should take in account the insight in the scenario's profit and cost factors, which is obtained from *all* three requirement viewpoints. For example, for the centralized variant, database costs for FAPs are likely to be lower than in the decentralized variant, but the redistribution fee may be higher due to a database, which has to be exploited by the Ad Association.

   (b) *Calculate expected profit/costs per scenario occurrence.* For each scenario, an actor is involved in, totalize the profits found for each viewpoint in the previous step.

   (c) *Calculate total expected profit for an actor.* Calculate the total profit for an actor, by, for each scenario, multiplying the number of times a scenario occurs with the profit of a scenario occurence, and by totalizing all these products for scenarios performed by the actor.

Note that we must aggregate profits/costs of each viewpoint on the *scenario* level and not on the *scenario path* level. Scenarios are conceptually the same for each viewpoint, but it is possible that viewpoints contain different numbers of scenario *paths* for the same conceptual scenario.

## 4 Conclusions and Lessons Learned

The key point of this paper is that in a first stage of requirements engineering for innovative e-business ideas, quick justification is needed concerning the commercial viability of the initial idea as well as its technical feasibility, before starting an in-depth requirement engineerings track.

To this end, we have proposed a scenario method that exploits separation of concerns to facilitate a clear discussion with a wide range of stakeholders. In our approach, scenarios play two roles: (1) estimation of profits and costs in each viewpoint to determine overall profitability, and (2) integration of separately developed viewpoints.

Our approach is unique in the sense that we deal in an *integrated* way with requirements related to business processes, system architecture, *and* business value networks of actors. The advantage of this is that innovative forms of doing e-business are designed in a way coherent with business process and information systems.

We have presented a first design cycle which aims at assessing the commercial and technical feasibility of the e-business idea. In this cycle, we develop first a requirements viewpoint which described CxO-level requirements and essentially shows *who* is offering *what* to *whom* and expects *what* in return. Second, we identify requirements regarding *how* a value model should be operationalized in terms of multi-actor business processes. Third, we develop (alternative) system architectures supporting the business value and process models.

For each requirement viewpoint, we identify the *same* set of scenarios, expressed by *different*, in structure as well as in quantity, scenario paths. Exploration of these paths lead to profit and costs sheets for scenarios, carried out by actors, for a specific viewpoint. If we estimate profit and cost factors in these scenario paths, and totalize them, we get a good impression of the profitability of the business idea for each actor.

Our experience in applying this method in a number of e-business projects, is that stakeholders are not seeking for an exact, precise quantification of profits and costs in a first stage of requirements gathering. However, they *are* interested in identifying dominant profit and cost drivers, and in reasoning about these drivers when investigating design alternatives, at an early stage.

**Acknowledgement.** This work has been partly sponsored by the Stichting voor Technische Wetenschappen (STW), project nr VWI.4949.

## References

[1] R.J.A. Buhr. Use case maps as architectural entities for complex systems. *IEEE Transactions on Software Engineering*, 24(12):1131–1155, 1998.

[2] Soon-Yong Choi, Dale O Stahl, and Andrew B. Whinston. *The economics of doing business in the electronic marketplace*. Macmillan Technical Publishing, Indianapolis, 1997.

[3] J. Gordijn, J.M. Akkermans, and J.C. van Vliet. Business modelling is not process modelling. In Stephen W. Liddle and Heinrich C. Mayr, editors, *Conceptual Modeling for E-Business and the Web, Salt Lake City, Utah, 9-12 October*, volume LNCS 1921, pages 40–51. Springer Verlag, 2000. Also available from http://www.cs.vu.nl/˜gordijn.

[4] J. Gordijn, J.M. Akkermans, and J.C. van Vliet. Selling bits: A matter of creating consumer value. In Kurt Bauknecht, Sanjay Kumar Madria, and Gunther Pernul, editors, *First International Conference on Electronic Commerce and Web Technologies EC-Web 2000, London, UK, 4-6 September*, volume LNCS 1875, pages 48–62, Berlin, 2000. Springer Verlag. Also available from http://www.cs.vu.nl/˜gordijn.

[5] J. Gordijn, J.M. Akkermans, and J.C. van Vliet. Value based requirements creation for electronic commerce applications. In *Proceedings of the 33rd Hawaii International Conference On System Sciences (HICSS-33)*, pages CD–ROM. IEEE, January 4-7 2000. Also available from http://www.cs.vu.nl/˜gordijn.

[6] J. Gordijn, J.M. Akkermans, and J.C. van Vliet. What's in an electronic business model. In R. Dieng and O. Corb, editors, *Knowledge Engineering and Knowledge Management - Methods, Models, and Tools, 12th International Conference, EKAW 2000, Juan-les-Prins, France, October 2000*, volume LNAI 1937, pages 257–273. Springer Verlag, 2000. Also available from http://www.cs.vu.nl/˜gordijn.

[7] Morris B. Holbrook. *Consumer value: a framework for analysis and research*. Routledge, New York, 1999.

[8] Gerald Kotonya and Ian Sommerville. *Requirements Engineering - Processes and Techniques*. John Wiley & Sons Inc., Chichester, England, 1998.

[9] R. Motschnig-Pitrig, H.W. Nissen, and M. Jarke. View-directed requirements engineering: A framework and metamodel. In *Proceedings of the 9th International Conference On Software Engineering and Knowledge Engineering, SEKE'97*, June 1997.

[10] Martyn A. Ould. *Business Processes - Modelling and Analysis for Re-engineering and Improvement*. John Wiley & Sons, Chichester, England, 1995.

[11] James Rumbaugh, Ivar Jacobson, and Grady Booch. *The Unified Modelling Language Reference Manual*. Addison Wesley Longmann, Inc., Reading, Massachusetts, 1999.

[12] Ian Sommerville and Pete Sawyer. Viewpoints: principles, problems and a practical approach to requirements engineering. *Annals of Software Engineering*, 3:101–130, 1997.