On Industrial Use of Requirements Engineering Techniques

Lars Bækgaard  
Department of Business Studies  
Aarhus School of Business, University of Aarhus  
Fuglesangs Alle 4, DK-8210 Aarhus V, Denmark  
lab@asb.dk

Jens Bæk Jørgensen and Kristian Bisgaard Lassen  
Department of Computer Science, University of Aarhus  
IT-parken, Aabogade 34, DK-8200 Aarhus N, Denmark  
jbj@daimi.au.dk  
k.b.lassen@daimi.au.dk

Abstract. We discuss two experiments in which requirements engineering techniques has been used and evaluated. In the first experiment a technique called Executable Use Cases is applied in the development of an IT system for public utility services. In the second experiment a technique called Activity Cases is applied in the development of an IT system for a public library. For each experiment we discuss the lessons that we have learned. We use the lessons to sketch a plan for future research in terms of a set of scenarios for combined use of Executable Use Cases and Activity Cases.

1 Introduction

We present, discuss, and learn from two experiments in which two requirements engineering techniques called Executable Use Cases and Activity Cases has been applied and evaluated. Executable Use Cases support animation of process models and Activity Cases support flexible activity modeling. Based on the experiments we present a set of scenarios that combine the two techniques.

Development methods like Structured Analysis [De78], Multiview [AW90] and Contextual Design [BH98] are based on the assumption that information systems development should be based on a thorough understanding of the work contexts of the systems. Our work is based on the same assumption and our aim is to provide create techniques that provide such understanding and facilitate creative requirements processes that are based on well-founded understandings of work contexts.

Our research method can be characterized as design science [MS95], [HM04]. We have created the techniques and experimented with their use in order to evaluate their relevance as requirements engineering techniques.
We conducted two experiments based on result from a requirements specification workshop in which the techniques Executable Use Cases and Activity Cases were presented and discussed. The experiments were conducted in real-world requirements engineering situations. The experiments represents the first of a series of evaluation activities in our design science project. In future activities in the project we will experiment with combinations of the two techniques and we will decide if it is desirable to modify the techniques. This implies that the presented results are preliminary results from a larger ongoing project.

The paper is structured as follows. Section 2 we introduce the notion of executable use cases and discuss an experiment where a set of business processes are modeled by means of executable use cases. In Section 3 we introduce the notion of activity cases and we discuss an experiment where a set of business processes are modeled by means of activity cases. In Section 4 we discuss some of the implications of the experiments and we suggest directions for future research.

2 Experiment 1 - Executable Use Cases

The Public Utilities Aalborg (PUA) are responsible for providing gas, electricity, heating, water, sewer services, and garbage collection to the Aalborg region, which has approximately 160,000 inhabitants. PUA is the shared administration for six companies, one for each type of service. In total, PUA has around 450 employees. The employees use a number of IT systems in their daily work. Examples are accounting systems and geographical information systems (GIS).

PUA has hired the software company WM-data to deliver a new IT system, which we will call the Authorization System (AS). AS will be used by PUA to administer the access rights for different users of different IT systems. AS should support creation of new user accounts, update of the rights of existing users, and closing of user accounts.

WM-data is a Scandinavian company, whose branch in Aalborg will develop AS. It is beforehand decided that AS should be based on OIM—a standard off-the-shelf system for management of user rights and privileges across enterprises. This means that in the first place, WM-data must carry out analysis work aimed at aligning the real needs of PUA with what is actually possible with the given technology, i.e., OIM.

As part of this analysis work, WM-data has been interested in trying out the requirements engineering approach that we call Executable Use Cases (EUCs) [JB04b], and we have used the AS project as an experiment for our research on EUCs.

---

1 Experiment 1 was carried out by the second and the third authors.
2 Oracle Identity Manager, formerly known as Oracle Xellerate Identity Provisioning.
We will describe use of EUCs in the AS analysis project in this section and we will report on some preliminary findings, based on work carried out in the period April-June 2007, where a series of analysis workshops were held; the participants in the workshops were personnel from PUA, analysts and developers from WM-data, and the second and the third authors of this paper.

**Executable Use Cases (EUCs)**

An *Executable Use Case (EUC)* [JB04b] supports specification, validation, and elicitation of requirements. EUCs spur communication between stakeholders and can be used to narrow the gap between informal ideas about requirements and the formalisation that eventually emerges when a system is implemented. An EUC consists of three tiers. Each tier represents the considered work processes that must be supported by a new system. The tiers use different representations: Tier 1 (the *informal tier*) is an informal description; tier 2 (the *formal tier*) is a formal, executable model; tier 3 (the *animation tier*) is a graphical animation of tier 2, which uses only concepts and terminology that are familiar to and understandable for the future users of the new system. Tier 3 has the potential to offer significant advantages as a means of communication.

The three tiers of an EUC should be created and executed in an iterative fashion. The first version of tier 1 is based on domain analysis, and the first version of tiers 2 and 3, respectively, is based on the tier immediately below. Figure 1 illustrates the approach.

![Executable Use Cases](image)

**Figure 1** Executable Use Cases

EUCs have notable similarities with traditional high-fidelity prototypes of IT systems; this comparison is made in more detail in [BJ04]. In [JB04a], it is described how an EUC can be used to link and ensure consistency between, in the sense of Jackson [Ja04], user-level requirements and technical software specifications.
An EUC can have a broader scope than a traditional use case [Co01]. The latter is a description of a sequence of interactions between external actors and a system that happens at the interface of the system. An EUC can go further into the environment of the system and also describe potentially relevant behaviour in the environment that does not happen at the interface. Moreover, an EUC does not necessarily fully specify which parts of the considered work processes will remain manual, which will be supported by the new system, and which will be entirely automated by the new system. An EUC can be similar to, in the sense of Lauesen [La03], a task description.

AS EUC Tier 1: Analysis Documentation Produced by WM-data

At the workshops that we report on here, analysts from WM-data documented work processes (current and future), primarily by drawing swim lane diagrams as the one shown in Figure 2 which depicts the workflow that must be carried out, when a new user is created, who should be allowed to access the PUA network. The input necessary to make the descriptions came from relevant personnel from PUA, who participated in the workshops.

Figure 2 Partiel swim lane diagram showing creation of a new user of the PUA network

3 The partial swimlane diagram is in danish and included to illustrate the diagrams that were used to create EUCs.
A number of swim lane diagrams were created. In total, $3 \times 6 = 18$ different work processes are considered. The number 3 comes from the fact that we consider three different action types, namely (1) creation, (2) update, and (3) closing of user accounts; the number 6 comes from the fact that we consider 6 different IT systems (or more generally, “resources”), namely the systems named Network, Navision, GIS, Xellent, EDoc, Geo Environ.

**AS EUC Tiers 2 and 3**

Tier 2 is (1) a formalized version of tier 1, and (2) an execution engine that drives the graphical animation of tier 3. The animation tier, tier 3, is created with the help of Magee et al’s SceneBeans animation framework [MP00]; Figure 3 shows a snapshot.

The animation tier is consistent with the CPN model of the formal tier. At any time, the graphical animation represents the current state of the CPN model and mimics its execution. Technically, the link between the CPN model and the animation tier is that the CPN model calls drawing functions when it executes. The CPN model thus causes graphical objects like organisational unit icons and document icons to be created, moved, deleted, etc. in the animation.

![Figure 3](image.png)

**Figure 3 Snapshot of the animation tier of the EUC**

Figure 3 mimics a situation, where five stakeholders are cooperating in the handling of a request to open a new user account. The stakeholders are the department manager, a secretary, the chief executive, the IT department of PUA, and an external IT centre. The document icon represents the request. In the current situation, the request is on its way from the secretary to the chief executive; the animation user sees the document icon moving. The piece of information attached to the icon counts how many times the modelled document has been handed over.
When the animation starts, it presents a dialog box that prompts the animation user to specify what he wants to mimic. Which system is in concern? Do we want to consider creation, update, or closing of a user account? It can also be specified whether that handling mode should be normal or urgent; in the latter case, some of the stakeholders which are involved in the normal handling are bypassed.

The animation layer is in general generated manually, as was the case of our EUC as well. This meant that we had to map each state change/transition in the model to a concept in the animation, so e.g. when the transition that moves a letter from one person to another occur in the model, we made the necessary SceneBeans command to reflect this in the animation.

Lessons learned

Through meetings with PUA we learned that EUCs are valid in many phases in the lifecycle of the system that we are working towards. The lessons that we learned can be divided into four parts that we explain in the remainder of this section.

_EUCs offer better understanding of requirements_. Text documents as well as swim lane diagrams (Figure 2) were used to structure the requirements found in workshops by users and experts that knew of existing workflows. The attendees at the workshops found it is hard to get a full overview of the whole process just using text documents and swim lane diagrams, since these descriptions describe scenarios, rather than representing a complete model. Our EUC had all the requirements encoded as rules and when it is executed, it shows how all the requirements play together. A further improvement is that with the EUC, it is easy to get users to talk about requirements; since they did not have to talk of the requirements from e.g. the swim lane diagrams’ box and arrows, instead they used terminology that the EUC’s graphical presentation offer.

_EUCs are useful when selling a new system_. When presenting users for how a new system will work, we learned that text documents and swim lane diagrams, often seem inappropriate to convey the ideas behind the new design. The analysts at PUA thought that the EUC was a much better means of communication when talking to non-technical users or management that do not have time to fully understand the traditional requirement artefacts. Furthermore, PUA would like to present users of the system, how the current workflows are and how the future workflows will be, using the same graphical layer.

_EUCs give momentum in the Software Development Phases_. A software developer attended the meeting with PUA and software analysts when we presented the EUC. He claimed that this EUC gave him more contexts of the workflows that occur at PUA, with regards to who does what and when, and also that he thought that it was easier to remember them, than using e.g. swim lane diagrams.
EUCs keep focus on what is important. It was noted when we presented the EUC that in contrast to traditional prototyping, people actually discussed what was important at this stage of the analysis: The workflows. One person noted that over the years he has observed that if you show a person a screenshot of a user-interface and want him to discuss the workflow in the system, the person may begin to focus on the user-interface itself.

This could be because he does not have anything in particular to say about the workflow, but feels obligated to comment on something. In this way a lot of unnecessary and untimely discussions were generated. He felt that EUCs, because of their simple and non-user-interface appearance make people consider how the systems workflow are and should be, rather on e.g. prematurely designing the user-interface.

EUCs are useful after deployment. Artefacts generated in the analysis phase of a software project, are seldomly used for anything else but input to the design phase. One of the people at PUA thought that EUCs would be very useful as a supplement to manuals and other kinds of documentation of the system. For non-technical users these traditional ways of documenting the functionality of the system are hard to read and understand, so instead of using the documentation they will often resort to simply asking other people instead. The PUA person continued, and said that the EUC would help the individual to understand his or her role in the organization.

3 Experiment 2 - Activity Cases

The experiment

The experiment was carried out as a part of a pre-analysis project at a public Danish library, Vejle Library. The purpose of the pre-analysis project was to identify possible improvements of the library’s mediation of library user’s search for relevant information. A library user that requests search mediation will be serviced by a librarian. During a search session a librarian will use the library user’s expressed information needs to search for relevant information via search engines and databases.

During the pre-analysis project a number of analysis and design activities were carried out. User simulations were used to establish understandings of the current activities. Modeling of current activities were used to capture aspects of these understandings. Formulation of future stories and brain storms were used to create visions about changed activities and new ways of using IT systems. Modeling of Future situations were used to capture aspects of the visions.

Experiment 2 was carried out by the first author.
The purpose of the research project was to experiment with activity modeling by means of activity cases and interaction patterns. The research was carried out by a librarian and a researcher. The librarian was the primary actor in the pre-analysis project. The researcher served as a consultant in the pre-analysis project.

**Modeling techniques**

We used activity cases, informal process models, and interaction patterns to model existing and envisioned information search activities.

An activity case defines selected characteristics of an existing activity or a vision about a new activity [Bæø5]. We use the following four aspects (NICE) to characterize activity cases. The *name* expresses the essence of the corresponding activity. The name is important because we use to refer to an activity and indicate what it is about. The *intention* expresses the purpose of an activity. The *content* of an activity can be described in terms of actions, events, actors, resources etc. The content can be described by means of, say, activity diagrams [RJ99], BPMN diagrams [Wh04], and EPC diagrams [De02], [LL06], or they can be described in terms of text. The *environment* of an activity can be described in terms of actors that interact or statically related with the activity.

We have used interaction patterns to define the structural aspects of the information search activities in the library. An interaction pattern defines a dynamic relation between two participants. At least one of the participants must be an actor [Bæø6].

![Interaction patterns](image)

Figure 4 Interaction patterns

Figure 4 shows the notation we use to model interaction patterns. Circles represent actors. Rectangles represent objects (things or information). Dotted rectangles represent locations. MOVE is a pattern where an actor moves an object to a destination. GIVE is a pattern where an actor gives an object to another actor. SENSE is a pattern where an actor senses aspects of an object (the dotted line indicates that no visible, physical action is taken place). MODIFY is a pattern where an actor modifies an object. CONTROL is a pattern where one actor controls the activities of another actor.

**The situation**

The experiment focused on an activity where a library user has a set of information. A librarian uses his understanding of these needs to search for information resources. The user and the library engage in a dialogue about the user’s information needs, potential search terms, and the relevance of search results.
The information search activity is characterized by communicative interaction between user and librarian and it is characterized by a shared interaction with IT-systems like the Internet and library databases. Also, there is an important element of cognitive activities where the user and the librarian tries to understand the problem at hand and where they consider possibilities and think about formulations and search results.

Activity case 1 – current situation

This activity case represents selected aspects of the current mediation of user’s information search. It is based on user simulations during which two librarian’s simulated the information search activity playing the roles of user and librarian.

Name. Information search.

Intention. To mediate a library user’s search for information resources.

Content. The leftmost diagram in Figure 5 represents the current search activity in terms of a set of activities that the librarian and the user carry out. The notation is informal. Rectangles represent objects and round-corner rectangles represent activities. Formulate query is an activity in which the user expresses information needs and the librarian asks questions and suggests interpretations. Based on this the librarian formulates a set of search terms that is used as input to a search activity in which the librarian uses the search terms to ask a query to a search system. The user and the librarian analyse and evaluate the answer—a set of objects that is returned from the search system (database, Internet search engine, etc.). Select is an activity in which the librarian uses “cut & paste” to copy relevant resources from an answer to the resource collection—a text document in which the selected resources from search answers are stored. Finish is an activity in which the resource collection is formatted and enhanced with clarifying comments.

![Figure 5 Current search activities](image)
The rightmost diagram in Figure 5 represents the current search activity in terms of a set of interactions. Two persons, a librarian and a user, participates in the activities. The user expresses information needs. The librarian asks questions and suggests interpretations. The librarian adjusts search parameters and poses query with search terms to the search system. The search system creates an answer to a query. The librarian updates the resource collection that contains selected resources from query answers. The user can see the answers and the resource collection.

**Environment.** Other activities in which librarian and user engage.

**Activity case 2 – future situation**

This activity case represents selected aspects of the current mediation of user’s information search. It is based on brain storms and analysis of Activity Case 1b.

**Name.** Information search.

**Intention.** To mediate a library user’s search for information resources.

**Content.** The leftmost diagram in Figure 6 represents the envisioned future search activity in terms of a set of activities that the librarian and the user carry out. The notation is informal. Rectangles represent objects and round-corner rectangles represent activities.

![Figure 6 Future search activities](image)

The major change when compared to the model in Figure 5 is that the model in Figure 6 presumes that an actor (IT system) called an Resource Manager can be used to manage the resources that are evaluated as relevant parts of the query answers. Rather than manually updating an unstructured resource collection then librarian uses the Resource Manager to select and modify selected resources.
The rightmost diagram in Figure 6 represents the envisioned future in terms of a set of interactions. Two persons, a librarian and a user, participates in the activities. The resource manager updates the resource collection. The user controls the resource manager. The user expresses information needs. The librarian asks questions and suggests interpretations. The user adjusts search system parameters. The user poses query with search terms to the search system. The search system creates an answer to a query. The librarian adjusts search system parameters. The librarian poses query with search terms to the search system. The librarian controls the resource manager. The resource collection is “read” by other systems.

**Environment.** Other activities in which librarian and user engage. Environment of other systems.

**Lessons learned**

Activity case 1 were created in order to understand the existing situation and to support discussions about potential improvements. Activity case 2 represents the results of these discussions. The following lessons are based on the observed experiences from the use of activity cases and from informal conversations with library staff.

*Activity cases support modeling flexibility*. Activity cases support a balanced focus on the elements of a modeled activity that are considered relevant, i.e., actors, activities, things, and information etc. No specific modeling languages are presumed. Any combination of formal and informal modeling languages can be used. This turned out to be very relevant in the library case because the informal activity flow modeling notation turned out to give an insufficient view of the activities. The library case is characterized by many iterative interactions that are not easily captured in terms of activity flow. The interaction patterns turned out to supplement the activity flows.

*Activity cases support innovation*. Interaction patterns facilitate creative discussions about the roles played by IT-systems in business activities. The use of interaction patterns in activity cases turned out to be useful in creative discussions about potential uses of IT-systems to improve the library search activities. The reason is that all interactions can be mediated by actors that are added between the interacting elements. For example, the Resource Manager can be viewed as a mediator that mediates the librarian’s interaction with the selected resources.

*Activity cases go beyond workflow modeling*. An activity case focuses on a selected activity system in terms of its name, intention, content, and environment. This implies that the modeler is encouraged to go beyond mere workflow modeling. In then library case we modeled the selected activity system (information search) in terms of both workflow and interaction patterns because the workflow perspective turned out to be an inadequate representation of the the roles of mediating tools in the search process.
4 Discussion

The following scenarios describe potential combinations of EUCs and activity cases. They are based on the lessons learned from the two experiments. All scenarios are based on the assumption that the requirements engineering process is structured around three EUC-tiers.

**Scenario 1 – Activity cases may be used as organizing principle.** When the three EUC-tiers are created, a significant amount of models may be created. These models can be organized as activity cases to ensure that all models are named in a coherent manner and contextualized in terms of their intentions and environments. The NICE template offers a framework for model organization. Different models of a specific activity can be traced if all models (Tier 1, Tier 2, Tier 3) are contextualized by means of the NICE template and if a coherent naming scheme is used.

**Scenario 2 – Activity cases may be used to support innovation.** Activity cases can be used to support innovation and creativity when Tier 1 is created. Activity Cases are useful for gaining the first ground in understanding a problem, and they do this by structuring the domain in actors, resources, actions, and events. EUCs on the other hand are useful for making a full behavioral model, of the discrete descriptions captured by each activity case, and to establish in a greater degree the resource perspective and control-flow perspective of the overall system. EUCs are used to support deep understanding of the dynamics of complex processes.

**Scenario 3 – Activity cases may be used to support flexible clarification.** Activity cases are used to support fast experimentation with processes that have turned out to be problematic in EUCs. When a process model is animated one or more deficiencies may be detected. In such situations the flexibility of activity cases may be used to experiment with ways to overcome each identified deficiency without having to formally model the process.

The next series of evaluation experiments in our design science project will be based on scenarios like these. The experiments will be based on industrial requirements engineering activities in which a set of promising scenarios will be evaluated. The scenarios implicitly characterize a requirements engineering process that combines flexibility and agility with formality and animated processes. It is very likely that such a process could have been used to improve our own experiments. The public services case could have used the NICE template to organize the various models and it could have benefited from the flexibility offered by Activity Cases. The library search case could have benefited from the animations offered by EUC. Animations of the search process could be distributed to relevant librarians and selected users as a basis of a more thorough validation of the proposed new search activity.
5 Conclusion

We have presented two different requirements engineering techniques called EUCs and Activity Cases. An EUC consists of three different models of the work processes that must be supported by a new system. Tier 1 is an informal model. Tier 2 is a formal, executable model that is based on Tier 1. Tier 3 is a graphical animation of tier 2. Tier 1 is created in an informal modeling activity. Tier 2 is created in a formal modeling activity. Tier 3 is an animation activity in which the formal model from Tier 2 is animated. An activity case is model of a work activity. It is based on a template with four headlines: Name, Intention, Content, and Environment. Activity cases can be used to experiment with models of existing and envisioned business processes. For each technique we have described and discussed an experiment in which the technique has been applied and evaluated. And we have sketched three scenarios that represent possible combinations of EUCs and Activity Cases. Future research includes industrial experiments that are based on these scenarios. The purpose is to identify characteristics of a requirements engineering process that combines flexible modeling, formal modeling, and process animation.

References


