Process Model Driven Requirements Engineering

Dag Rojahn Karlsen[1], Helle Frisak Sem[1], Steinar Carlsen[1]

[1] Computas AS, Lysaker Torg 45, N-1327 Lysaker, Norway
{dag.rojahn.karlsen, helle.frisak.sem, steinar.carlsen}@computas.com

Abstract. This paper demonstrates an approach to requirements engineering with process models as the hub for a family of consistent sub-models. The process model taken as a starting point gives focus, and the different requirement perspectives are served through the enrichment of the process model in several dimensions. Different views into the model family give the well-known specification models; information models, exchange models, service models, etc. The family of related sub models are refined iteratively to describe the need for information (NFI), the need for services (NFS), a shared common information model (CIM) and refined domain information models (DIM). We call this family of interrelated sub models of information systems *knowledge models*, since they capture both the structure and the behaviour of the subject matter domain. Working in an integrated way with several models serves to enhance consistency and to test the quality of each particular model.

Keywords: Process models, knowledge models, requirements engineering, enterprise models, information models

1 Background

This paper presents an enterprise modelling approach to requirements engineering discovered in our work in a project for a large Norwegian public sector organisation - LNOP. LNOP manages a substantial part of the national budget in Norway and administers a diverse set of schemes under precise regulations.

LNOP was established as a merger between two existing organisations, each with a long history. The local authorities and central government cooperate through 456 LNOP offices in municipalities and city boroughs. LNOP employs around 19,000 people. Of these around 14,000 are employed by the central government, and around 5,000 are employed by the local authorities. In addition to the local LNOP offices there are more than one hundred special units that perform centralised duties that would not be appropriate for front line local LNOP offices to perform.

LNOP has a large portfolio of legacy systems, some of them originating as far back as 1976. In order to move to an efficient IT situation characterised by automation and self-service, LNOP launched a large, high-profile IT modernisation programme in 2012, with projects involving several contractors, employing up to 300 internal and external staff members.

2 Overall Process

The project utilizes TOGAF [1] as its enterprise architecture framework. To produce the different architectural artefacts in the specification process, the programme has chosen the modelling languages BPMN 2.0 [10] for process modelling and UML 2.0 [11] for information and other modelling. This paper describes how we have combined process modelling and information modelling in a way that has been fruitful in the specification process.

We know that there are enterprise architecture frameworks in the market, such as ArchiMate [13], with the aim to describe the different layers of architecture in an integrated and holistic way. We cannot see, however, that these cover the integration between process and information models with a purpose as described below. We are aware, however, that this may be achieved using other languages and styles, but the aim of this paper is to describe what we did in an actual project.

To maintain consistency and control in a project with several contractors developing a number of applications, and a large portfolio of legacy systems, the project utilizes a service-oriented [2] [3] and model driven approach, specifying systems and their responsibilities and interrelations within a variety of architectural styles and models. For an example of service-oriented modelling methodology, see [4].

An ambitious goal of the project is to develop a Common Information Model, which is a common model for the exchange of information between applications, old and new, creating an information exchange language for LNOP. The motivation behind this is to minimize the dependencies between integrated applications, following an accommodated version of the pattern called Canonical Data Model [5].

2.1 The Overall Process for the Development Project

The goal of the project is to develop a number of new applications based on a common methodology. The agile method scrum [6] [7] is used as the development method. The business needs are specified on two levels, both administered in Atlassian JIRA [8] and documented in Atlassian Confluence [9]. The top level is called epic. Each epic is further specified by a set of user stories. Prioritising and planning is performed by means of the epics and the user stories.

Suitable sets of user stories are identified to belong to the same application, and the specification and modelling necessary for development starts, resulting in a Solution Architecture under the responsibility of the LNOP, and a Solution Specification under the responsibility of the contractor responsible for application development. Contracts for the development of a small set of user stories are then entered into and developed through three-week development sprints.

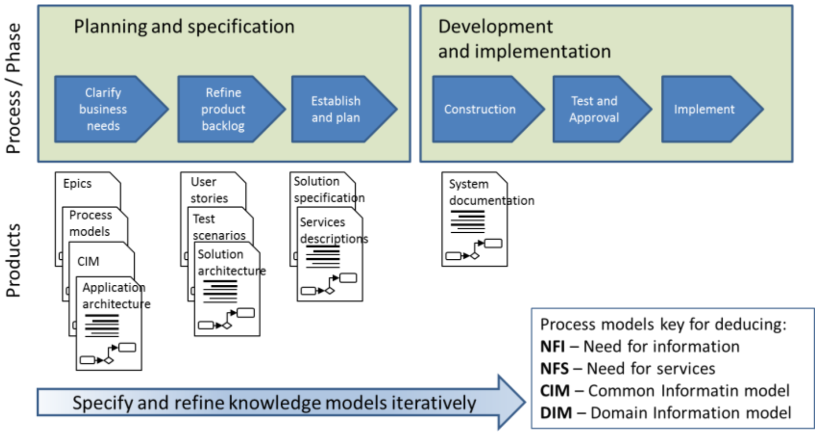

Fig. 1. The overall development process with phases and documentation products.

The paper focuses mainly on the first three steps, the planning and specification phase and how to work with some of the models involved.

2.2 Definitions

NFI – Need for Information
The information needed by business processes at a conceptual level as identified by process modellers.

NFS – Need for Services
The services needed by a process/activity. The service is required to provide information as identified by one or more needs (NFIs).

CIM – Common Information Model
The common information model specifies the common language used in the data exchange between all applications. Service information models are based on the CIM.

DIM – Domain Information Model
The information model is designed to support a given application domain. Parts of the model will cover information to communicate with other application domains – accordingly DIM and CIM are related.

2.3 Planning and Specification – The Initial Way

The project is organized as business requirement teams, process modelling, information modelling and solution specification teams. The requirement teams are responsible for identifying business needs, specified by epics and user stories.

The process modellers identify and model the business process models, and based on business needs they identify corresponding information needs.

The information modelling teams are of two kinds: One to develop the common information model and corresponding service information models, and another to develop the domain information models.

The solution specification teams are also of two kinds: One to develop the solution architectures and another to develop the solution specifications.

LNOP as client is responsible for all teams except for the domain information modelling teams and the solution specifications teams that are under the responsibility of the contractors.

The common information model team is continuously developing the common information model, and provides the solution specification team with a dedicated representative. Architects on the client side and on the contractor side are both engaged in the specification phase, presenting their work to the client's business experts for the field in question for adjustment and approval.

Initially, each model and architecture product was made separately, often with different people and not enough communication between the teams working on different models. Much time has been spent in order to align the different models, not to speak of the time wasted in the establishment and planning step, or even the construction step, because the models were not consistent with each other. Working in an integrated way with more than one model serves both to enhance consistency and to test the quality of each model.

3 The new approach

Getting an overview of all the dependent activities and their relations has been one of the biggest challenges in the project.

The approach so far has been to look at different dimensions of the project more or less independently, resulting in a fragmented world of documents and models. In addition, the size of the project has made it extremely difficult to get the necessary overview of how all designed artefacts are linked and related.

To meet these challenges, we have developed a holistic approach that integrates models from different domains into what we like to call *knowledge models*.

A *knowledge model* in our terminology must include the process dimension – it must describe what is going on (processes and tasks). In addition the processes must link to other dimensions, such as information, organisation and roles, competence and skills, products and services, etc. depending on the purpose of the model.

The main thought is that process models are key – they represent the glue that directly or indirectly ties the different dimensions together, enabling us to build holistic knowledge models.

The approach has been developed in an information systems development project, a project within the larger modernisation programme, in a team covering roles from LNOP and contractor, including technical and business expert members, as well as process and information modellers. The task of the team has been to give necessary specifications for the development of a particular application, already identified in earlier architectural work, provide the necessary models, in particular information models, and identify system context and dependencies to other applications.

The outcome has been a holistic model that includes:

- **Conceptual views** of the processes (business level) identifying the need for information (NFI) with links to a catalogue of concepts and terms
- **Logical views** of the processes identifying the need for services (NFS), and for data exchange between processes/applications as a first step towards aommon Information Model (CIM), and for information structures to support the domain applications
- **Logical models** that combine process and information constructs in the same diagrams to detail the data exchange models and application domain models

3.1 The Conceptual Models

We have developed a way of working to combine business process models (using BPMN) with Need For Information (NFI) and links to the concept catalogue. The technique keeps the process diagram simple by linking to separate diagrams that model the information needs and relate them to the terms. This allows the same information need to be referred several places in the process diagram.

These process models are built by business analysts who know the business processes and have the knowledge to identify the information needed in the processes.

Another important part of the project is to have a concept catalogue where the different terms and concepts are defined, enabling a common language between people from different disciplines.

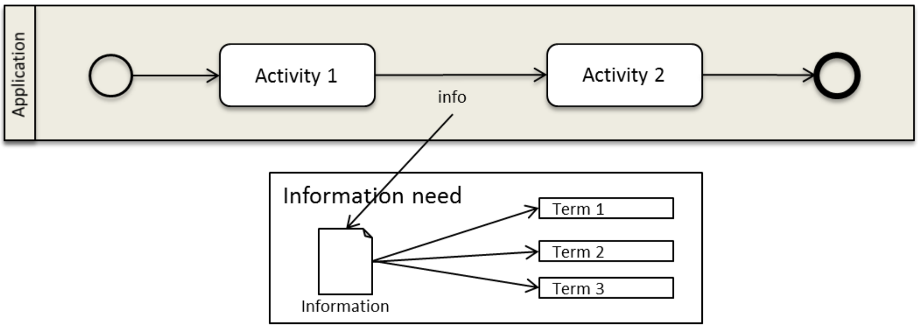

Fig. 2. Conceptual process model identifying Need for Information (NFI) with links to concepts / terms.

3.2 The Logical Models

Below we summarise properties regarding the various logical models involved in the approach.

3.2.1 Need for Services

The project is service-oriented and needs-driven. Needs for Services (NFS) must be identified early in the specification process. This is achieved by building process models that identify the communication between applications and add logical entities that represent the NFS'es to the process models.

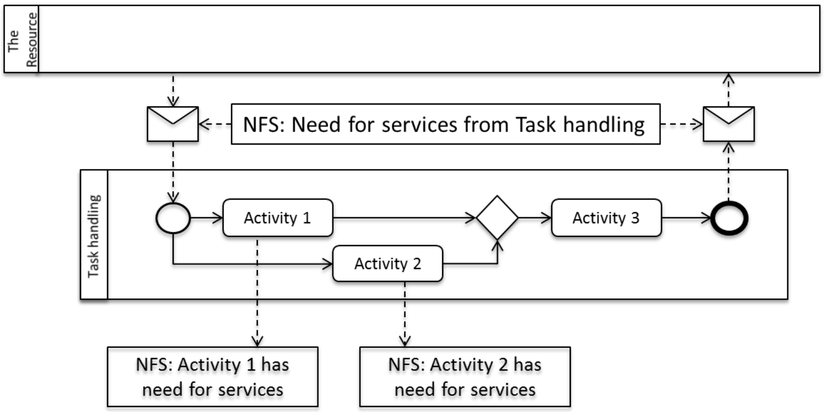
The Resource
NFS: Need for services from Task handling
Task handling
Activity 1
Activity 2
Activity 3
NFS: Activity 1 has need for services
NFS: Activity 2 has need for services

Fig. 3. Need for services illustrated.

The system in focus is the application being specified. The figure illustrates that other LNOP systems may require services from the application being specified, and also that the application itself has a need for services from other service providers at LNOP.

By doing this for all processes in the application being specified, and for other LNOP applications to be developed, the central integration team of the project gets an overview of all the need for services both to be provided by these applications, as well as those needed by the applications themselves.

In this process the central integration team will be able to identify if the same or a similar service may be needed by several of the LNOP applications and then they can harmonize the needs and specify one service realization that supports all similar needs.

3.2.2 Identifying CIM as part of Data Exchange Process Modelling

One of the main targets in the project is to establish a Common Information Model (CIM), requiring a special focus on data exchange, as the modelling language used in the communication between applications is required to be CIM.

The Data exchanged is represented as UML classes, as a first step towards a physical representation. Detailing the attributes is not required at this stage.

In Figure 4 we specify what data are exchanged between the two applications and how they are represented as UML classes. The classes define the protocol used in the communication between the two. As the customer requires that all communication between applications are according to a common language model, the UML classes is supposed to be a part of CIM. Accordingly the definition of the UML classes has to take into account the already existing parts of CIM, if there are any (sub-) models related to the task at hand.

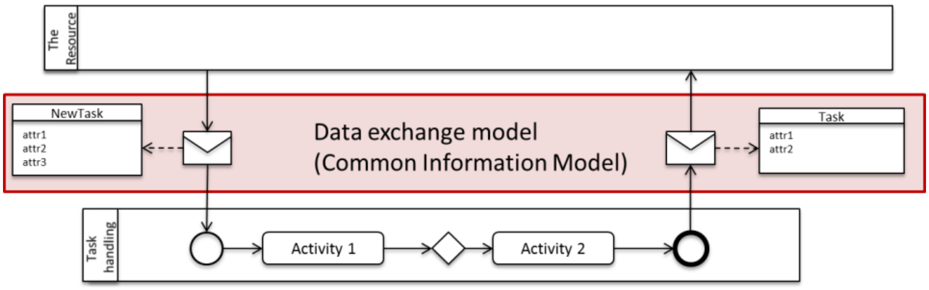

Fig. 4. Data exchange model.

The simple model shown in Figure 4 also illustrates that our approach combines BPMN and UML artefacts in the same model in the same diagram. By doing so, we take the step from process models and information models that live their own lives, to a combined representation of the two that focuses on the role information plays in the process – we take the step from information to knowledge.

3.2.3 Identifying DIM as part of Application Processes Modelling

The Domain Information Model is identified as the need for data objects to support a given application. Our approach is to model the processes that the application is performing and identify their need for information in the same model.

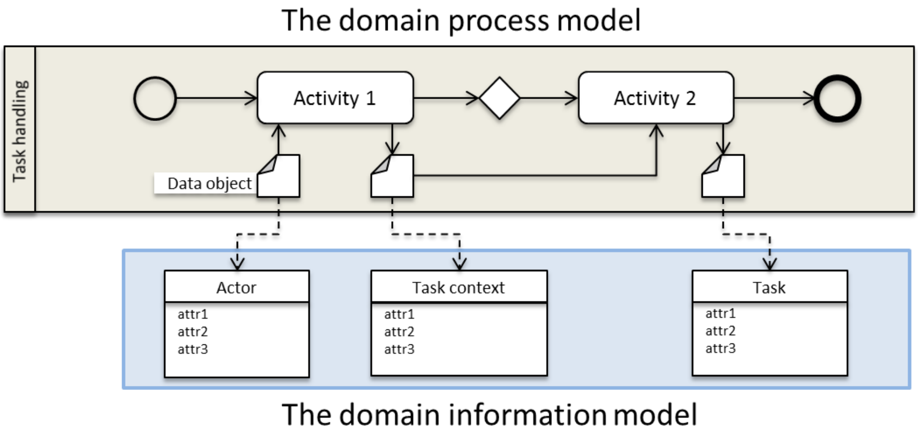

Fig. 5. Domain information model.

This is illustrated in Figure 5, which shows a standard BPMN model that includes data flow, combined with UML classes that detail the data objects in the process model. By combining the two, the information modeller understands the role the information will play in the application and has a much easier task of providing a good result. Again we take the step from information to knowledge.

3.2.4 Knowledge Models and Information Models

The detailed knowledge models are a result of detailing the UML classes with the actual attributes both for the data exchange (CIM) models and the domain specific (DIM) models.

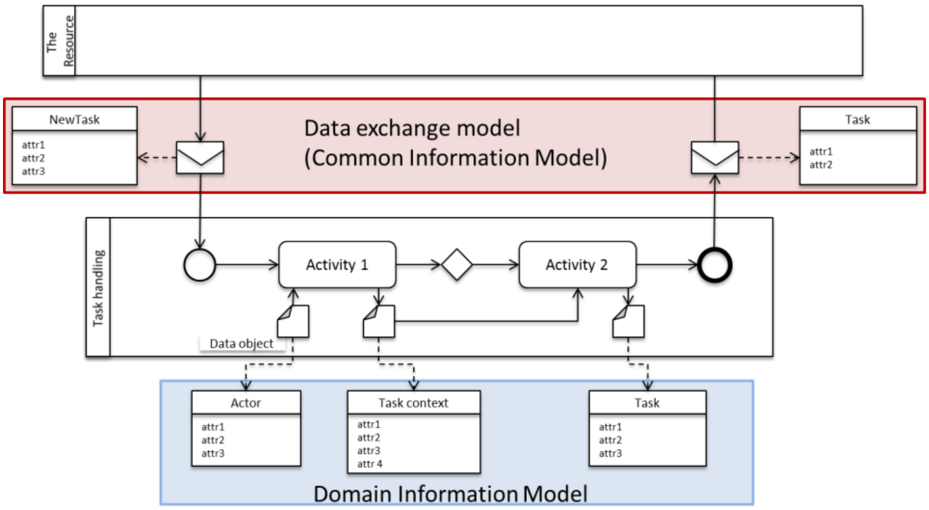

Fig. 6. Process model combined with CIM and DIM models.

Figures 4 and 5 show CIM and DIM as separate models. Figure 6 has combined the two to show how the Common Information Model and the Domain Information Model both relate to the process model. CIM and DIM will always talk about some of the same entities, but it is not required to be in the same way.

There are no restrictions to how DIM represents data, but to avoid complex transformations it is recommended that the CIM and DIM representations of a given entity are as similar as possible. Working with both in the same diagram will make this easier to achieve.

In addition to the combined knowledge models CIM and DIM will have their own diagrams showing them as standard UML class models. But these are primarily derived from the work done in the combined process and information models.

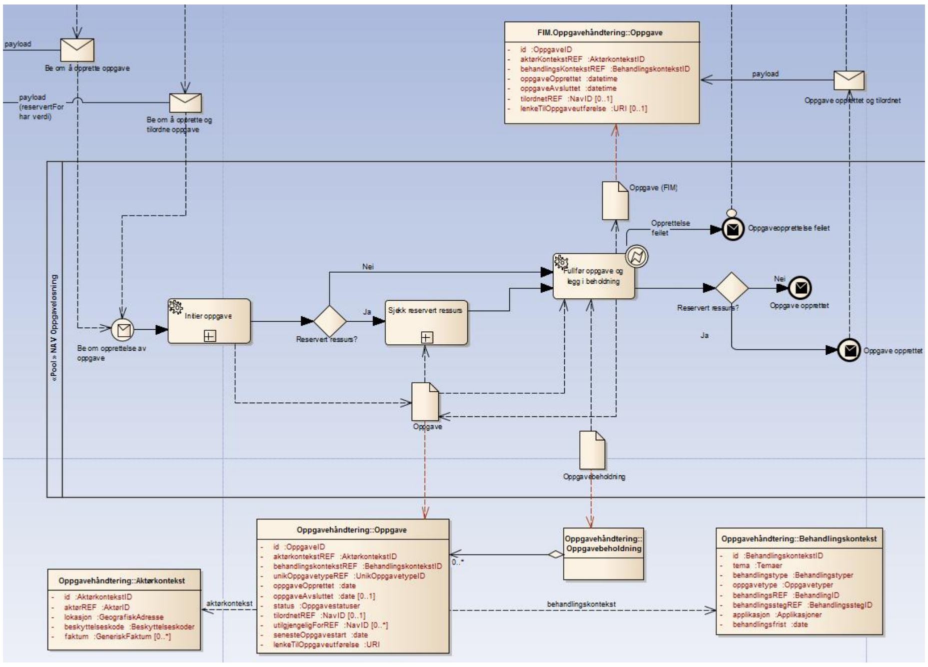
payload
Be om å opprette oppgave
payload
(reservertFor
har verdi)
Be om å opprette og
tilordne oppgave
FIM.Oppgavehåndtering::Oppgave
- id :OppgaveID
- aktørKontekstREF :AktørkontekstID
- behandlingsKontekstREF :BehandlingskontekstID
- oppgaveOpprettet :datetime
- oppgaveAvsluttet :datetime
- tilordnetREF :NavID [0..1]
- lenkeTilOppgaveutførelse :URI [0..1]
payload
Oppgave opprettet og tilordnet
«Pool» NAV Oppgaveløsning
Oppgave (FIM)
Opprettelse
feilet
Oppgaveopprettelse feilet
Nei
Fullfør oppgave og
legg i beholdning
Be om opprettelse av
oppgave
Initier oppgave
Reservert ressurs?
Ja
Sjekk reservert ressurs
Reservert ressurs?
Nei
Oppgave opprettet
Ja
Oppgave opprettet
Oppgave
Oppgavebeholdning
Oppgavehåndtering::Oppgave
- id :OppgaveID
- aktørkontekstREF :AktørkontekstID
- behandlingskontekstREF :BehandlingskontekstID
- unikOppgavetypeREF :UnikOppgavetypeID
- oppgaveOpprettet :date
- oppgaveAvsluttet :date [0..1]
- status :Oppgavestatuser
- tilordnetREF :NavID [0..1]
- utilgjengeligForREF :NavID [0..*]
- senesteOppgavestart :date
- lenkeTilOppgaveutførelse :URI
0..*
Oppgavehåndtering::
Oppgavebeholdning
Oppgavehåndtering::Behandlingskontekst
- id :BehandlingskontekstID
- tema :Temaer
- behandlingstype :Behandlingstyper
- oppgavetype :Oppgavetyper
- behandlingsREF :BehandlingID
- behandlingsstegREF :BehandlingsstegID
- applikasjon :Applikasjoner
- behandlingsfrist :date
Oppgavehåndtering::Aktørkontekst
- id :AktørkontekstID
- aktørREF :AktørID
- lokasjon :GeografiskAdresse
- beskyttelseskode :Beskyttelseskoder
- faktum :GeneriskFaktum [0..*]
aktørkontekst
behandlingskontekst

Fig 7. Real-world example of a process model combined with CIM and DIM models.

4 Conclusion

The novelty of the proposed requirements engineering approach as well as the usage experience so far is summarised below.

4.1 Related work / added value

The new approach presented in this paper allows us to build knowledge models that support the development process from conceptual to logical design models, allowing us to combine several dimensions in the same view.

The approach described is similar to the use of holistic and integrated (sub)models in enterprise modelling [12] [13], with two major differences. Firstly, the domain here is not “enterprise modelling” as such, but requirements engineering as part of information system development. Secondly, the approach is based on integrating industry-standard modelling languages (BPMN and UML); modelling languages that were not designed for each other, but which currently are being used in the industry at a large scale.

4.2 Experience with the approach

LNOP has four major streams of IS development, and the approach was crafted in one of these streams and later has spread to two of the other streams. For each of these streams, there is a set of model producers (representing the customer, partly staffed with contractors) and an audience of model consumers (from the contractors). In the originating stream, there were two full-time process modelers, and 7-8 information modelers (dealing with the CIM, for all streams). The model consumers roughly are split into two groups, service developers and application developers. Service development employs 3-4 developers at the customer side (related to CIM) and 4-5 developers at the contractor side (related to DIM and consuming CIM services). The application developer model audience is between 10 and 20 developers per IS development stream. In principle, behind the scenes there is one central information model and one central process model that are adapted to their audiences in tens of "model diagrams" each representing a particular view of the underlying holistic models. These model diagrams are not "uncoordinated sketches", they are all mutually consistent views of the central model. The described modeling approach for requirements engineering in LNOP so far has been considered vital for the purpose of coordinating requirements engineering and development work – within the streams and between the streams.

References

1. The Open Group: The Open Group Group Architecture Framework (TOGAF), version 9.1. Internet (2011), http://pubs.opengroup.org/architecture/togaf9-doc/arch/
2. The Open Group: Service-Oriented Architecture, Architecture for Boundaryless Information Flow (2011), http://www.opengroup.org/subjectareas/soa
3. Erl, T.: Service Oriented Architecture. Concepts, Technology, and Design. Prentice Hall (2005) ISBN 0-13-185858-0
4. Service-Oriented modelling (SOMA), Wikipedia (2013), http://en.wikipedia.org/wiki/Service-oriented_modeling
5. Hohpe, G., Woolf, B.: Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions. Addison-Wesley (2003) ISBN 0-321-20068-3
6. Sutherland, J.V., Schwaber, K.: Business object design and implementation. In: OOPSLA'95 workshop proceedings, p. 118. The University of Michigan (1995) ISBN 3-540-76096-2
7. Schwaber, K., Beedle, M.: Agile software development with Scrum. Prentice Hall (2002), ISBN 0-13-067634-9
8. Documentation for JIRA 6.0, Atlassian, 2012, https://www.atlassian.com/software/jira
9. Documentation for CONFLUENCE 5.1, Atlassian, 2012, https://www.atlassian.com/software/confluence
10. Business Process Model & Notation (BPMN), OMG, 2013, http://www.bpmn.org/
11. UML (Unified Modeling Language) Resource Page, OMG, 2013, http://www.uml.org/
12. Lillehagen, F., Krogstie, J.: Active Knowledge Modeling of Enterprises, Springer (2008), ISBN 978-3-540-79416-5
13. The Open Group: Archimate, version 2.0 Internet (2013), http://www.opengroup.org/subjectareas/enterprise/archimate/