



Project title:

**Networked Enterprise transFormation
and resource management in Future
internet enabled Innovation CloudS**

Acronym: NEFFICS

Project No: 258076



THEME: ICT-2009.1.3
Internet of Things and Enterprise
Environments

Deliverable D5.4

Case Management Process Models and Methods

Work Package 5

Date: 2012.10.31

Version: 1.0

Leading partner:	Cordys
Editor(s):	Henk de Man
Author(s) (alphabetically):	Henk de Man and Brian Elvesæter
Dissemination level:	Public

Versioning and contribution history

Version	Date issued	Description	Contributors
0.1a	2012.01.09	First table of contents (ToC).	Henk de Man (Cordys)
0.1b	2012.01.18	Revised table of contents (ToC).	Brian Elvesæter (SINTEF)
0.1c	2012.10.29	Text-complete version, as complement to the CMMN final submission	Henk de Man (Cordys)
1.0	2012.10.31	Updated based on review comments	Henk de Man (Cordys)

Peer review summary

Internal review 1			
Reviewer	Jasper Lentjes		
Received for review	2012.10.30	Date of review	2012.10.30

Internal review 2			
Reviewer	Jens-Christian Bjerkek (Induct)		
Received for review	2012.10.30	Date of review	2012.10.30

Review by Project Manager			
Reviewer	Arne-Jørgen Berre (SINTEF)		
Received for review	2012.10.31	Date of review	2012.10.31

Executive Summary

This document introduces the executable modelling language Case Management Model and Notation (CMMN). It can be considered a complement to the already existing Business Process Model and Notation (BPMN) specification, and is meant to model, and model-based manage, activities that are not part of predefined and repeatable processes, but instead depend on evolving circumstances and ad-hoc decisions by knowledge workers regarding a particular situation, a case.

Development of CMMN has been a core part in NEFFICS, and its specification, as will be submitted to the Object Management Group (OMG) for adoption in December 2012, is part of this document.

In addition to the CMMN specification, the document also clarifies the application of CMMN to use cases in NEFFICS, and provides some insights in ways how CMMN might be used, which might be useful as input to development of a methodology for applying CMMN. Methodology as such is not subject of this document, but is subject to other deliverables in NEFFICS. Technology support for CMMN modelling and CMMN model execution is subject of other deliverables as well.

Table of contents

EXECUTIVE SUMMARY	3
TABLE OF CONTENTS.....	4
1 INTRODUCTION	5
2 CMMN: A NEW MODELLING LANGUAGE	6
2.1 CONCEPT, META-MODEL AND NOTATION.....	6
2.2 EXECUTABLE LANGUAGE	6
2.3 RELATIONSHIP TO PROCESS MODELLING LANGUAGES.....	7
3 ASPECTS OF METHODOLOGY.....	8
3.1 LEARN-AS-YOU-GO.....	8
3.2 EVOLVE FROM LESS TO MORE REPEATABILITY.....	8
3.3 GRADUALLY BUILD PLANNING TABLES BASED ON BEST PRACTICE	9
4 APPLICATION OF CMMN IN NEFFICS.....	10
4.1 INTENDED USE CASE APPLICATIONS.....	10
4.2 CORDYS CASE MANAGEMENT VERSUS CMMN.....	12
5 CONCLUSION	13
5.1 SUMMARY OF RESULTS	13
5.2 NEXT STEPS AND FUTURE WORK	13
6 REFERENCES	14
7 APPENDIX A – GLOSSARY.....	15

1 Introduction

This document introduces the new modelling language Case Management Model and Notation (CMMN).

The document consists of two parts, actually two related documents.

- This document itself, which can be referenced as NEFFICS D5.4 – Part A, and which positions CMMN in the context of NEFFICS.
- The formal description of CMMN, as contained in NEFFICS D5.4 – Part B. It will be submitted to the Object Management Group (OMG), November 12, 2012, with the intention to get it adopted as CMMN 1.0, during the upcoming OMG Technical Meeting in December 2012. Upon adoption, it will be published by the OMG. The submission process will then be extended into a “finalization” period, starting March 2013, during which feedback from, in particular implementing vendors, will be evaluated and whereby some changes might be made to the specification, via a formal change management procedure. The “finalization” effort will expectedly end September 2013, and will have a “general available” version of the specification as result.

Chapter 2 provides a brief introduction into and positioning of CMMN. Though this document does not aim to provide a balanced CMMN-based case modeling & management methodology, some aspects of methodology are briefly discussed in Chapter 3. Chapter 4 clarifies the intended application of CMMN in NEFFICS.

2 CMMN: A new modelling language

2.1 Concept, Meta-model and Notation

A Case is a proceeding that involves actions taken regarding a subject in a particular situation to achieve a desired outcome.

A Case is different from a process. A process can be defined as “a predefined, fully specified and repeatable sequence or flow of activities and gateways to direct the sequence along alternative paths or through iterations”. The design-time specified model of a process serves as its plan for execution. Process planning is equivalent to process modelling.

The plan to execute a case evolves in run-time and does only exist in run-time, whereas its building blocks are defined in design-time (as part of the model). The plan is normally incomplete at case-start, and will evolve during execution of the case, based on case worker decisions (“planning”), as guided by the system based on available information in the case file.

In the context of case management, planning is about “adding elements to the plan, in run-time, by case workers, to their discretion, with the intent to adapt the plan, based on new insights and information that has become available to the case.”

Information that has become available to the case is stored in the case file, being the “collection of data about the Case”. Documents and other unstructured or structured data about a case are captured and referenced in the case file for decision-making by case workers, and as basis for system-provided guidance to and management of case workers.

A CMMN model defines the plan of a case at case instance creation (or initiation of a case). This might be considered the “default” or “initial” plan. A CMMN model might (and often will) also define “discretionary” elements, available to case workers, to their discretion, to plan-as-they-go, or to extend/advance the plan as part of their case work, while going. Case models contain the “seeds” for evolving case plans therefore.

NEFFICS D5.4 – Part B explains the purpose of CMMN and the concept of CMMN-based modeling and management of cases. It provides complete and detailed normative semantics, including both a meta-model and graphical notation (diagramming technique), as well. The reader should refer to NEFFICS D5.4 – Part B for explanation and detailed understanding of CMMN concepts, meta-model and notation elements. An overview of definition of some concept terms is provided in the Glossary in Appendix A for convenience.

A leading principle for the development of the notation has been to avoid confusion between the two related modeling languages, CMMN and BPMN, especially in situations where they are used in combination, such as in a tool that implements both, in integrated fashion. In situations of semantic overlap (e.g. both specifications deal with activities and events), similar shapes (symbols) are used. In situations where semantics is CMMN specific, and purposely different from BPMN semantics, shapes are used that do not have counterparts in BPMN (such as for sentries, stages, plan fragments and milestones).

2.2 Executable Language

The specification, as contained in NEFFICS D5.4 – Part B has, in fact, two main parts:

- A design-time part: normative meta-model and notation
- A run-time part: normative execution semantics.

CMMN is an executable modelling language, and aims to provide a complete specification of run-time semantics. It does that by normatively specifying lifecycles (i.e. states and transitions between them) of both information elements (instances of case file items) and elements of case plan models (instances of tasks, states, event listeners and milestones). It also specifies how transitions in these lifecycles influence each other, which user interactions are expected in which states of them, and when the various types of rules in CMMN are evaluated based on states in these lifecycles.

The CMMN specification allows tool vendors to provide both CMMN modelling support and a CMMN “engine” (run-time framework) to interpret CMMN models. Development of tool support for CMMN is not subject of this document. It is subject of deliverables in work package WP7 in NEFFICS.

2.3 Relationship to process modelling languages

CMMN is complementary to existing process modelling languages, and in particular to BPMN. The CMMN specification includes a representation of processes, abstracted from any particular process modelling specification. It does that in order to provide a means to invoke processes, from cases. CMMN specifies cases in a way that interaction is possible between a case and its environment. For instance, BPMN processes may update information in the case file of cases, and thereby trigger follow-up activity in these cases. CMMN-based cases and (BPMN-) based processes are interoperable.

3 Aspects of Methodology

This chapter will not provide a CMMN modelling methodology, and neither a blueprint of how CMMN can be applied in broader methodologies, but will just be limited to providing some understanding of the typical context in which CMMN can be used, and of some aspects of its use.

3.1 Learn-as-you-go

Applying BPMN-like process models require that business analysts have thorough understanding of the process, and that they model the complete “process plan”, including all “flow paths” that might be required, in design-time. People or “systems” in run-time execute what has been modelled.

CMMN follows a different paradigm. A CMMN model will work, is valid, and can be interpreted in run-time, without any upfront modelling of sequences of activities.

A business analyst is assumed to, minimally, model the following, in design-time:

- The structure of the information (“documents”) that the case is assumed to require or produce.
- The roles that perform work in the cases and that maybe authorized to decide on “follow-up” work during the case.
- A non-exhaustive, and maybe only preliminary, set of tasks that might be or become relevant in the context of the case. These can be included as “discretionary” tasks in a so-called planning table.

It is expected that a business analyst, might also want to model some main “episodes” (or stages) and milestones of the case.

This is sufficient basis to start applying case models, and to create and execute case instances accordingly.

During execution, probably also based on observing a from then on growing set of historic cases, insights will emerge in effective combinations of tasks, milestones, event-listeners, etc., as well as in rules that further regulate their applicability or behaviour. Most of these might just be added as “discretionary” tasks and plan fragments (or stages) to planning tables, without any impact on existing and active case instances in the system. Some tasks or stages might be included in the model as part of the “default” plan for next instances to create.

Might it happen that some trajectories (or sub-proceedings) of as-executed cases appear to be structured enough to standardize them as processes, then these processes can be included in the model as process tasks, i.e. tasks that invoke a process (as specified by BPMN).

3.2 Evolve from less to more repeatability

The discussion in 3.1 implies that case models can be positioned on a scale between two extremes:

- Case models that can be executed but that do not enforce any repeatability.
- Case models for which the “default” plan fully specifies the entire case proceeding, and are completely repeatable therefore.

Most case models will be positioned in between these extremes, and many of them will start from a position relatively close to the first (“non-repeatable”) extreme, and will gradually evolve into more repeatability. But probably very few will ever reach the (“repeatable”) extreme.

Evolving to more repeatability can be achieved by extending case models as follows:

- Defining the “default” plan (i.e. elements in the case plan model that will be part of the case plan on case instance creation) in more detail.
- Defining more, larger and more refined “discretionary” plan fragments or stages.
- Applying more rules on either the “default” plan elements, “discretionary” plan elements, or both.
- Including more process tasks, that invoke “complete” processes.

3.3 Gradually build Planning Tables based on best practice

Probably the best, i.e. less start-up investment requiring and most “relaxed” way of applying CMMN-based case management, is to start from underspecified case models, and build best practice knowledge by adding “discretionary” elements to planning tables. When insights grow into what is repeatable, and once there is consensus about that, certain “discretionary” elements might be promoted to “fixed” elements, i.e. elements that come into the plan on case instance creation.

4 Application of CMMN in NEFFICS

4.1 Intended use case applications

Figure 1 represents model-driven business innovation as an iterative process.

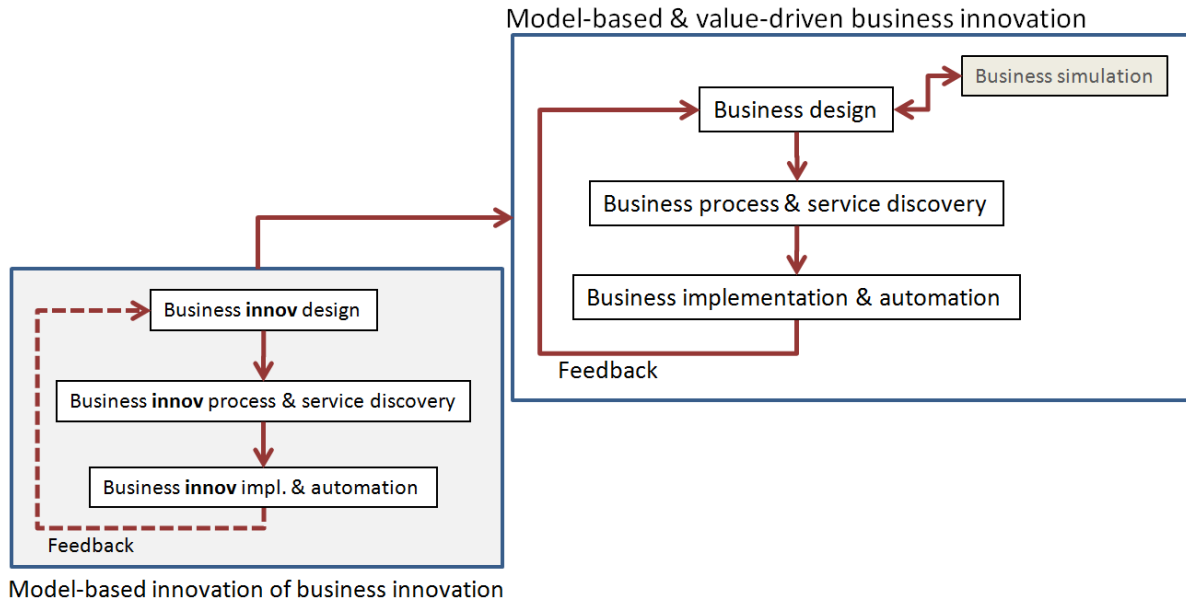


Figure 1: Model-driven business innovation

Two cycles of innovation can be distinguished:

- The process of “business innovation”. Scope of business innovation could be a “business model” (as e.g. defined by NEFFICS D4.1 (2011) and NEFFICS D4.3 (2012)), or a part of the business system that supports one or more “business models”.
- The process of “innovation of business innovation”. In this context, the process of innovation may itself be analyzed, modelled, and managed based on executing models, in order to make it more transparent, manage-able, measurable and enabled for continuous improvement.

The application of models, in the cycle of “business innovation” is as follows:

- Analyse and design the business, based on Value delivery models (VDML). NEFFICS D1.4 (2012) demonstrates that, for instance in the context of the “AGR” business model. NEFFICS D3.3 – Part B provides the VDML specification. VDML is explained from a use case perspective in NEFFICS D3.3 – Part A.
- So-called “capability methods” in VDML-based business designs maybe transformed into and associated with business process models (following the BPMN (2011) specification) or case models (following the CMMN specification as provided by NEFFICS D5.4 – Part B). Case models are used when the process is majorly human and/or event-driven.
- CMMN and BPMN models can be automated based on model execution, which will then result in business data and measurement feedback which may feed next cycles of analysis and innovation.

In the context of WP1 (see NEFFICS D1.4 (2012)), intended use case examples of capability methods, implemented as “cases” are:

- Product change management
- RFQ-to-Order processing
- Order-to-Production processing

- Order change management
- Quality control

The application of models, in the cycle of “innovation of business innovation” is similar, and consists of the following phases:

- Analyse and design the part of the business system that is concerned with “business innovation”, based on Value delivery models (VDML). NEFFICS D1.3 (2012) contains a series of VDML-based role-collaboration diagrams, also known as “value network diagrams”, to demonstrate this. As an example, Figure 2 represents the “system of business innovation” as “value network”.

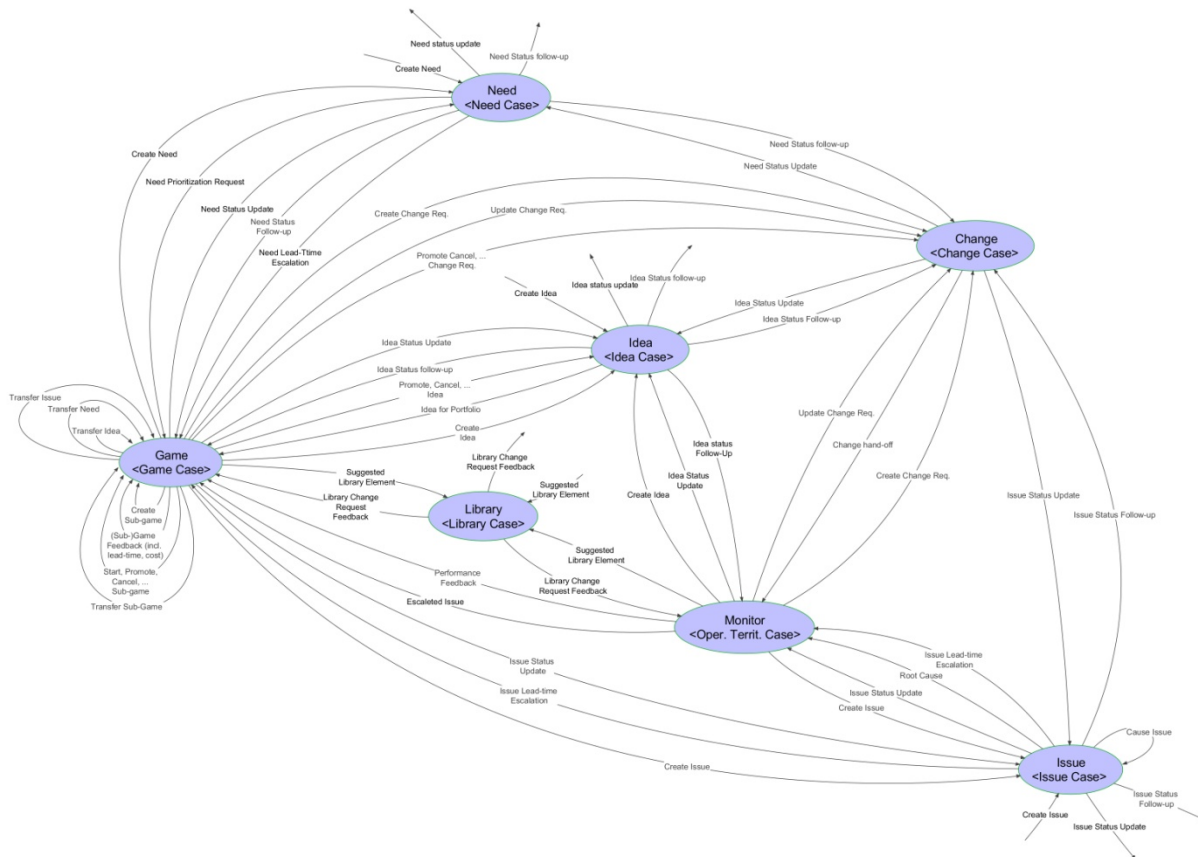


Figure 2: Role-collaboration view on business innovation

- Roles in the collaboration in Figure 2 can be filled by “capability methods”, which are assumed to have their counterparts as CMMN-based case models.
- Execution of case models will provide model-based management (including guidance and monitoring) of innovation work. Results can serve as feedback for potential further innovation of the “process of business innovation”. Tasks in CMMN-based case models will, in this context, be implemented based on innovation services as provided by WP6, where appropriate.

Assume-ably the NEFFICS roadmap, related to implementation, demonstration and validation of the two related iterative processes in Figure 1, is as indicated in Figure 3.

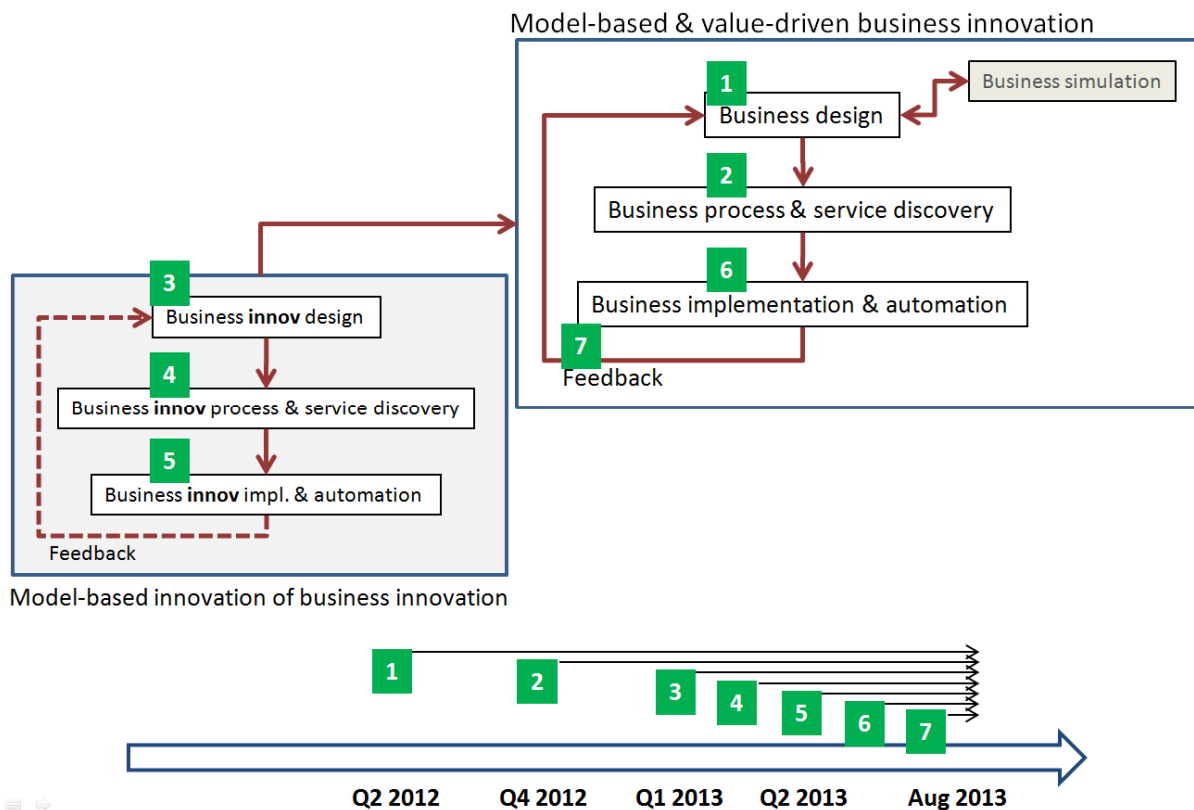


Figure 3: Use case phasing of model-driven business innovation

As Figure 3 suggests, NEFFICS currently focuses on applying VDML to analysis and modelling of “business models” (and their supporting parts of the business system). Preparatory work has been conducted to investigate how VDML and CMMN can be applied to model and manage the process of innovation, but this work has not yet resulted into implemented and integration application.

Subsequent WP7 deliverables will deal with technology support for VDML and CMMN. Subsequent WP1 deliverables, and probably WP2 deliverables, will deal with applying VDML and CMMN to support business innovation and its innovation.

4.2 Cordys Case Management versus CMMN

Technology support for VDML and CMMN modelling, as well as execution of CMMN models is provided as a result of WP7 in NEFFICS. The Cordys Business Operations Platform (BOP) is being extended for this purpose. WP7 deliverables will document these results.

WP7 deliverables will also clarify how Cordys Case Management support relates to (and is compliant with) the CMMN specification.

5 Conclusion

5.1 Summary of results

CMMN-based case models define models that support case workers in planning case work as-they-go, and is particularly adequate as basis for activity management, when the human decision factor and/or the need to respond to events are dominant.

“Discovering” CMMN-based models differs from “discovering” process models, such as BPMN-based process models, in the following ways:

- Process modeling, such as BPMN-based process modeling, requires upfront knowledge of “everything that can happen”, and the “sequences according to everything need to be performed”. This is not required at all for CMMN modeling.
- Changing process models, such as BPMN-based process models, has significant impact on running instances of the process model. This is neither the case for CMMN models.

This will make CMMN-based modelling a relatively low-barrier effort, requiring relatively low upfront investment, and makes it particularly useful in the context of continuous innovation.

NEFFICS intends to apply CMMN-based models, in the context of use case(s), for the purpose of both innovating business, and innovating the process of innovation itself.

A final submission of the CMMN specification to the OMG is due in November 2012.

5.2 Next steps and future work

Follow-up work is required in the context of several work packages.

As part of WP1, and as far as appropriate and feasible also as part of WP2, the application of VDML will be broadened from modelling “core business”, to modelling the process of innovation itself, and CMMN will be applied to both.

As part of WP5, model-based methodology will be further elaborated by incorporating the use of VDML models and CMMN models.

As part of WP7, the work package that is involved in developing modelling support for, amongst others, CMMN and VDML, and execution support for CMMN, the following is required, in the context of case management:

- Integration of VDML models with BPMN models, CMMN models and web-service interfaces. Where appropriate and feasible also: transformation from parts of VDML models to BPMN models and CMMN models.
- Demonstrating compliance of Cordys Case Management with CMMN, and where possible and feasible in the context of NEFFICS, also improve compliance.

These subjects will expectedly be covered in NEFFICS D7.4 (forthcoming, and due September 2013).

6 References

BPMN, *Business Process Model and Notation*, Version 2.0, Object Management Group, Release Date: January 2011, <http://www.omg.org/spec/BPMN/2.0/> .

Fowler, M., *UML Distilled: A Brief Guide to the Standard Object Modeling Language*, Addison-Wesley, 2004

Harmon, P., *Business Process Change: A Guide for Business Managers and BPM and Six Sigma Professionals*, Morgan Kaufman, 2007.

NEFFICS DoW, *Description of Work*, Annex 1 to Proposal 258076 (STREP), April 2010.

NEFFICS D1.3, *Initial Release of the Virtual Extended Factory*, April 2011.

NEFFICS D1.4, *Second Release of the Virtual Extended Factory*, September, 2012 .

NEFFICS D3.3 – Part A, *Value Delivery Model and Methods*, June 2012.

NEFFICS D3.3 – Part B, *Value Delivery Modeling Language (VDML)*, Version 0.2, Object Management Group, Submission for May 21, 2012.

NEFFICS D4.1, *Baseline for Networked Innovation Models*, April 2011.

NEFFICS D4.3, *Models for Network-based Open Business Model Innovation*, June 2012.

NEFFICS D5.4 – Part B, *Case Management Model and Notation (CMMN)*, Version 1.0, Object Management Group, Submission for November 12, 2012.

NEFFICS D7.4, *Extension of platforms to support continuous evolution and innovation of networked enterprises*, Forthcoming, Due September 2013.

SoaML, *SOA Modeling Language*, Version 1.0, Release Date: March 2012, <http://www.omg.org/spec/SoaML/1.0/> .

UML, *Unified Modeling Language, Superstructure*, Version 2.4.1, Object Management Group, Release Date: August 2011, <http://www.omg.org/spec/UML/2.4.1/Superstructure/PDF/> .

7 Appendix A – Glossary

Case. A proceeding that involves actions taken regarding a subject in a particular situation to achieve a desired outcome.

Case file. The collection of data about the Case. Documents and other unstructured or structured data about a Case are captured and referenced in the case file for decision-making by case workers, and as basis for system-provided guidance to and management of case workers.

Event. Something that “happens” during the course of a case. According to CMMN, any event can be represented by a transition in the lifecycle of an element that is tracked in the context of a case instance.

Event listener. A plan element, with track-able lifecycle in run-time, used to capture design-time defined timer events or user events. These are events that cannot be represented by transitions in the lifecycles of the other run-time track-able elements in CMMN (information elements, tasks, milestones and stages).

Lifecycle. Configuration of states, and transitions between them, providing the basis for tracking of progress of a run-time element. CMMN defines lifecycles for instances of information elements in the case file, and for instances of tasks, stages, event listeners and milestones in the plan.

Milestone. A plan element, that represents an achievable target, defined to enable evaluation of progress of the case. The lifecycle of a milestone instance indicates whether or not the target has been achieved.

Plan. A configuration of (instances of) tasks, event listeners and milestones, typically organized by (instances of) stages, meant to handle the case. The plan of a case does only exist in run-time, whereas its building blocks are defined in design-time (as part of the model). The plan is normally incomplete at case-start, and will evolve during execution of the case, based on case worker decisions (“planning”), as guided by the system based on available information in the case file.

Plan fragment. A configuration of tasks and/or event listeners and/or milestones, possibly organized by stages and possibly dependent on each other and/or on information events from the case file, which configuration serves as re-usable and prototypical “fragment” of a case plan

Planning. Adding elements to the plan, in run-time, by case workers, to their discretion, with the intent to adapt the plan, based on new insights and information that has become available to the case.

Planning table. A set of elements that can be considered for planning in a certain context, being a stage or a human task, and which is presented in table format. Next to these elements it also contains rules that regulate their applicability and reference to roles that are authorized to plan them to their “discretion”.

Process. A predefined, fully specified and repeatable sequence or flow of activities and gateways to direct the sequence along alternative paths or through iterations (this is a variation to the BPMN definition of process in BPMN (2011)).

Sentry. A criterion to enable or terminate a task or stage or achieve a milestone. It is defined as combination of zero or more events and zero or one condition. A sentry, as criterion, is satisfied when the event(s) has (have) occurred, and the condition has evaluated to “true”.

Stage. A plan fragment that, when planned, gets itself a run-time representation in the plan, so that its progress and completion can be tracked based on its lifecycle. A stage can also serve as context for planning new elements into it during execution. Stages might serve as “episodes” of cases.

Task. An atomic unit of work. Tasks serve as human tasks, or to invoke processes or other cases.