Design of the process of developing DCM's with regard to eMeasures

A case study at a large teaching hospital in the Netherlands

Rick Beukeboom

UMCG, Programma Nieuw EPD
RUG, Master Technology & Operations Management

rijksuniversiteit groningen

Groningen, februari 2015

Studentenbureau UMCG

Universitair Medisch Centrum Groningen

Design of the process of developing DCM's with regard to eMeasures

A case study at a large teaching hospital in the Netherlands

Groningen, februari 2015

Auteur	Rick Beukeboom
Studentnummer	s1867938
Afstudeerscriptie in het kader van	Faculteit Economie & Bedrijfskunde Technology & Operations Management Rijksuniversiteit Groningen
Opdrachtgever	G.A.T. Lesman-Leegte Nieuw EPD, UMCG
Begeleider onderwijsinstelling	dr. H. Balsters Faculteit Economie & Bedrijfskunde Rijksuniversiteit Groningen
Begeleider UMCG	drs. A. Kuipers Nieuw EPD, UMCG

Trefw Business Process Modeling Notation, eMeasure, Detailed Clinical Model

PREFACE

This thesis is the end product of my master studies Technology & Operations Management at the University of Groningen.
It was both a challenging as an interesting project. Due to the great support of the Large Teaching Hospital in the Netherlands (LTHN), the project that has been done felt very welcomed and appreciated. I would like to thank all staff of the LTHN that helped during this project, with special thanks to the DCM analyst who inspired me most. Furthermore, I also want to thank my fellow students, Pim van de Laar and Peter Martena for their cooperation and input during discussions, which were of great value for my part of the project.
At last but definitely not least, I would like to thank my first supervisor dr H. Balsters for his guidance and support for the last couple of months and his effort to meet almost weekly for the sake of this project. Additionally, I would like to thank my co-assessor dr. W.M.C. Van Wezel for assessing my thesis.

TABLE OF CONTENTS

ABSTRACT

Due to the increasing use of electronic patient records and other health care information technology, an increase in requests to utilize the data that comes forth from these is seen. One of these requests relates to the data regarding the quality of care to gain better insights for further improvements in health care. Health Level 7 created a standard to format quality performance indicators in so-called 'eMeasures' for effective communication purposes. A standardized method for developing these eMeasures however is not given. A large teaching hospital in het Netherlands is currently developing an innovative method to generate eMeasures with the use of Detailed Clinical Models. This project will develop process models using Business Process Modeling Notation to illustrate the process that makes use of Detailed Clinical Models to generate fast and reliable eMeasures.

1 INTRODUCTION

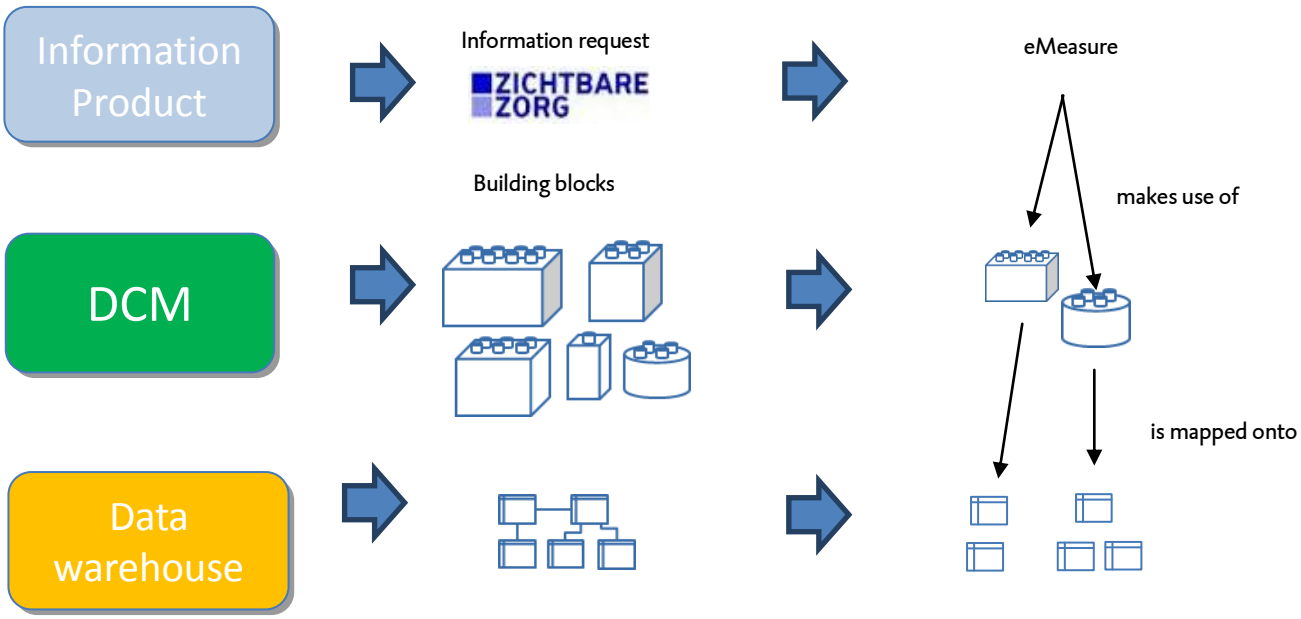
Currently, the health care sector experiences a big change with the introduction of the Electronic Health Record (EHR). An EHR is a system that includes patient-focused, electronically documented information about the health care of individuals, focused on activities and processes directly related to patient health care (Boll, 2006). Due to the increasing use of electronic health records and other health care information technology, an increase in requests to utilize the data that comes forth from it is seen (Goossen et al, 2010). Example is the government that wants to gain better insights in the quality of care to have as basis for policy making.

Nowadays, in Dutch hospitals including a Large Teaching Hospital in the Netherlands (LTHN), these information requests of, inter alia the government, are tackled by directly translating the information request to the database. In other words, one receives an information request and looks up the necessary information in the hospital's database and tries to answer this request at its best. It is a method that works, but it is an immense job to come up with the requested information and has some serious other downsides. One is, that due to the lack of a generic functional layer in between, the information requests are not able to be compared between different data warehouses. Moreover, quality checks where this data comes from and interchangeability of information request are not or barely possible. To overcome these problems, a new method is required.

A way to create insight in the quality of health care is with the use of standardized eMeasures defined by Health Level 7 (HL7). HL7 is one of the leading standards for exchange of clinical and administrative data among healthcare information systems (Hooda et al. 2004). An eMeasure is a health measure encoded in the Health Quality Measures Format (HQMF), which is a standard for representing a health quality measure as an electronic document (HL7 V3, 2014).

Although HL7 gives a formalized standard on how to format an eMeasure as is mentioned above, HL7 does not provide a standard on how to develop such eMeasures. Currently a LTHN is in the initial stages of developing an innovative method to generate eMeasures with the use of Detailed Clinical Models (DCM). DCM is a new way to structure health care information which can be regarded as reusable building blocks or as a generic functional layer where eMeasures can be connected onto. Herein, domain expertise, data specification and terminology are combined in information models which enable various technical elaborations. The aim of a DCM is to provide consistent and precise data and terminology specification which are sharable between multiple care providers (Stichting DCM, 2014). To get a better understanding of how a DCM looks like, an example of an information model of a DCM is given in appendix II.

Below a graphical representation is shown of how this new method of developing eMeasures looks like to get a general idea. More detailed explanation will be given later in this thesis.



The main aim of this research project is to design and validate the innovative general process that develops eMeasures with the use of DCM's. This is done by Van de Laar (2015) and this thesis. At first a general overview will be created by both. Subsequently, Van de Laar will focus on the process of the application and creation of an information product and an eMeasure, while the focus of this thesis will be more on the process of developing DCM's related to the eMeasures.

The design of the process models will be done by using Business Process Modeling Notation (BPMN). The main goal of BPMN is to provide a notation that is readily understandable by all business users, from the business analysts who create the initial drafts of the processes, to the technical developers responsible for implementing the technology that will execute those processes, and, finally, to the business people who will manage and monitor those processes (White, 2004). Since BPMN is an easily understandable language for everyone, it is a useful tool for designing process models.

Additionally, due to the easily interpretability characteristic of BPMN, Balsters (2013a) developed a general database design method where he used BPMN to systematically map basic business process models to create data models. Although Balster's method is already successful in practice, this method has to be theoretically evaluated. As Wieringa and Heerkens (2006) state in their paper, validation is often lacking for software engineering papers. In line with this, also the use of BPMN in combination with a method for designing data models needs more validation. This research will provide the BPMN models from which the data models can be created. It can be seen as the secondary objective of this thesis.

The overarching project that is conducted at the LTHN is done by two fellow MSc-students of the University of Groningen and me. The first part of the research is to illustrate the process of developing eMeasures with its related DCM's into process models using BPMN. Subsequently, these models are validated by the stakeholders. This part is executed by Van de Laar (2015) and me as is explained before. Additionally, as second part of this project, these models will be the input for generating ontologies using Object-Role Modeling (ORM). In short, an ontology is a specification of a conceptualization (Gruber, 1992). This part is performed by Martena (2015). The process models and ontologies need to be validated by all three students.

The outcomes of this individual thesis, i.e. the resulting validated process models, will give clear insights of the process of developing DCM's with regard to eMeasures for the organization. On top of that, it may be used as guidelines for other hospitals to implement this process which is on the frontier of pioneering. Finally, the outcomes of the overarching project can be used as supportive data for a forthcoming paper of Huitema and Balsters on process- and data migration.

1.1 RESEARCH OBJECTIVE AND QUESTIONS

The research objective for this thesis is:

Design a validated process for developing DCM's with regard to eMeasures using BPMN and Systems Thinking.

The main research question for this research that can be derived from this objective is:

How can validated process models be derived from the business processes related to the development of DCM's with regard to eMeasures?

In order to attain to this objective and answer this research question, several sub-questions have to be answered first. These questions are listed below.

- How to build BPMN models using Systems Thinking?
- Who are the stakeholders and what are their goals and activities related to the process?
- How to validate process models of business processes?
- Do the designed process models meet the stakeholder's expectations and satisfaction?

1.2 ACADEMIC AND PRACTICAL RELEVANCE

The academic relevance of this research is that it comes up with a validated design of an innovative process within health care to systematically develop eMeasures using DCM's. So far, this does not exist. Additionally, this is a case study at a single hospital and if it turns out to be successful, it can be generalized to other hospitals. Furthermore, because the process to be modeled in BMPN is expected to be complex, it will give insights whether this notation is capable of handling these kinds of complex processes.

Another point to bear in mind is that it contributes, by creating BMPN models, to the validation of a general proposed method where BPMN models are used to systematically map basic business process models to create ontologies.

The practical relevance of this research is that it contributes to the understanding of how eMeasures and its related DCM's are developed. This process is namely a new and possibly better way of tackling incoming information requests and is expected to be very efficient. For the Management Team (MT) in order to convey this message, BPMN models are a suitable tool to do this. One could even state that it has societal relevance to care in general, due to its innovativeness in its sector.

2 THEORETICAL BACKGROUND

The theoretical background section is split up in multiple parts. At first, requirements engineering (2.1) will be discussed, followed up by business process modeling (2.2) and Systems Thinking (2.3) which is directly interrelated. Next, ontology design (2.4) is shortly described which is not part of this thesis but does need some attention for the understanding of the overarching project. Section 2.5 describes ways to test validity.

2.1 REQUIREMENT ENGINEERING

According to Selvakumar & Rajaram (2011), "Requirement Engineering (RE) deals with the requirements of a proposed solution and handles conflicting requirements of the various stakeholders and is critical to the success of a project". In other words, they define the critical success factors for a project. Another definition given by Van Lamsweerde (2000) is that "RE is concerned with the identification of the goals to be achieved by the envisioned system". We can therefore state it is important for proposing a solution, to first identify the stakeholders and their goals. Subsequently, the related critical success factors have to be investigated and any conflicting requirements have to be analyzed. An increasing number of stakeholders increase the chances of conflicting requirements and therefore also choices to be made to solve the conflicts.

Li, Eberlein & Far (2004) classified RE into Functional Requirements (FR) and Non Functional Requirements (NFR). FR deal with the requirements that affect the system's functionality, whereas NFR deal with the requirements that affect the system's constraints. To put it differently, FR are characterized by simple language, specific to a business requirement and it describes the "What should the system do?" question (Selvakumar & Rajaram, 2011). NFR identifies user or system constraints which are characterized by features such as user-friendliness, response time, portability, reliability and maintainability (Selvakumar & Rajaram, 2011). NFR describes more the "How should the system do?" question.

2.2 BUSINESS PROCESS MODELING

Aguilar-Saven (2004) states that a business process is the combination of a set of activities within an enterprise with a structure describing their logical order and dependence whose objective is to produce a desired result. Business process modeling enables a common understanding and analysis of such a business process. To do that, a wide variety of business process notations are existing such as Petri nets, BPEL (Business Process Execution Language), Gantt Chart and BPMN (Business Process Modeling Notation) to name a few (Aalst, 2012). The latter one, is currently the leading standard in the frame of business processes and workflow modeling languages, i.e. the state-of-the-art in the field (Chinosi & Trombetta, 2012). It is developed to provide a graphical notation in order to represent a business process in a Business Process Diagram (BPD). Moreover, specific software around BPMN is available in order to translate BPMN-process models to software applications and automated systems.

In this research, only basic BPMN will be considered and used. This entails the following according to Balsters (2014a):

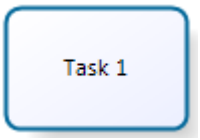
-A business process is organized as a collection of communicating pools. A pool is used to show a process where several participants can be involved which leads to subdivisions (see swim lanes) within the same organization. A depiction of a pool is showed below:

-A pool consists of swim lanes, which are used to organize participants within a process. A swim lane belongs to exactly one stakeholder. The figure below shows a pool consisting of two swim lanes:

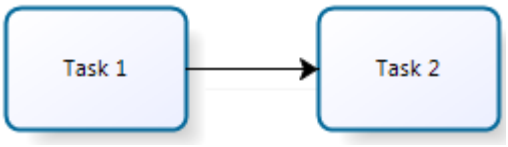
- A process has one start event, and one or more stop events, which are depicted as follows:

Start event:  End event: 

-A process consists of activities/tasks, which shows what is done at what point in the process. The graphical depiction is shown below:



-Activities can be sequenced, which may look like this:



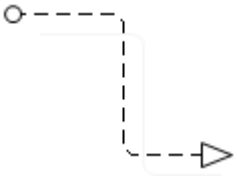
-An activity can lead to a branching to two (or more) activities. This is done by using gateways to show the divergence of sequence flows. Two sorts of gateways are depicted below:

XOR gateway:  Parrell gateway: 

XOR gateway indicates that only *one* of the following paths can be taken. A parrelell gateway means that either *all* following paths are taken simultaneously or *all* incoming branches have to be completed first before continuing the process.

-An activity can be nested. This means that an activity is build up in different levels. One could zoom in/out on an activity. In other words, the process contains sub-processes which are a finer level of detail of that part of the process. When an activity/task contains one or more levels, it is depicted as follows:

-Pools can communicate through messages. A message is shown by a data object. The flow of the information which is shown in a data object is represented by a message flow. Both are shown below:

Data object:  Message flow: 

An example of a Business Process Diagram (BPD) is given in figure 1 where some elements described above are used.

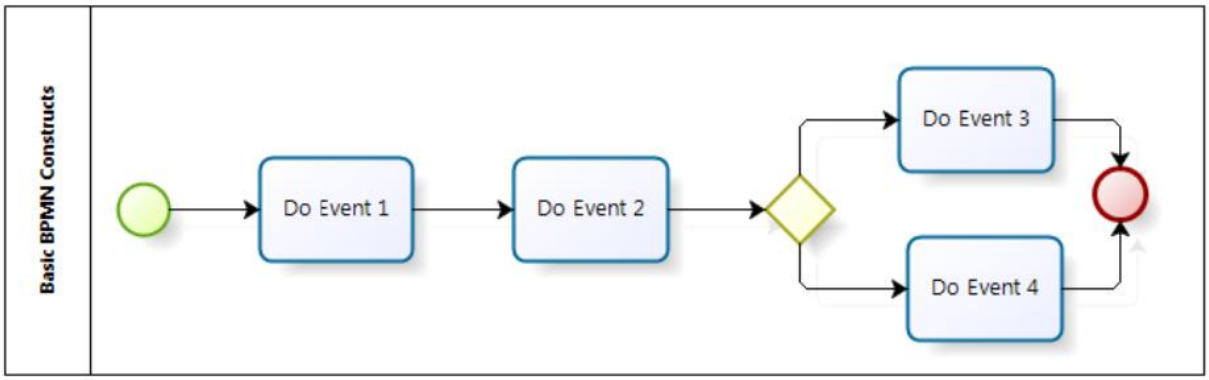


FIGURE 1: AN EXAPMLE BPD BY BALSTERS (2013A)

2.3 SYSTEMS THINKING

In order to come up with these BPD's, three phases have to be gone through according to Systems Thinking of In 't Veld (2006). These three phases consist of:

1. Coding of input
2. Transforming input to output
3. Decoding of output

Coding: Coding is the (high-level) function that prepares 'Input' for 'Transformation' to 'Output'. This is done by changing the given process model into a proper format. During this research this will be in a BPMN format, only employing a set of basic constructs, which is structured according to Systems Thinking principles. Additionally, coding pertains to input quality and input quantity (Balsters, 2014a). This is done by:

-Quality control: check demands with respect to quality of input before it is allowed to be transformed.

-Quantity control: check demands with respect to number of qualified input items before transforming can take place.

Coding is typically the hardest and most time-consuming part in constructing BPD's.

Transforming: The new process model as input is transformed into an ideal data model and yields an associated ideal database (Balsters, 2014a).

Decoding: The ideal database is to be regarded as virtual, and gets its population from an existing (set of) database(s) (Balsters, 2014a).

Systems thinking embodies the thought of first defining the system before one can investigate smaller components of the system (In 't Veld, 2006).

2.4 ONTOLOGY DESIGN

To create more understanding of what an ontology is, first a definition is given by Li, Yang & Wu (2005): 'An ontology is a formal description of a domain of discourse, intended for sharing among different applications, and expressed in a language that can be used for reasoning.'

When designing an ontology for a business domain, a model of it has to be created. According to Halpin & Morgan (2008) the business domain that has to be modeled is called the 'universe of discourse' (UoD). This is the universe (or world) that we are

interested in discoursing (or talking) about. The main challenge of modeling is to describe the UoD clearly and precisely, since errors introduced here, filter through later stages in software development (Halpin & Morgan 2008). The authors of the book '*Information Modeling and Relational Databases*' identify three different information modeling approaches, namely: Entity-Relationship modeling (ER), Object-Oriented (OO) modeling, and Fact-Based Modeling (FBM). Since this latter one will be used during this research, a small description is given below.

2.4.1 Fact-Based Modelling

According to Halpin and Bloesch (1999), Object-Role Modeling (ORM) is currently the most popular fact-based approach to data modeling. ORM is a semantic modeling approach that views the world solely in terms of objects (things) playing roles (parts in relationships). A relationship is shown as a named sequence of one or more role boxes; each connected to the object type whose instances play that role (Halpin & Morgan, 2008). An example of an ORM diagram is shown in the figure below:

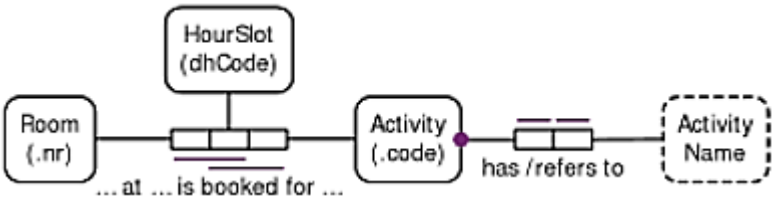


FIGURE 2: AN ORM DIAGRAM FOR ROOM SCHEDULING (HALPIN & MORGAN, 2008)

ORM do not use attributes in its base models, where this is the case in ER. All the facts are expressed by objects playing roles. This often leads to larger diagrams, but due to this attribute-free approach some advantages come along for conceptual analysis. This entails simplicity, stability, and ease of validation (Halpin & Morgan, 2008). Another advantage is that because of the fact that ORM schemas are specified in unambiguous sentences backed up by illustrative examples, it is not necessary for domain experts to understand the diagram notation at all (Halpin & Morgan, 2008).

A drawback of fact-based modeling (FBM) specified in ORM schemes is that it is hard to communicate these schemes to software engineers using UML. Balsters & Doesburg (2012) however, offer a translation from the fact-based modeling (FBM)-specifications to UML class diagrams.

2.5 VALIDITY

According to Karlsson (2009) the following tests must be addressed for any academic research in design science for it to be valid and reliable:

- Construct validity: This is done by letting the interviewees review the drafts and conclusions, by using multiple interviewers and interviewing multiple interviewees. This repeats until the interviewee states that they are correct and valid.
- Internal validity: This is done by matching the outcomes of interviews and interviewees to identify possible causal relationship between them.
- External validity: This is done by checking whether or not the findings can be made generalizable.
- Reliability: This is guaranteed by using a research protocol and storing interviews and it results in a database.

METHODOLOGY

In this chapter the methodology of this research is described. At first the type of research is discussed, followed by its related framework. Next, the specification of the methodology of the overarching project is outlined and finally the specification of the individual project.

3.1 TYPE OF RESEARCH

The research that will be conducted in this project must be labeled as a 'Design Science' type. This is due to the fact that the aim of the project is to create and validate an artifact that solves real-world problems. In other words, design science aims at solving practical-knowledge problems, which has goal to resolve a difference between the way stakeholders experience the world and the way they would like to experience the world (Balsters 2014b).

It should however be taken into account that solving design problems often leads to solving pure-knowledge sub-problems (Balsters 2014b). In our case, an example of these pure-knowledge sub-problems is defining all stakeholders with their goals and corresponding success-criteria. Additionally, the eventual proposed solutions have to be validated on its correctness.

The design of the process models concerning the development of DCM's related to eMeasures will be done during a case-study at a LTHN.

3.2 RESEARCH FRAMEWORK

For filling the gap between theory and practice for practical problem solving, Van Strien (1997) developed the regulative cycle. This cycle will be used as the fundament of the research framework. The regulative cycle of Van Strien (1997) is conceptually illustrated below.

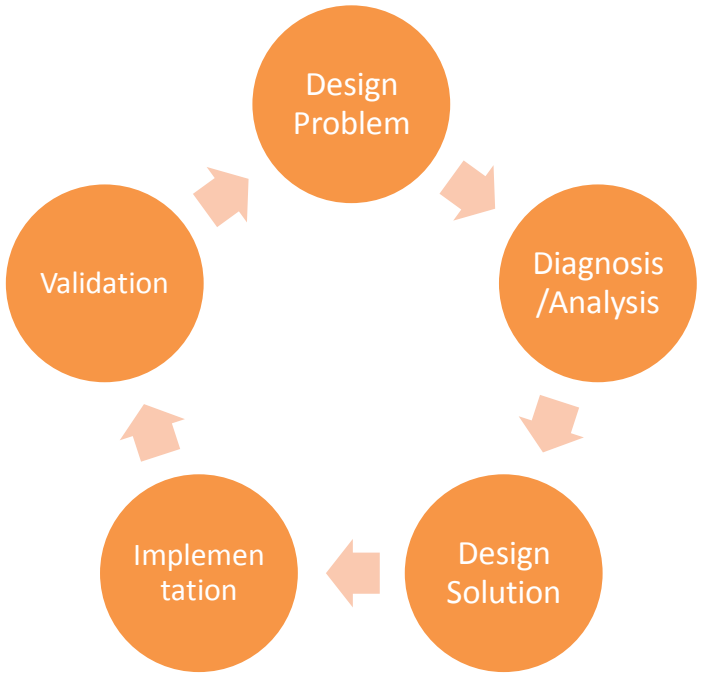


FIGURE 3: REGAULATIVE CYCLE (VAN STRIEN, 1997)

As can be seen from figure 3, the regulative cycle consist of five process phases, namely 'design problem', 'diagnosis/analysis', 'design solution', 'implementation' and 'validation'. To specify these phases in more depth, Balsters (2014b) formulated questions that should be answered to complete each phase. These questions are stated below:

Design problem phase:

- Who are the stakeholders?
- What are the goals for each stakeholder?
- What are the Critical Success Factors (CFS's) for each goal?

Diagnosis/analysis phase:

- What are possible causes of the difficulty of resolving a CSF?
- What are the quality attributes of CSF's?
- Is there an order-dependency in which the CSF's should be treated in order to achieve a properly working solution?

Design solution phase:

- Which solution alternatives are available?
- Can we assemble old solutions to build a new solution?
- Can we (and must we) invent a new solution completely from scratch?

Validation Phase:

- How to design test methods for each CSF?
- Are all CSF's met?
- What are the tradeoffs involved in choosing one solution over the other?
- How scalable is the solution/implementation?
- How well does the solution perform compared to the earlier defined CSF quality attributes?
- Have we encountered new CSF's in the implementation result?

As one might notice, no further questions are mentioned for the 'implementation' phase. This has to do that the implementation is simply a matter of executing what you have designed. The 'validation' phase is in place to verify and validate this execution. Moreover, often the validation of a design solution does not require an implementation first, but can also be done without it (Balsters, 2014b). This is also the case for this research. Therefore a modified regulative cycle will be used during this project, which is depicted in figure 4.

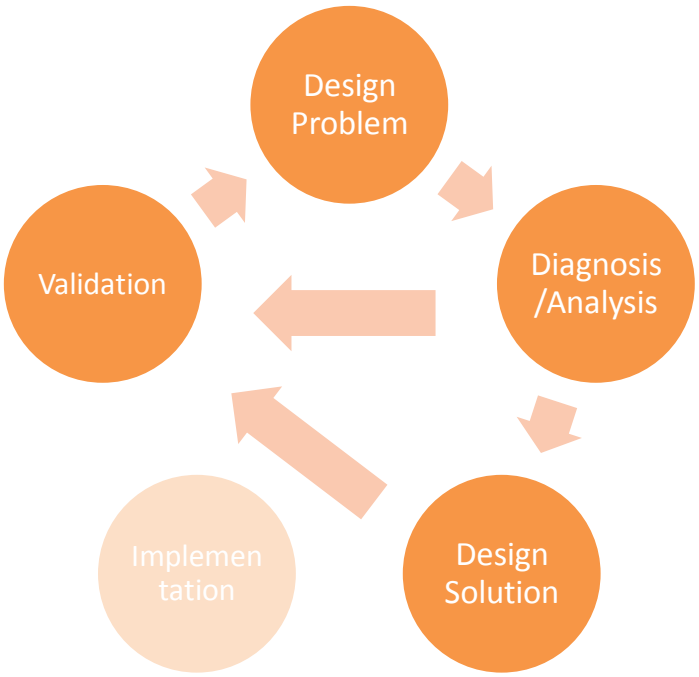


FIGURE 4: MODIFIED REGULATIVE CYCLE

In the modified regulative cycle, it is seen that the implementation phase is skipped, but on the other hand an extra validation is included. With respect to limited resources, time and authorization it is not possible to perform the implementation in the EHR system of the LTHN. The extra validation after the diagnosis/analysis part is included to ensure that the input for the design solution is correct.

It must be noted that when performing design science, the regulative cycle goes through iteratively before arriving at the ultimate design.

3.3 SPECIFICATION METHODOLOGY OVERARCHING PROJECT

This section describes more specified to the case where the three researchers will take their role in the overarching research project. This will be done according to the sequence of steps that have to be undertaken.

3.3.1 Constructing the process models

For developing the business process models in the LTHN the entire regulative cycle has to be walked through. From design problem to validation; and that a few times to arrive at the eventual design. At first data will be gathered from existing documentation how processes flow. In order to get a complete understanding and in case of missing information, structured interviews with stakeholders will be held. Through combining all the information, process models are constructed using BPMN in Bizagi Process Modeler. Furthermore, it is important to keep Systems Thinking in mind when designing process models whereby 'Coding' will usually takes most of time and effort. At last, these process models are validated by the end-users to ensure correctness of the models. This part of the overarching project is performed in parallel by Van de Laar (2015) and Beukeboom (2015) and is more described in detail in section 0.

3.3.2 From process models to ontologies

Once the process models are realized, the design of the ontologies can be started. This part of the overarching project consists of the 'design solution' phase of the regulative cycle. The translation of business process models to data models will be done by the Fact-Based Business Process Modeling (FB-BPM) method described by Balsters (2013a). This method ensures that a corporate data model will be created associated to the given BPMN process models. ORM will be used to create these models. This part of the project is done by Martena (2015).

3.3.3 Validation of the process models and ontologies

In the last phase, the ontologies and its related business process models have to be validated for completeness and correctness. Obviously, this belongs to the 'validation' phase of the regulative cycle of Van Strien (1997). During the making of the ontologies in the previous step, feedback will be provided on the process models when things are missing or incorrect. This can be seen as an extra validation of the process models after the first validation by interviewing internal stakeholders. After the feedback is processed and the process models are updated, again the validity can be tested as is explained by Karlsson (2009) in section 2.4. This will be done iteratively until all stakeholders are satisfied with the process models and ontologies. This part of the project is performed by all three students (Beukeboom, Martena & Van de Laar, 2015).

3.4 SPECIFICATION METHODOLOGY INDIVIDUAL PROJECT

In this section a step-by-step specification of this individual research is given. The biggest contribution and focus of this thesis will be on the first and last part of the overarching project: constructing and validating the process models.

3.4.1. Step 1: Get a general overview of the process

Firstly, it is important to get a general idea of the process to create a basic insight. This must be done, because it then becomes clear which processes have to be investigated and where the boundaries of the system are. To do this, already existing documents and presentations about the processes can be used. Some initial drafts in Bizagi can be made for further analysis later on. This general overview identifies which stakeholders and end-users are involved and some initial interviews may already be planned, due to potential limited time and tight schedules of stakeholders.

3.4.2. Step 2: Conduct stakeholder analysis

The next thing to do is to perform a stakeholder analysis. This will be done by conducting structured interviews with stakeholders. The questions that will be used are derived from the 'design problem' and 'diagnosis/analysis' phase of the regulative cycle as described in section 3.2. On top of posing these questions to the stakeholder, the initial drafts in Bizagi can be brought so that the stakeholder already gets familiar with the appearance of BPMN. Moreover, he/she can pinpoint important aspects that are wrong or missing in the general overview. From the outcomes of the interviews, the functional and non-functional requirements can be captured. These will be kept at hand during the development of the process models.

3.4.3 Step 3: Construct the process models

From the information gathered by step 1 and step 2, process models can be developed. At first it is important to construct a process as it should go, the so-called "happy flow". After the completion of the happy flow, exceptions of the process can be built in to make the model more accurate to the real-world situation. When it is done this way, one does not lose track of the overview of the process. This approach can be seen as a top-down approach, because processes can be described in more detail in a deeper level.

3.4.4 Step 4: Validate the process models

When the process models are built, it is time to validate these. For validation, the questions of the 'validation' phase of the regulative cycle formulated in section 3.2 can be used. Together with the same stakeholders who were interviewed in step 2 the process models will be evaluated to test its validity according to Karlsson's (2009) guidelines. Outcomes of the evaluation can be used to start the cycle again at step 1 and to adapt the process models until the stakeholders are satisfied.

4 RESULTS

In this chapter the results are described of the steps mentioned in section 3.4. At first a general overview of the process is given, followed by an extensive stakeholder analysis. Next, the process models are given and finally the validation of these models is described.

4.1 PROCESS OVERVIEW

At first an overview of the eMeasure process is made in order to set the boundaries. Figure 5 shows the general process model of the development of eMeasures using DCM's on a high aggregation level. It shows that an applicant of an eMeasure first checks by himself whether the requested information product is already available in a digital repository. This digital repository is something that has to be made available in the future, which will function as a platform to be freely accessible for clinical use or research. In the case of already available information products in the digital repository the process of developing eMeasures using DCM's is not initiated. However, to fill this digital repository this process does have to be gone through first. In order to create an information product, it is checked whether necessary eMeasures are available. An information product can consists of (i) (a set of) eMeasure(s), i.e. (health) indicator(s), (ii) a signal or (iii) a data set. In this project we will focus only on the creation of eMeasures. These eMeasures needs to be connected on DCM's. In order to do this, it is necessary that these DCM's are available. Whenever, the required information cannot be connected on existing DCM's, a request is sent to the DCM analyst to enable this information. The DCM analyst on its turn develops these DCM's whereby he maps DCM's on the existing data warehouse of the LTHN. The database developer provides the data in the data warehouse. Whenever the data is provided, DCM's are created and eMeasures are made, the information product can be completed and published. This piece of software, i.e. the information product, will be put in the digital repository, so the applicant can generate its own information.

In this process, two different views towards coding, transforming and decoding can be held. At first, from the applicant's point of view the coding part consist of the request for an information product and its whether or not necessary receiving of it (depending on if it was already available). The generation of the information product is the transformation; the input is transformed to the desired output. The eventual requested information product is the output, which must be decoded by the applicant himself for further use; for example doing research on the requested information/data.

From the project team's point of view, the coding part consists of the incoming request for an information product and the readily available making of DCM's and its mappings on the data. These activities are essential for the making of an information product, which is the transformation. Finally, the publishing of the information product is the decoding part in order to ensure that the output can be used outside the system.

After the development of the general process overview, it has been decided that Van de Laar will focus on the process related to the applicant and the eMeasure analyst and I will focus more on the development of DCM's and its mapping on the data by the DCM analyst.

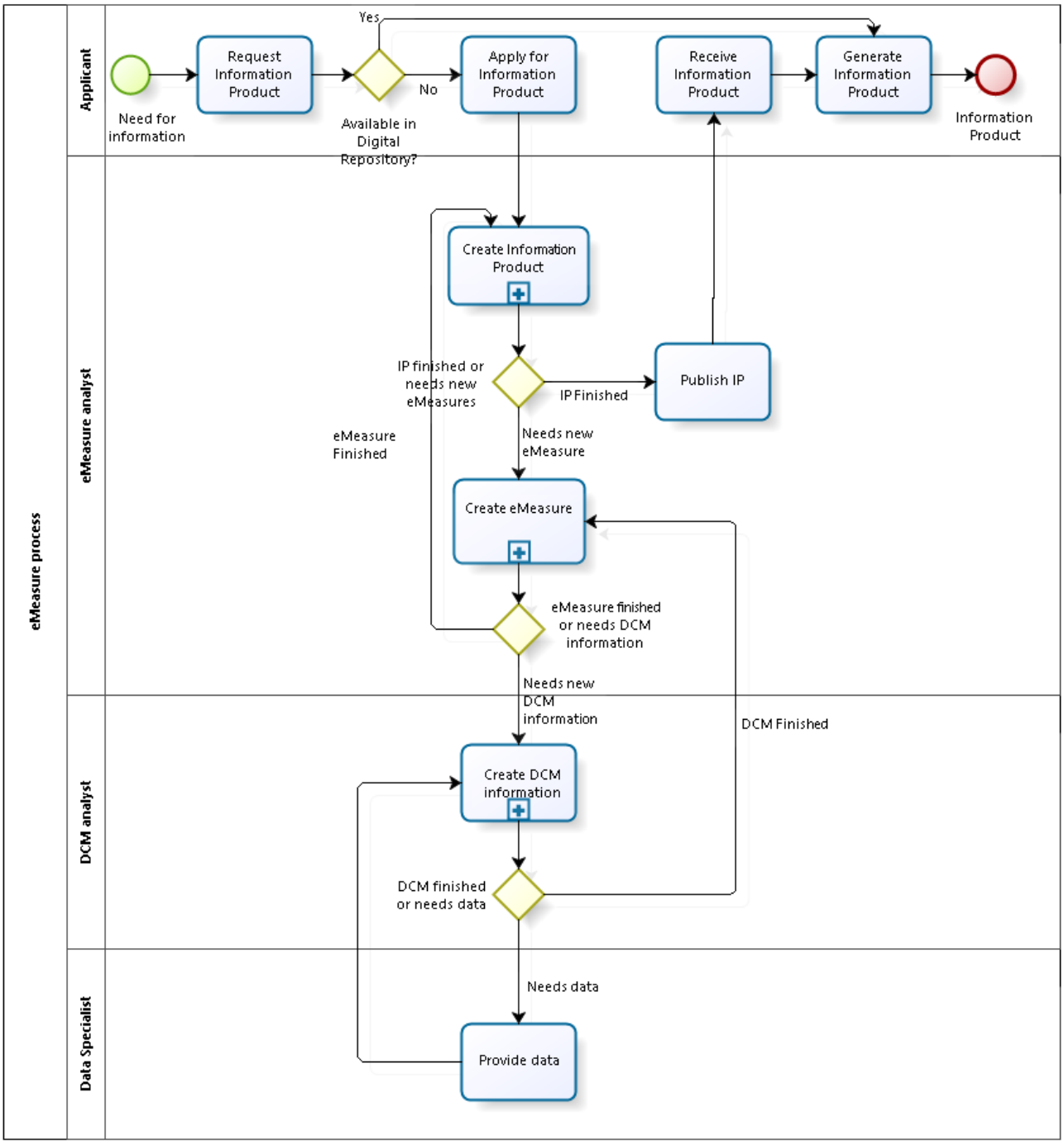


FIGURE 5: OVERVIEW EMEASURE PROCESS

4.2 STAKEHOLDER ANALYSIS

In this section all the stakeholders are identified and analyzed. Based on interviews, a list of stakeholders regarded to the process of developing eMeasures and its related DCM's is shown in table 1. A distinction is made between internal and external stakeholders regarding this process. What is meant by internal stakeholders is people who directly play role within the process of developing information products, eMeasures and DCM's. External stakeholders are people or instances who are affected by or have influence on this business process.

Stakeholders of the process of developing information products, eMeasures and DCM's

Internal	Applicant of an information product
	eMeasure analyst
	Team leader
	DCM analyst
	Quality assurer
	Data governance architect
	Domain expert
	Database developer
External	Researcher
	Management of the hospital
	External health care providers
	Patient
	Government
	HL7 NL, Nictiz

TABLE 1: STAKEHOLDERS EMEASURE PROCESS

The stakeholders are more elaborated on separately below with its relation to the process and their goals:

4.2.1 Internal stakeholders

Applicant

An applicant for a certain information product could be either from the hospital internally or externally. An example of a possible internal applicant is a doctor who wants to gain insight of his patients' survival rate after the use of a certain medicine. An example of a possible external applicant is IGZ (in Dutch: Inspectie voor de Gezondheidszorg), an authority who wants to receive every year a quality report relating to a wide variety of aspects within the hospital specified in eMeasures. An applicant comes up with an information product request for the hospital which initiates the entire to-be-performed process. Additionally, since the applicant is the one who wants to receive information about something, he/she also acts as a domain expert for questions relating to the request in case of any ambiguities. The goal of the applicant is to gain correct and clear information.

eMeasure analyst

The eMeasure analyst is the one who receives the requests of applicants which are not self-computable via the digital repository (which will be become available in the future) and has to find out what this request entails precisely. After a thorough analysis he/she connects the requested information to DCM's and makes the information product available in the repository. The eMeasure analyst is therefore the contact person between the applicant and the hospital. The analyst's goal is to provide a reliable and correct information product to the applicant that aligns with his request.

Team leader

The team leader of department A is responsible for 'delivering' the requested information on time in full. Therefore he/she needs to prioritize incoming information product requests. His goal is to achieve a service level as high as possible towards applicants.

DCM analyst

The DCM analyst is responsible for ensuring that the eMeasure analyst can connect requested information product (eMeasures) to DCM's. In case an information product cannot be connected to existing DCM's due to missing necessary attributes or values, he is responsible for adding these. Here, the DCM analyst can be regarded as the initiator of the ontologies by proposing attributes and values. In some cases, new DCM's need to be created when there is no relevant DCM yet available. Due to the fact that nowadays not a lot of DCM's are already available, this has to be done extensively in the near future, but will decrease in later stages. The goal of the DCM analyst is to provide widely-applicable and correct DCM's for both the connection of requested information products to it and other purposes (which will be discussed later).

Quality assurer
The quality assurer needs to assure that the proposed DCM of the DCM analyst is aligned with the standards of HL7 and Nictiz (a centre of expertise for standardisation and eHealth). His/her goal is to have standardized DCM's, which can be published on a public platform that is freely accessible for usage within a clinical context.

Data governance architect
The stake of the governance architect in this business process is to make sure that in the data creation as little as possible is free text. In other words, the possibility of a doctor to write in a box whatever he can. It is his goal to minimize this freedom and provide concise and clear value sets where doctors can easily choose from. This is done in order to structure data, though minimizing the loss of reality, so it can be made computable and will be of more use by researchers. Additionally, he/she determines from the proposal of the DCM analyst, which values are ultimately chosen and where these can be mapped on. Here, the governance architect can be regarded as the keeper of the ontologies.

Domain expert
Domain experts can vary broadly in what they are expert in. The main experts meant here are doctors, who can be regarded as health care domain experts within their specialization. They are consulted in the process of editing existing or making new DCM's for clinical information. They are advised for determining value sets within the DCM's and needed for filling in the clinical content within a DCM. Moreover, they perform a final check to validate the DCM before it is published. Their goal is to assist the DCM analyst in the correctness of DCM's for the clinical part. Doctors, can also play role as an internal applicant as is mentioned before. Another type of domain expert can for example be someone who is expert of a specific needed part of a data warehouse. All these types of domain expert have as goal to assist in the DCM process where needed and to ensure correctness.

Database developer
A DCM needs to be mapped on an existing database. Although this mapping is done by the DCM analyst, it will need validation of someone else whether this is done correctly. This can be done by the person who is responsible for the data registration at the source, i.e. the database developer. His goal is to provide all the data necessary and validate the mapping of DCM to the database. We assume in this research that all requested information is available in the existing databases of the LTHN.

4.2.2. External Stakeholders

External stakeholders are not taken up into the process models, but it is important to mention their stake of the process to show its importance. Their relevance will be explained together with their goals.

Researcher
A researcher wants to perform statistical clinical research, on which improvements can be made for health care. He/she has a great interest in structured registration, whereby value sets are of much more use than free text. Moreover, he/she wants to reuse the clinical data. DCM's are for this matter extremely useful, whereby text boxes are only used when there is no alternative. Additionally, eMeasures and DCM's are published in a catalog and open for consultation for the researcher. These are then able to be reused multiple times. On top of that, due to addition of clinical content of DCM's by domain experts, these can be easily and unambiguously interpreted by the researcher.

Management
The management of a hospital has as goal to create insights in company records and use that for reporting purposes. eMeasures and DCM's are very useful in reporting, due to its generic and standardized characteristics. Additionally, the process models describing the eMeasure and DCM process comes in very useful for explanatory purposes towards the employees of the organization to persuade them to cooperate with this new method.

External health care providers
With the use of DCM's, patient information provided by the LTHN can easily be processed in the registration of external health care providers, like other hospitals or general practitioners (GP's). This is due to the digitalized way of exchanging patient data and its standardized form. Moreover, the research between other LTHN is stimulated by the use of generic DCM's.

Patient
In health care the most important thing is of course the patient that needs good care. A patient wants the guarantee that the health care providers make the best decisions on behalf of correct and sufficient information. The development of eMeasures and DCM's will increase the degree of transparency of health care and stimulate better research. This will in its turn lead to better health care for patients.

Government
The government has as goal to gain information about the health care sector to have as basis for policy making. eMeasures will therefore give clear insights about the quality of care within a hospital. Due to the standardization of eMeasures and DCM's, one

is sure that the quality between different health care providers can be righteous compared. Moreover, the regulations set by the government may not be violated by the LTHN.

HL7 NL, Nictiz
HL7 NL and Nictiz are initiatives that want to create unity of language in the health care sector and have as goal to make easily and clear exchange of information possible for health care and research. They created standards to enable this, where eMeasures and DCM's needs to be consistent with.

4.2.3. Critical success factors

In this section the Critical Success Factors (CSF's) of the business processes and products are defined based on the goals of the stakeholders, described in the previous section. These CSF's can be considered as the basis for the business requirements which will be used for the development of the process models.

Related to the eMeasure part, where Van der Laar's focus will be on:

- The requested information product must have a logical structure, i.e. the same format for every request, e.g. standard form
- The requested information product must be clear, i.e. unambiguous
- The requested information product must contain all needed information
- The development process of new information products and eMeasures must be transparent
- The eMeasure analyst must have a tool that can develop eMeasures in HQMF
- For developing eMeasures, a feasibility test must be conducted whether the data is available
- For developing eMeasures, the needed DCM's must be available
- For developing eMeasures, the needed DCM's must be correct
- For reviewing eMeasures a predetermined test must be available
- Information products and eMeasures must be able to be reused

Related to the DCM part, where this thesis' focus will be on:

- DCM's must be as generic as possible
- DCM's must not be redundant
- DCM's must be able to be reused
- The development process of DCM's must be transparent
- DCM's must be described unambiguously
- DCM's must be coded according to an accepted code scheme
- DCM's must be defined in such a way that they are useful in a clinical context
- DCM's must indicate in which context they can be used
- DCM's must be published in a catalog on a public platform for clinical usage
- DCM's must be mapped onto a data warehouse in order to generate data
- DCM's must be validated and satisfy the ISO-standard
- DCM's must be intuitively understandable without thorough knowledge of ICT-systems or terminology
- DCM's must be able to be applied for reporting purposes

4.2.4 Business requirements

From the CSF's that were formulated from all stakeholders' goals, a translation is made to business requirements. In other words, the CSF's are described in a way that they can be used for the construction of the process models. Only the business requirements relevant for this thesis is further explained, i.e. DCM-focused.
As described in section 2.1, a distinction is made between functional requirements (FR) and non-functional requirements (NFR). The FR are expressed in the form "the system shall <do requirement>", whereas the NFR are expressed in the form "the system shall be <requirement>".

- The eMeasure process shall complete a correct and validated eMeasure
- The DCM process shall complete a correct and validated DCM
- A DCM request shall be clearly explained until fully understood by the DCM analyst
- The DCM process shall first come up with a candidate DCM so the eMeasure analyst can already start with connecting the eMeasures to the DCM to save time
- The DCM process shall then come up with a final DCM and a mapped DCM
- A DCM shall be reviewed on its structure by a quality assurer
- DCM value sets shall be asked for consultation to domain experts

- A DCM shall be reviewed on its clinical correctness by domain experts
- A DCM mapping shall be evaluated by a database developer
- DCM's shall be formatted in such a way that free text spaces will only be available when absolutely necessary
- The database shall log information; the responsible person and time for each activity in the eMeasure and DCM process, to support audits.

4.3 DEVELOPING THE PROCESS MODELS

Based on the previously described process overview and extensive stakeholder analysis, where the goals of each stakeholder were identified together with the CSF's and business requirements, the process models can be developed. The internal stakeholders are taken up into the process as the different swim lanes, whereby the goal of each stakeholder needs to be met at the end of its swim lane. Furthermore, the process models needs to conform all CSF's and business requirements.

As mentioned before, the focus of this thesis will be on the process of developing DCM's with regard to eMeasures. This means that whenever an eMeasure needs DCM information that is not yet available, the process of developing DCM's will be relevant. A first design of the DCM process regarding eMeasures is made based on some limited existing documentation and an interview with the DCM analyst who has a good view of how the process should look like and is key actor within this process. During the construction of the process models Systems Thinking was used, whereby a top-down approach is adopted. This means activities may be zoomed in to provide more details. Within all process flows, the three phases of System Thinking 'coding of input', 'transforming input to output' and 'decoding of output' can be identified.

The decoupling point between the part of Van de Laar (2015) and this thesis is shown in figure 6 in a slightly simplified way to create clear insights. It can be seen that the development of DCM's exists of three major activities. First, when a request from the eMeasure analyst comes in due to insufficient information in existing DCM's, a candidate DCM is made. From here, the process splits up into three parallel activities. One is that the eMeasure analyst can already connect the eMeasure to the candidate DCM so time savings can be realized. The other two activities, 'Make final DCM' and 'Map DCM', are executed by (or at least falls within the responsibility area of) the DCM analyst. A final DCM means that clinical context is added and the DCM is reviewed by a domain expert. When this is done, the eMeasure analyst can verify whether its previous connection to the eMeasure is still correct. A final connection of an eMeasure to a final DCM may be completed, but in order to actually derive data from it, the mapping of the DCM to the data warehouse needs to be completed first.

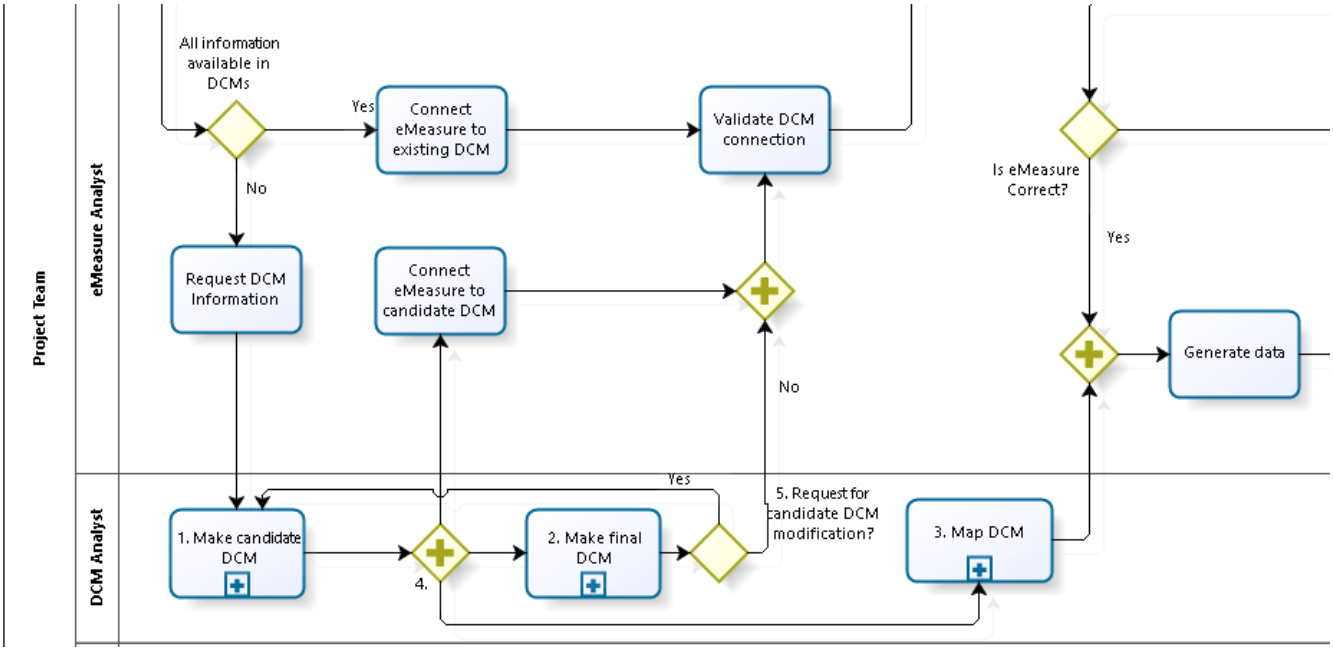


FIGURE 6: HIGH AGGREGATE LEVEL OF DCM PROCESS RELATED TO EMEASURE PROCESS (SIMPLIFIED)

In the initial stages of constructing the process models it became clear that the processes of developing eMeasures and its related DCM's are complex. Therefore, in order not to create big spaghetti diagrams and to keep the processes digestible, nested activities are essential. Important notice is that a nested activity can still be performed by multiple stakeholders, but the swim lane in which the nested activity is placed indicates who is the main executer/responsible person for this activity.

The three nested activities performed by the DCM analyst depicted in figure 6 are further modeled in lower aggregate level(s) and are given in appendix IV.i. This appendix contains the first designed process models. These were constructed by interviewing the prime actor within this process, the DCM analyst.

Now the basics of the process are defined, these preliminary results could be used as input for the next step in the overarching project. In other words, Martena (2015) uses the process models from appendix IV.i to start building ORM models. An advantage of starting this next step of the overarching project in an early stage is that early feedback can be given. This feedback will be used as a starting point for further improvements of the process models.

According to the feedback of Martena (2015) the process models are adjusted and are shown in appendix IV.ii. These improved process models were used during interviews with different stakeholders (2 DCM analysts, eMeasure analyst, data governance architect and a doctor) for validation. More explanation of the validation is given in next section. The outcomes of this validation were used to either ensure correctness or to find out whether some modifications needed to be made. Subsequently, adjustments were made to the process models and again were validated by the stakeholders. This process went on until all stakeholders were satisfied. This ultimately has led to the final process models which are shown in appendix IV.iii.

Explanation per activity and gateway is given in appendix III: the BPMN appendix. To clarify how the process models act during a real-case request, an example is used that runs through the process and is described into detail in appendix V.

4.4 VALIDATING THE PROCESS MODELS

As is already mentioned shortly before, the initial process models of the previous step needs to be validated in order to ensure correct models. To do this, Karlsson's guidelines described in section 2.5 and the questions mentioned under the validation phase in section 3.2 were used. In addition, during the construction of the ORM-models by Martena (2015), issues came up related to the process models that needed to be solved.

At first, feedback received from Martena (2015) is used to improve the process models so that he could make ORM-models in a proper way. Examples are that notation is needed at every start and stop event and the insight that an applicant could be better taken as a separate swim lane instead of a separate pool. It was done this way, because difficulties arose due to the need for multiple start events (within the eMeasure process), which is not allowed according to the BPMN rules. Moreover, a separate pool for a stakeholder indicates that it has its own database. This occurs often in for example Business-to-Business processes. Since in our process this is not the necessarily the case, one pool containing all the stakeholders satisfies and ensures that the process model does not become unnecessarily complex. Another example was that gateway '1.2 all information self-explanatory?' was too general for Martena (2015) being able to develop proper data models. Due to the lack of possibilities to deepen on gateways and not to create unnecessary big process models, an attachment including a checklist is given within the process model. This is being visualized within this thesis by implementing the checklist in the BPMN appendix so clear insights are created for Martena (2015).

When looking at Karlsson's tests, construct validity has been addressed by interviewing multiple stakeholders that participate within the DCM process, namely the eMeasure analyst, two DCM analysts, a data governance architect and a doctor. It is chosen to use them for validation, since they act as the most important stakeholders. By showing and explaining them the process models, feedback was received whether they could identify themselves with their tasks and sequence of events. From these interviews the process is taken into consideration from multiple angles and therefore tested for construct validity. Moreover, multiple interviewers (Van der Laar & Beukeboom, 2015) were used to create the overview of the process in order to be sure that no individual misinterpretation led to a defect in an early stage.

Internal validity has been addressed by checking whether discrepancies exist between the outcomes of interviews with different stakeholders. During meetings where multiple stakeholders attended, conflicts and differing opinions were discussed. An example that came up was the difference in opinion when a DCM needs to be mapped on the data warehouse within the process. After consolidation, a decision was made where the concerned stakeholders were both satisfied with and therefore was internally validated.

External validity checks whether the solution can be made generalizable. Since the constructed process models are made to view the generic process of developing DCM's related to eMeasures this is definitely the case. All LTHN specific elements are purposely kept out of the process models. Therefore these process models can easily be adopted by other hospitals.

Reliability of the research is guaranteed by the use of a predetermined methodology explained in section 3. In here, the questions that were used during every interview are stated and can therefore be repeated if desirable. Unfortunately due to the lack of a tape recorder, the interviews itself were not recorded and can therefore not be retrieved from a database.

The questions listed in section 3.2 under 'validation phase' were brought up during the multiple interviews. It must however be noted that not all of these questions were answered, due to the fact that the designed process is not implemented yet. However, by going through the process 'on paper' it is seen that all CSF's are met. Only the business requirement saying the process shall do data log information is not visualized in the process models, but will be facilitated by the ORM-models which put a time stamp on the activities. Additionally, as is already discussed before, the solution, i.e. the process models, is very well able to upscale towards other hospitals.

5 DISCUSSION

In this section the performed research will be discussed, both the content as on how the research was conducted.

5.1 BPMN CONSIDERATIONS

In the early stages of the project while developing the process models, some shortcomings of BPMN came across. Firstly, BPMN is not able to deal with a main activity that zooms in a sub process that consist of multiple swim lanes, e.g. one stakeholder is the main task owner, but needs for that specific activity also others to complete this action. To keep the process diagram clear and not get a giant spaghetti diagram this is necessary. Therefore, to solve this problem, a new diagram in Bizagi has been created that is named the same in the higher level diagram without the explicit link between them. Therefore, one is able to create new pools and lanes in the zoomed-in activity.

Secondly, BPMN is not able to view one activity that is done by two persons (i.e. different swim lanes) at the same time in a proper manner, e.g. reviewing a candidate DCM which is done in a conversation. This can only be done by message flows when they differ from pools, but not swim lanes. Therefore, to solve this problem, Van der Laar and I decided to make a parallel gateway and name the processes the same for both stakeholders to show that this is done at the same time and the next activity cannot be preceded before these are both done, this is shown again with a parallel gateway. The lane in which the parallel gateway is located shows who is the primary action taker of/responsible for that activity. An example of this is given in the first process models. It must be noted that in later stages of the project this has been handled differently, since it turned out that it did not actually happen in a conversation sort of way. This does not mean, though, that this is not a BPMN limitation.

5.2 VALIDATION OF THE PROCESS MODELS

The validation of the business process models has been mainly done by checking whether all CSF's, business requirements and goals of each stakeholder have been met. Moreover the process models have been presented to different stakeholders to hear their view of a process that is not yet put in practice. Naturally, this is something that has to be done first before actually initiating the process in practice. However, there is high risk of missing issues which will only be uncovered when the process is effectively in use. Unfortunately, due to limited time resources, the implementation phase could not be witnessed and therefore a more extensive validation was not being able to be performed.

A remarkable thing, though, that came up during the validation when interviewing a second DCM analyst was why the term DCM is actually used, and not DM. To explain this, one has to understand that eMeasures cover quality indicators in all sorts of areas. This does not only mean clinical indicators, but also financial, operational etc. Since eMeasures are connected on DCM's and DCM's are actually only a way of structuring information, this does not necessarily have to limit to clinical data. Financial and operational data is also able to be put in a similar way as clinical data. Therefore, it would be a legitimate argument to change DCM to for example DIM; Detailed Information Model, whereby the specification of 'Clinical' can be left out and made more general to 'Information'.

5.3 ALTERNATIVE TO DEPICT DCM-RELATED BUSINESS PROCESSES

During this project, the process of developing DCM's was the center of attention. First an overview of the process has been made, followed by a stakeholder analysis, which eventually led to the process models. All these activities were basically focused on the entity DCM. An interesting alternative method to develop business process models that makes use of this characteristic is described by Bhattacharya, Hull & Su (2009).

Bhattacharya, Hull & Su (2009) developed 'a design methodology for business processes and workflows that focuses first on 'business artifacts', which represent key business entities, including both the business-relevant data about them and their macro-level life cycles.' Within this research project, DCM's would suit well for that business artifact. Then, individual workflow services, i.e. tasks, are incorporated, by giving a specification on how they operate on the artifacts and fit into their life cycles. The resulting workflow is then specified in a particular artifact-centric workflow model. At the logical level such workflow model is largely declarative, in contrast with most traditional workflow models which are procedural and/or graph-based (Bhattacharya, Hull & Su, 2009). This design methodology sounds like a promising alternative to show the business processes around developing DCM's regarding eMeasures. It is therefore something to take into consideration when conducting a similar project.

6 CONCLUSION

In this chapter the conclusions regarding the research objective and questions will be presented. In addition, the limitations of the project and recommendations for further research are given.

6.1 RESEARCH CONCLUSIONS

This research was conducted to answer the following research question:

How can validated process models be derived from the business processes related to the development of DCM's with regard to eMeasures?

In order to answer that question, first several sub-questions had to be answered which are listed in section 1.1. The sub-questions 'How to build BPMN models using Systems Thinking?' and 'How to validate process models of business processes' have been answered in the theoretical background section. The other two sub-questions are extensively discussed in the results section. To answer the main research question, a short summary is given what is done:

At first, existing literature was used to gain an understanding of business process modeling and validation in general. Subsequently, the regulative cycle of Van Strien (1997) was used as methodology and kept as handhold for questions related to each phase. These questions were used during the interviews to perform an extensive stakeholder analysis. From this analysis, the requirements for the process models were drawn up and used as basis for the development of the first business process models. After receiving feedback from Martena (2015) and validation interviews with different stakeholders, the process models have been adopted accordingly until all interviewees were satisfied.

From the results, one can state that the objective of this research 'design a validated process for developing DCM's with regard to eMeasures using BPMN and Systems Thinking' has been successfully met. This is underlined by the fact that the LTHN, where the research was conducted, actually is using the constructed process models as fundament to incorporate in their project management tool.

6.2 LIMITATIONS OF THE PROJECT

One of the limitations of the project is, as is mentioned before in the discussion section, that it was unfortunately not possible to be present during the actual implementation phase; in other words, to witness when the LTHN actually puts the business process models into practice. From here, the business process models could have been validated more extensively and on its turn improved after feedback from the end-users.

Another point to mention is the missing view of the project leader who in fact initiated this entire project. The contribution of the project leader could have led to new different viewpoints that may have been missed now. After all, he had the best overview from the beginning. Unfortunately, he was not able to attend during the rest of the project due to personal circumstances.

6.3 RECOMMENDATIONS FOR FURTHER RESEARCH

A recommendation for further research, already brought to light in the discussion section, is to consider the possibility of designing business processes according to the design methodology of Bhattacharya, Hull & Su (2009). A key challenge in business process management is to enable business managers to understand, design and easily make changes to their business operations, so that they are confident that their goals are accurately reflected in the underlying IT workflows (Bhattacharya, Hull & Su, 2009). This challenge is addressed by the paper of Bhattacharya, Hull & Su, where they present a promising modeling framework and methodology, which is fundamentally centered around data rather than activity flows. More specifically, this data, captured in a 'business artifact', would in this project be the DCM's. Since the artifact-centric approach has already been successfully applied in business process design, it is expected that it may also work well for designing the processes around the development of DCM's regarding eMeasures.

A second recommendation for further research arises from the limitations, namely validating the process models after the process is put in practice. Using feedback from the end-users could further optimize the business process models. Another point that came to discussion during the project is that the necessary data for a detailed view of the process could be validated by showing user-interface mockups to the different stakeholders. Especially, for doctors, who are no technical domain experts, this could be a useful tool for validation. However, due to limited time resources it was not possible to perform this validation during this project and is therefore something to be executed for later studies.

7 REFERENCES

Aalst, W. M. P. (2012) "What makes a good process model?", *Software & Systems Modeling*, Vol. 11(4), pp 557–569.

Aguilar-Saven, R.S. (2004) "Business process modelling: Reviewand framework", *Int. J. Production Economics 90*, p.129–149

Beukeboom, R.T. (2015) "Design of the process of developing DCM's with regard to eMeasures", *University of Groningen*

Balsters, H. (2013a) "Mapping BPMN process models to ORM data models"; In: *Lecture Notes in Computer Science*, nr. LNCS 8186 pp 503-512.

Balsters, H. (2013b) "FB-BPM"; *Internal Report RUG*

Balsters, H. (2014a) "A System for extracting Data Models from basic Business Process Models", In: *Lecture Notes*, 1-36

Balsters, H. (2014b) Lecture Slides of course: "Design Methods" *University of Groningen*

Bhattacharya K., Hull R., Su J. (2009) "Data-Centric Design Methodology for Business Processes", *Handbook of Research on Business Process Modeling*, chapter 23, pp. 503-531

Boll, M., (2006) "Kritische succesfactoren bij de implementatie van een elektronisch patiëntendossier", *Doctoraal Informatiekunde*, p 1-47

Chinosi, M., & Trombetta, A. (2012) "BPMN: An introduction to the standard", *Computer Standards & Interfaces*, Vol. 34(1), pp 124–134

Curland, M., Halpin, T. (2007) "Model Driven Development with NORMA", *Proc. 40th Int. Conf. on System Sciences (HICSS-40).* IEEE Computer Society

Goossen W., Goossen-Baremans, A., Van der Zel, M. (2010) "Detailed Clinical Models: A Review", *Health Informatics Research (HIR),* 16(4), p.201-214

Gruber. T.R., (1993) "A Translation Approach to Portable Ontology Specifications", *Knowledge Acquisition*, 5(2) pp. 199-220

Halpin, T. & Bloesch, A. (1999) "Data modeling in UML and ORM: a comparison", *Journal of Database Management,* Vol. 10(4), pp 4-13.

Halplin T. & Morgan, T. (2008) *Information Modeling and Relational Databases*, 2nd ed., Morgan Kaufmann Publishers

Hooda J.S., Dogdu E., Sunderraman, R. (2004) "Health Level-7 compliant clinical patient records system", *Proceedings of the 2004 ACM symposium on Applied computing*, March 14-17

In 't Veld, J., Slatius, B., & In 't Veld, M. (2007) "Analyse van bedrijfsprocessen", *9th edition* Groningen: Wolters-Noordhoff.

Karlsson, C. (2009) "Researching Operations Management", Routledge: New York
Li, J., Eberlein A., Far B.H. (2004) "Evaluating the Requirements Engineering Process using Major Concerns", *Software Engineering*, vol. 418, pp. 237-252

Li. L, Yang Y., Wu B. (2005) "Agent-Based Ontology Mapping Towards Ontology Interoperability", *Lecture Notes in Computer Science*, Volume 3809, pp 843-846

Martena, P. (2015), "Object-role modeling: validation of a database design methodology in the context of an HER system", *University of Groningen*

Selvakumar, J., & Rajaram, M. (2011) "Performance Evaluation of Requirements Engineering Methodology for Automated Detection of Non Functional Requirements", *International Journal on Computer Science and Engineering*, Vol. *3*(8), pp 2991–2996.

Stichting DCM, (2014) available at http://www.detailedclinicalmodels.nl/ [Accessed Oct. 6th 2014]

Van de Laar, P. (2015) "Creating a general method for eMeasure development", *University of Groningen*

Van Strien, P. J. (1997) "Towards a Methodology of Psychological Practice: The Regulative Cycle", *Theory & Psychology*, Vol. 7(5), pp 683–700.

White S.A. (2004) "Introduction to BPMN", *BPTrends*, p. 1-11

Wieringa, R. J., & Heerkens, J.M.G., (2006) "The methodological soundness of requirements engineering papers: a conceptual framework and two case studies", *Requirements Engineering*, 11(4), 295-307

APPENDIX I: THEME POSTER ON DESIGNING EMEASURES AND BUILDING BLOCKS FOR AN EHR

Objective: The goal of this project is to design a quality control system for patient treatments at a Large Teaching Hospital Netherlands

Designing E-Measures and Building Blocks for an EHR

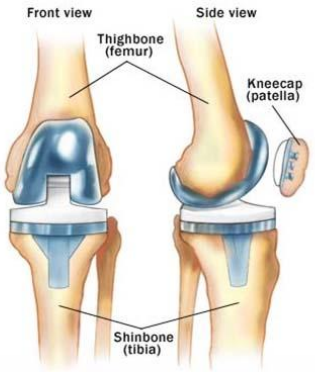
Introduction

- To reduce healthcare expenses, the government and Dutch national hospitals signed a covenant to bundle healthcare systems by building an Electronic Health Record System (EHR)
- Hospitals are building an EHR
- Controlling quality of patient treatments is one of the goals of the EHR
- So-called "Health Indicators" (HI) offer a protocol for quality control after a patient treatment has taken place
- HI-Reports are stored in a Datawarehouse
- Goals are to eventually create an Auditing System and a Research Database to obtain medical

As-is situation

- Word documents describing a HI-protocol
- Building Blocks (DCM= Detailed Clinical Models): reusable and often re-ocurring parts of patient treatments, along with their own (local) HI-protocols
- Examples of using a HI-protocol using existing building blocks

Example: Indicator "Knee Replacement"



Hospital Information System:

E-Measure (Quality control proces)**:**
- Antibiotic profylaxe applied?
- Routine control applied?
- Registration of all relevant data applied?
- Etc.

Building Blocks (reusable parts of the system)**:**
- Patient transfer
- Bloodpressure
- Etc.

To-be situation

- Design of a validated process model (**query flow**) for HI-protocols
- Such a process model is called an E-Measure
- Design of more validated Building Blocks

Research Design

Major questions:
- Who are the major stakeholders in an E-Measure system?
- What are their goals and critical success factors?
- What is the state of the art concerning E-Measures and Building Blocks?
- What are the requirements of a reliable and technically feasible design?
- How could we validate the correctness of our design?

Advisor

Dr. Herman Balsters
Associate Professor of Information Systems Design
h.balsters@rug.nl
+31-50-3633923

Student projects 2014-2015

Method: Systems approach

- E-Measurement as a System of Query flow, with input-, transformation-, and output functions
- Black-box approach in systematically building a more detailed system

BPMN

- The system to be designed can be modelled using BPMN
- BPMN is the international de facto standard for process modeling

From BPMN to Data Models

- By systematically extracting required data elements from activities inside the BPMN process models, we can gather tailor-made data for the E-Measure process
- After this extraction phase is completed, we have a complete set of data for the whole E-Measure process: the data model

From BPMN and Data Model to UI mock-ups: validating the end-user

- BPMN models and associated data models offer the ingredients for building UI mock-ups
- These UI mock-ups can be given to the end users to check (validate) that their requiremens have been fully met

From Practice to Science

- Abstract from examples of E-Measures to find the general principles of design of such systems
- Abstract from examples of validation of E-Measures to a general approach to validation of E-Measures

==========

4 students:

- Gather data pertaining to specific E-Measures and Building Blocks, literature research
- Construct process- and data models (query flows) for specific E-Measures
- Validate E-Measures using UI mock-ups
- Translate the practical results into Science: general models and methods

APPENDIX II: DCM EXAMPLE

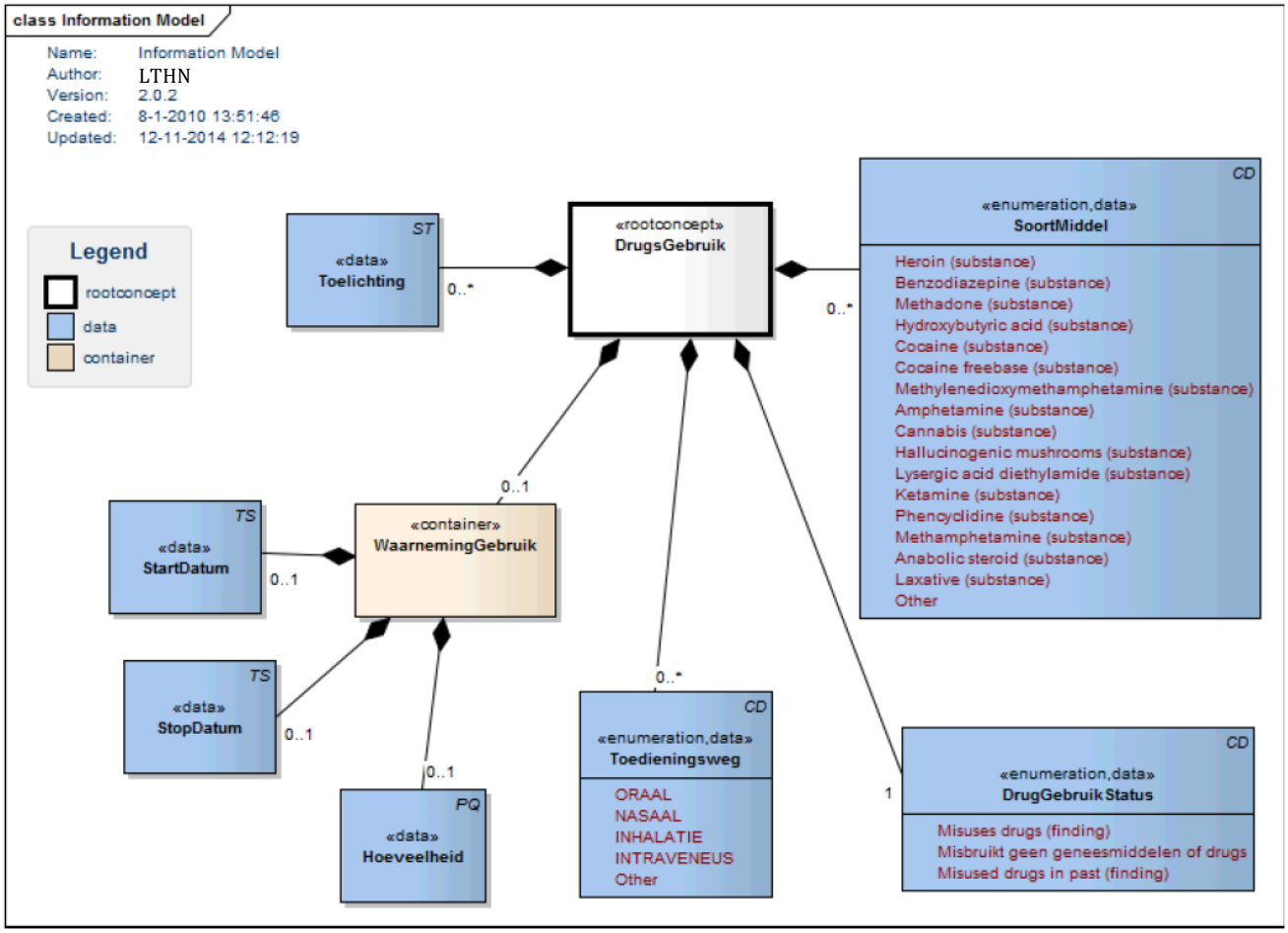


FIGURE 7: AN EXAMPLE OF THE CHAPTER INFORMATION MODEL OF A DCM ABOUT DRUG USE

APPENDIX III: BPMN APPENDIX

In this appendix all activities and gateways from the process models are briefly explained. The numbering of the processes indicates the aggregate level of the process. For example, number 1.8.3 refers to: the first number to the highest aggregate level (Make Candidate DCM), the second number to one level deeper (Analyze DCM request) and the third number to another level deeper (Estimate work load building new DCM).

1. Make Candidate DCM

This nested activity shows that a candidate DCM is made. A candidate DCM is a DCM where the DCM analyst does propositions for and can be used for a preliminary eMeasure connection.

Input: Request for DCM information

Output: Candidate DCM

1.1. Request DCM information

This start event indicates the need for DCM information since it is not available for the eMeasure analyst yet.

1.2. All information self-explanatory?

In this gateway the DCM analyst walks through a checklist to see whether all needed information is given and understood. The checklist consists of the following bullet points:

- Did the eMeasure analyst propose a possible DCM?
- Did the eMeasure analyst come up with a proposal for a grouping of attributes or with (a list of) (different) attribute(s)?
- Did the eMeasure analyst come up with the data type per attribute?
- Did the eMeasure analyst come up with requested value sets?
- Is all the needed clinical terminology clear?

Output: if all questions answered positively: Yes, otherwise: No

1.3. Request information

If the DCM analyst did not fully understand the incoming request, he/she makes a request for more information/explanation to the eMeasure analyst. What kind of information depends on the questions that were answered 'No' to in the previous gateway.

1.4. All information available?

In this gateway the eMeasure analyst checks whether he/she has all information available to answer the questions of the DCM analyst regarding the checklist.

Output: Yes/No

1.5. Request information

If in step 1.4 the output was 'No', the eMeasure analyst requests more information to the applicant in case of any unclarities. Since the applicant is most likely not familiar with DCM's, this will be regarding unclear clinical terminology. The eMeasure analyst functions here as the link between the DCM analyst and the applicant.

1.6. Provide information

The applicant provides information about any unclarities what he/she really wants. Especially, very specific clinical terminology is what mostly needs some more attention.

Input: Request for information

Output: Information/Explanation

1.7. Provide information

The eMeasure analyst provides information to the DCM analyst for unclarities regarding the request for DCM information which came up after the DCM's checklist.

Input: Provided information from the applicant, 'Yes' after the check if all information was available

Output: Information/Explanation regarding the request for DCM information.

1.8. Analyze DCM request

In this nested activity the DCM request is analyzed on what needs to be done and an estimation is given of the expected work load.

Input: A clear DCM information request

Output: An expected work load indication

1.8.1. DCM request

This start event indicates a request for DCM information in which is clear to the DCM analyst what needs to be done.

1.8.2. Fits within existing DCM?

This gateway checks whether the requested DCM information fits within an existing DCM.

Output: Yes/No

1.8.3. Estimate work load building new DCM

When in gateway 1.8.2, the output is 'No'; an estimate will be made for the work load of building a new DCM. This will in general be most time-consuming compared to the other options regarding DCM's.

1.8.4. Estimation work load

The outcome of the previous step results in the end-event of this sub-process, namely an estimation of the work load of building a new DCM.

1.8.5. Needs new attribute?

This gateway checks whether an new attribute needs to be added within an existing DCM. For explanatory reasons, the hierarchy is given by: a DCM consist of numerous attributes, which on its turn may exist out of numerous values.

Output: Yes/No

1.8.6. Determine what attribute in DCM needs to be added

In case the output of 1.8.5 is 'Yes', it is determined what attribute needs to be added to the DCM.

1.8.7. Needs new value?

This gateway checks whether a new value needs to be added to an attribute. Not all attributes needs values.

1.8.8. Determine what value needs to be added to which attribute

In case the output of 1.8.5 is 'No' and/or the output of gateway 1.8.7 is 'Yes', it is determined what value needs to be added to which attribute of the DCM.

1.8.9. Estimate work load

Subsequently an estimate is made about the work load of adding an attribute and/or adding a new value(s).

1.8.10. Estimation work load

The outcome of the previous step results in the end-event of this sub-process, namely an estimation of the work load.

1.9. Prioritize requests

The team leader prioritizes requests for DCM information based on their estimation of work load and importance. Its importance is already labeled by the eMeasure analyst in an earlier stage and is also taken into account by the team leader when prioritizing.

Input: Estimated work loads of all incoming requests, label of importance by eMeasure analyst

Output: A prioritized list of DCM requests

1.10. Appoint request

The team leader appoints the request to the DCM analyst based on its prioritized list.

Input: Prioritized list

Output: Appointed DCM requests

1.11. Propose Candidate DCM

This nested activity proposes a candidate DCM both for its structure as for values and their related possible mappings.

1.11.1. Initiation proposal candidate DCM

This start event indicates the initiation of the making of a proposal candidate DCM.

1.11.2. A new DCM needs to be created?

This gateway checks whether a whole new DCM needs to be created or not.

Output: Yes/No

1.11.3. Consult Existing libraries

In the case of a whole new creation of a DCM, existing libraries are consulted for generating ideas. Examples are DCM's from other LTHN-projects (PSI, GenOGeg, eOverdracht) or clinical models at CEM, CDE and OpenEHR.

1.11.4. Appoint root concept

In this activity the root concept is determined for the new to-be-developed DCM in the chapter Information Model. An example is 'Drugs Usage'.

1.11.5. Appoint attribute(s)

In this activity the attribute(s) is/are determined for the concerning DCM in the chapter Information Model. An example within 'Drugs Usage' is 'Sort of Drugs'.

1.11.6. Determine data type per attribute

In this activity the data type per attribute is determined for the concerning DCM in the chapter Information Model. The data type for 'Sort of Drugs' is CD.

1.11.7. Determine cardinality

In this activity the cardinality per attribute is determined in the chapter Information Model of the DCM. The cardinality of 'Sort of Drugs' is 0…*.

1.11.8. Needs new attribute?

This gateway checks whether a whole new attribute needs to be added to an existing DCM or only a new value to an existing attribute.

Output: Yes/No

1.11.9. Consult existing libraries

In the case of adding a new attribute within an existing DCM, existing libraries are consulted for generating ideas. Examples are DCM's from other LTHN-projects (PSI, GenOGeg, eOverdracht) or clinical models at CEM, CDE and OpenEHR.

1.11.10. Parallel gateway

This gateway shows that from this point multiple activities take place simultaneously, i.e. in parallel.

1.11.11. Propose value set

In this activity a proposition is made by the DCM analyst for the value sets that are options to choose from for doctors. A few examples of values within the attribute 'Sort of Drugs' are Heroin (substance), Benzodiazepine (substance) and Methadone (substance).

1.11.12. Propose possibilities mapping on data warehouse

Together with the propositions for value sets, possible mappings for these values on the data warehouse are proposed. This is done this way, so no value sets are proposed that are not available in the data warehouse.

1.11.13. Parallel gateway

This parallel gateway shows that first the previously two activities needs to be completed before the process can be continued.

1.11.14. Proposed candidate DCM

This end event indicates the end of the sub-process Propose candidate DCM with its output a proposed candidate DCM.

1.12. Review proposed Candidate DCM

The proposed candidate DCM needs to be reviewed by the architect, who is specialized in designing and defining DCM's according the official standards set. The architect focuses on the structure of the DCM (e.g. root concept, attributes, data type and cardinality) and not on the clinical content or mapping.

1.13. Is proposed Candidate DCM correct?

This gateway checks whether the Candidate DCM's structure is correct according to the architect. Either the candidate DCM needs to be adjusted based on the feedback of the architect or the process can be continued.

Output: Yes/No

1.14. Send Proposed value sets and possibilities mappings for review

When the DCM's structure is approved, the proposed value sets and possibilities for mapping are sent towards the governance architect to harmonize these and he makes the final call.

1.15. Harmonize Value sets

This nested activity determines which value sets, which codes and mappings are to be used in the DCM. This is done by the data governance architect.

1.15.1. Proposal Value Set and possibilities mappings

This is the start event where the data governance architect receives the proposed value sets and possibilities for mappings.

1.15.2. Determine per attribute what value set will be used

This activity determines per attribute what value set will be eventually used, regarding the possibilities of mappings.

1.15.3. Determine which code system will be applied

This activity determines which code system will be applied for the value sets for each attribute.

1.15.4. Determine which values from the code system are allowed for the concerning attribute

This activity determines which values from the code system are allowed for the concerning attribute.

1.15.5. Determine which values from the data warehouse must be mapped on the values of the code system

This activity determines which values from the data warehouse must be mapped on the values of the code system. In case of multiple candidate fields in the data warehouse, it is indicated how the different fields map on the attributes of the candidate DCM.

1.15.6. Harmonized value sets

This start event indicates the eventual harmonized value sets.

1.16. Needs more information from domain expert?

This gateway checks whether more information is needed from a domain expert to harmonize the value sets.

Output: Yes/No

1.17. Provide feedback

In case the data governance architect is not able to determine certain aspects within the harmonization process of the value sets and therefore needs more information, a domain expert is consulted, who provides feedback.

1.18. Finalize value sets

In case the value sets are harmonized in a satisfied manner by the governance architect, the DCM analyst can finalize the value sets within the DCM. Mainly, the code system has to be adopted and a finishing touch is given to the chapter Information Model.

1.19. Publish Candidate DCM

This activity publishes the Candidate DCM internally so the eMeasure analyst can use this for further practices.

1.20. Published Candidate DCM

The end-event indicates the output of the sub-process 'Make candidate DCM', namely a published candidate DCM.

2. Make Final DCM

This nested activity shows that a final DCM is made. Final means that the DCM is reviewed by a domain expert and all clinical information is added and validated. A final DCM is not necessarily also mapped.

Input: Candidate DCM

Output: Final DCM

2.1. Published Candidate DCM

This start event indicates that the process of making a final DCM starts with a published candidate DCM.

2.2. Send published Candidate DCM to Domain Expert

The DCM analyst sends the published Candidate DCM to a domain expert related to that expertise for review and let him/her add clinical content.

2.3. Parallel gateway

This gateway shows that from this point two activities are performed simultaneously.

2.4. Create DCM content

One of these parallel activities is the nested activity of creating DCM content with regard to the clinical aspects.

2.4.1 Candidate DCM without clinical content

This start-event indicates that the process of creating DCM content starts with a candidate DCM without clinical content.

2.4.2 Fill in chapter 'Content' of the DCM

This activity shows that the chapter 'Content' of the DCM is filled in by the domain expert.

2.4.3 Fill in chapter 'Purpose' of the DCM

This activity shows that the chapter 'Purpose' of the DCM is filled in by the domain expert.

2.4.4 Fill in chapter 'Patient Population' of the DCM

This activity shows that the chapter 'Patient Population' of the DCM is filled in by the domain expert.

2.4.5 Fill in chapter 'Evidence base' of the DCM

This activity shows that the chapter 'Evidence base' of the DCM is filled in by the domain expert.

2.4.6 Fill in chapter 'Instruction' of the DCM

This activity shows that the chapter 'Instruction' of the DCM is filled in by the domain expert.

2.4.7 Fill in chapter 'Interpretation' of the DCM

This activity shows that the chapter 'Interpretation' of the DCM is filled in by the domain expert.

2.4.8 Fill in chapter 'Care Process' of the DCM

This activity shows that the chapter 'Care Process' of the DCM is filled in by the domain expert.

2.4.9 Candidate DCM with clinical content

This end-event indicates that the activity 'create DCM content' is ended with as output a candidate DCM with clinical content.

2.5. Review chapter Information Model of DCM

The other parallel activity that is performed by the domain expert is reviewing the chapter Information Model of the DCM.

2.6. Is Information Model correct?

This gateway checks what the outcome is of the review of the domain expert regarding the Information Model. When the domain expert still sees some flaws, the candidate DCM needs to be modified.

Outcome: Yes/No

2.7. Send back to DCM analyst

In this activity the domain expert sends the Information Model back to the DCM analyst with his comments for modifications to the Candidate DCM.

2.8. Request for Candidate DCM modification

This end-event shows the output of where the process ends when the Information model is not fully correct, namely a request to the DCM analyst for a modification towards the Candidate DCM.

2.9. Parallel gateway

This gateway shows that the previously done activities have to be completed first before the process can be continued.

2.10. Finalize DCM

In this activity the DCM is finalized by putting all information in the correct chapters of the DCM and metadata (e.g. author, version update etc.) is added.

2.11. Check final DCM

The domain expert does a final quality check on the finalized DCM.

2.12. Is DCM ok?

This gateway checks whether the final DCM is approved by the domain expert. If not, the DCM is sent back to the DCM analyst to modify the DCM.

Output: Yes/No.

2.13. Publish final DCM

If the final DCM is fully approved, it will be published, so that the eMeasure analyst can make the final eMeasure-DCM connection.

2.14. Published Final DCM

This end-event shows the output of the sub-process 'Make final DCM', namely a publicly published final DCM.

3. Map DCM

This nested activity shows that a candidate DCM is mapped onto the existing data warehouse, so actual data can be generated from the DCM.

Input: Candidate DCM

Output: Mapped DCM

3.1. Candidate DCM

This start-event indicates that the process of 'Map DCM' starts with a candidate DCM.

3.2. Map DCM to data

In this activity the eventual determined value sets with their related mappings from the DCM are actually mapped on the fields of the existing data warehouse by the DCM analyst.

3.3. Evaluate Mapping

To verify the mapping of the previous activity, an evaluation is done by the database developer. This is the person who is responsible for the data at the source concerning the mapped fields, i.e. the existing data warehouse. This is a technical evaluation where is tested whether the mapping results in the correct data generation.

3.4. Mapping OK?

This gateway checks whether the mapping is accepted by the database developer or not. In case this is not the case, the mapping has to be modified according to the feedback of the database developer.

Output: Yes/No

3.5. Notify eMeasure analyst

If the mapping is approved by the database developer, the DCM analyst notifies the eMeasure analyst that the mapping is completed and he/she can generate the data for the eMeasures.

3.6. Mapped DCM

This end-event indicates that the output of the activity 'Map DCM' is a mapped DCM.

4. Parallel gateway

This parallel gateway shows that when a candidate DCM is published, three activities are done in parallel, namely 'connect eMeasure to candidate DCM' by the eMeasure analyst and 'Make final DCM' and 'Map DCM' by the DCM analyst.

5. Request for candidate DCM modification?

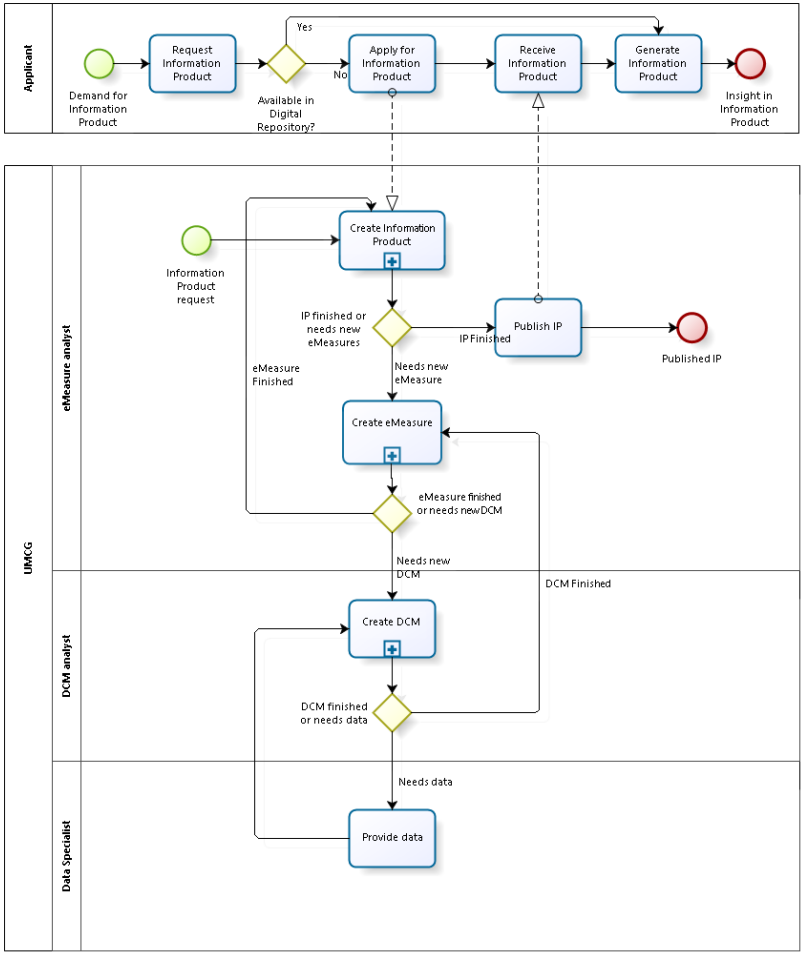
This gateway checks which of the two end-events in the nested activity 'Make final DCM' is chosen.

Output: Yes/No

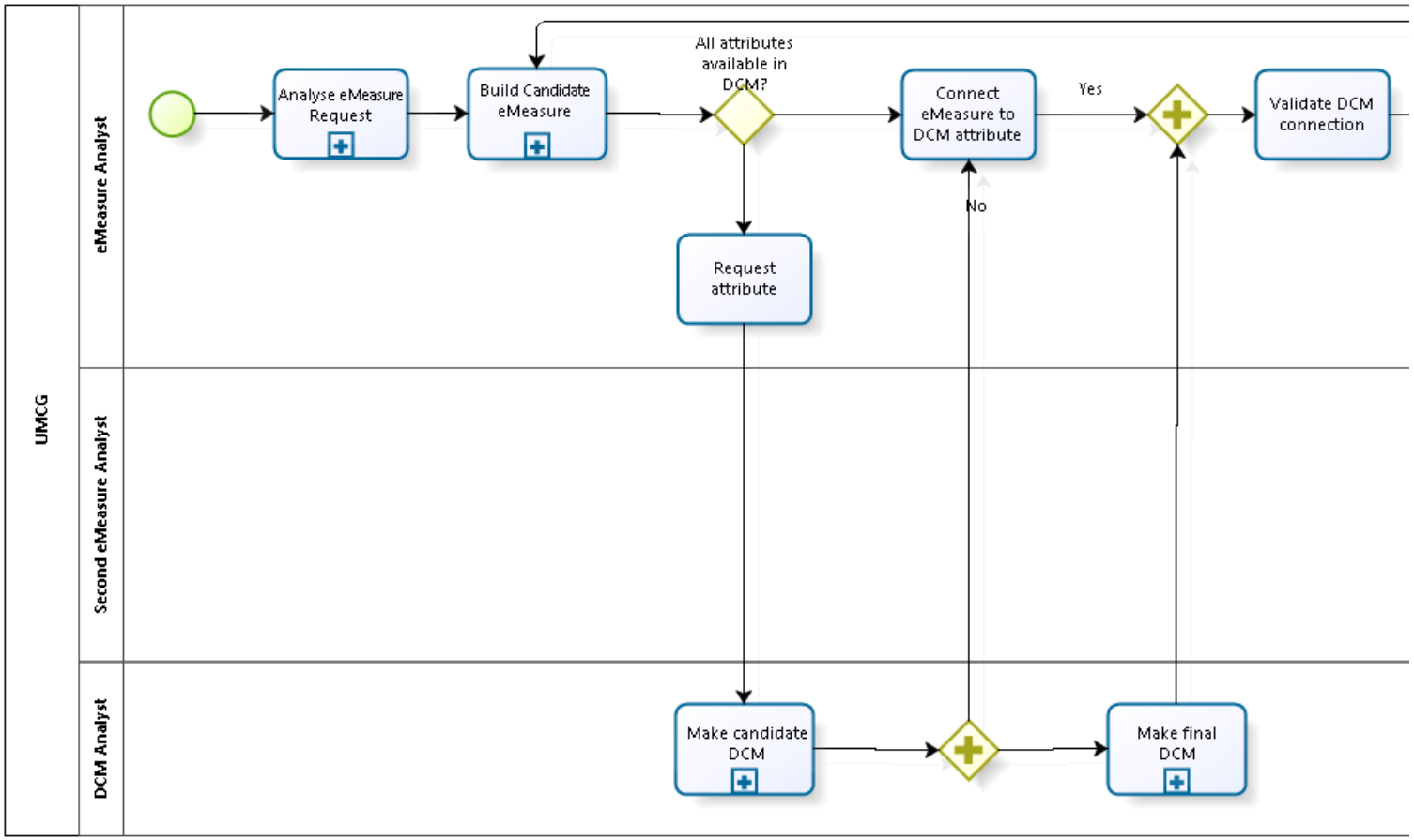
APPENDIX IV: PROCESS MODELS

i. FIRST PROCESS MODELS

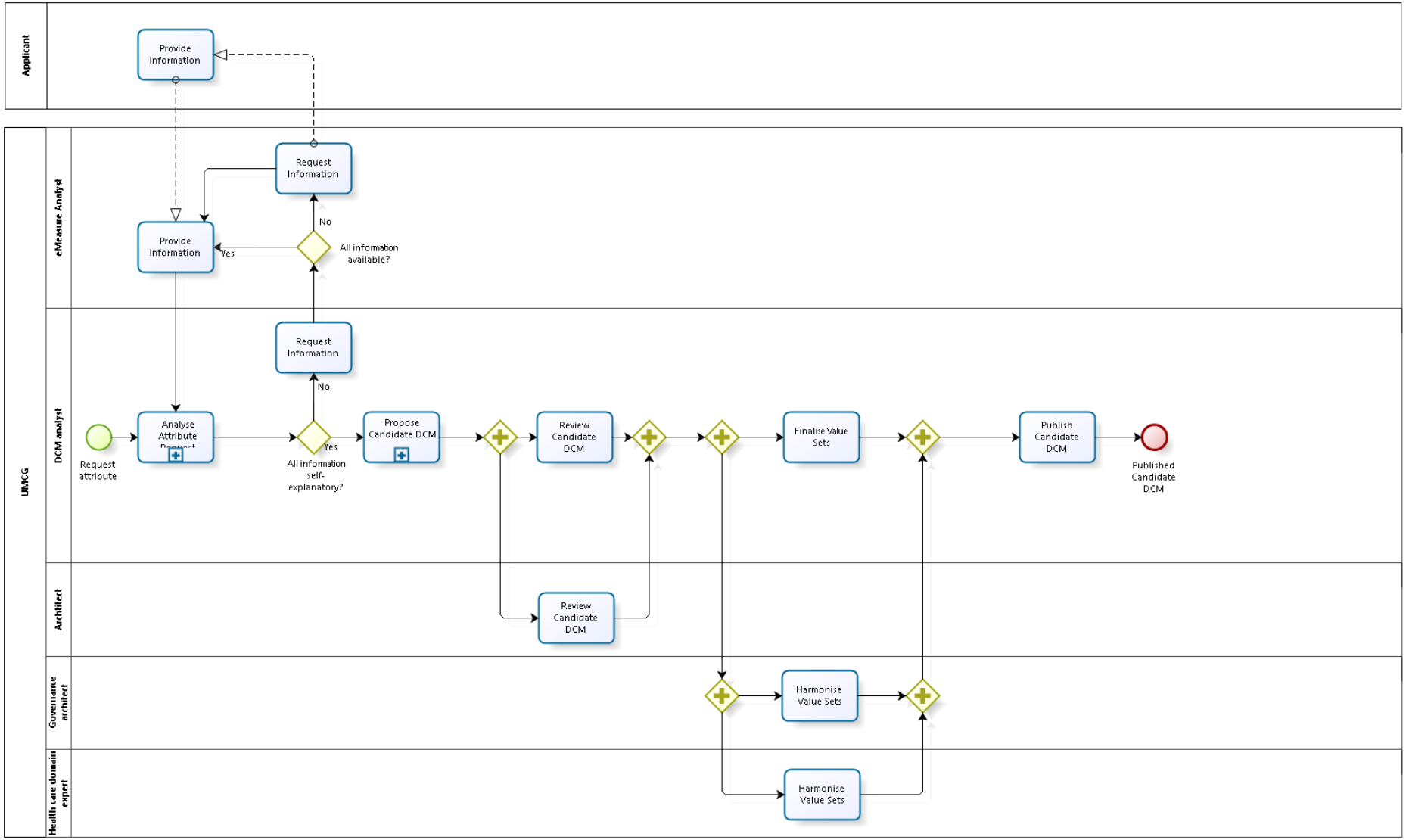
a. Process Overview



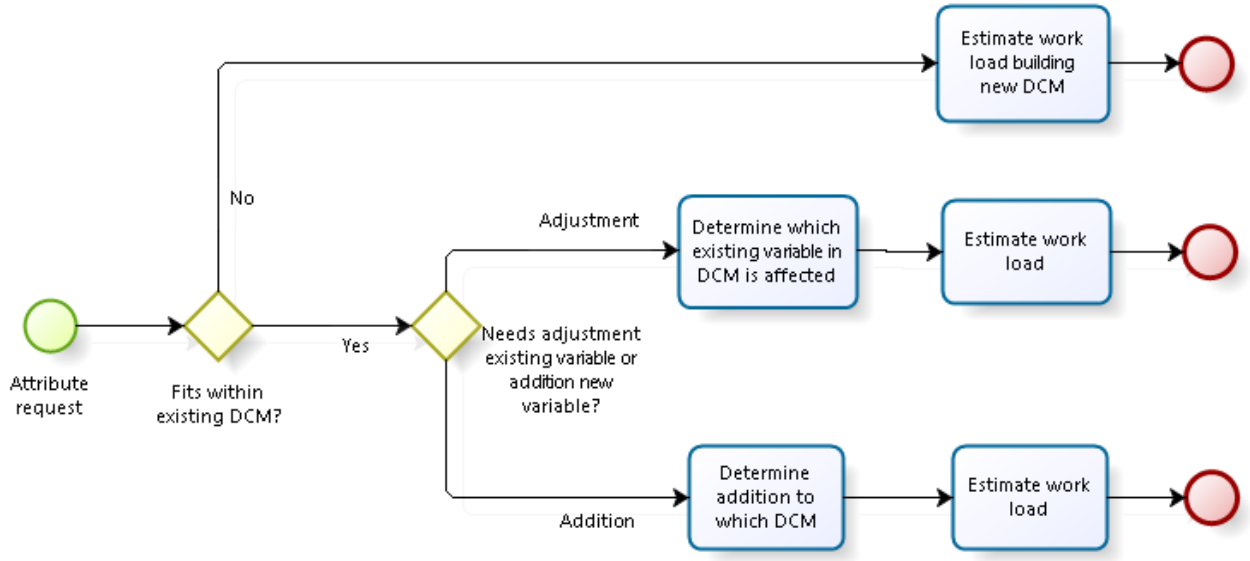
b. Decoupling point between eMeasure development process and DCM development process: Create eMeasure



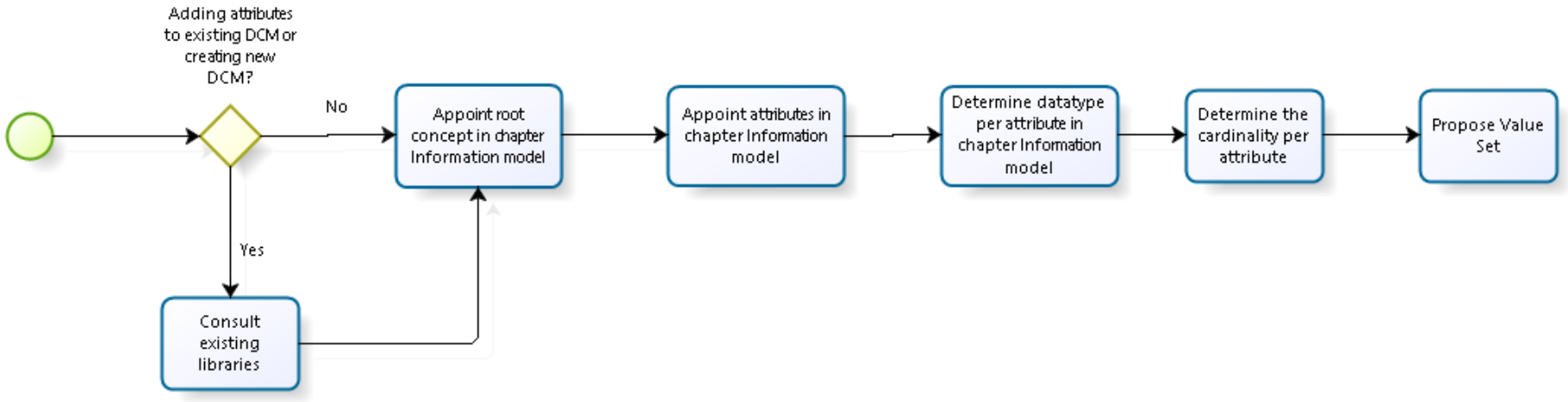
c. Make Candidate DCM



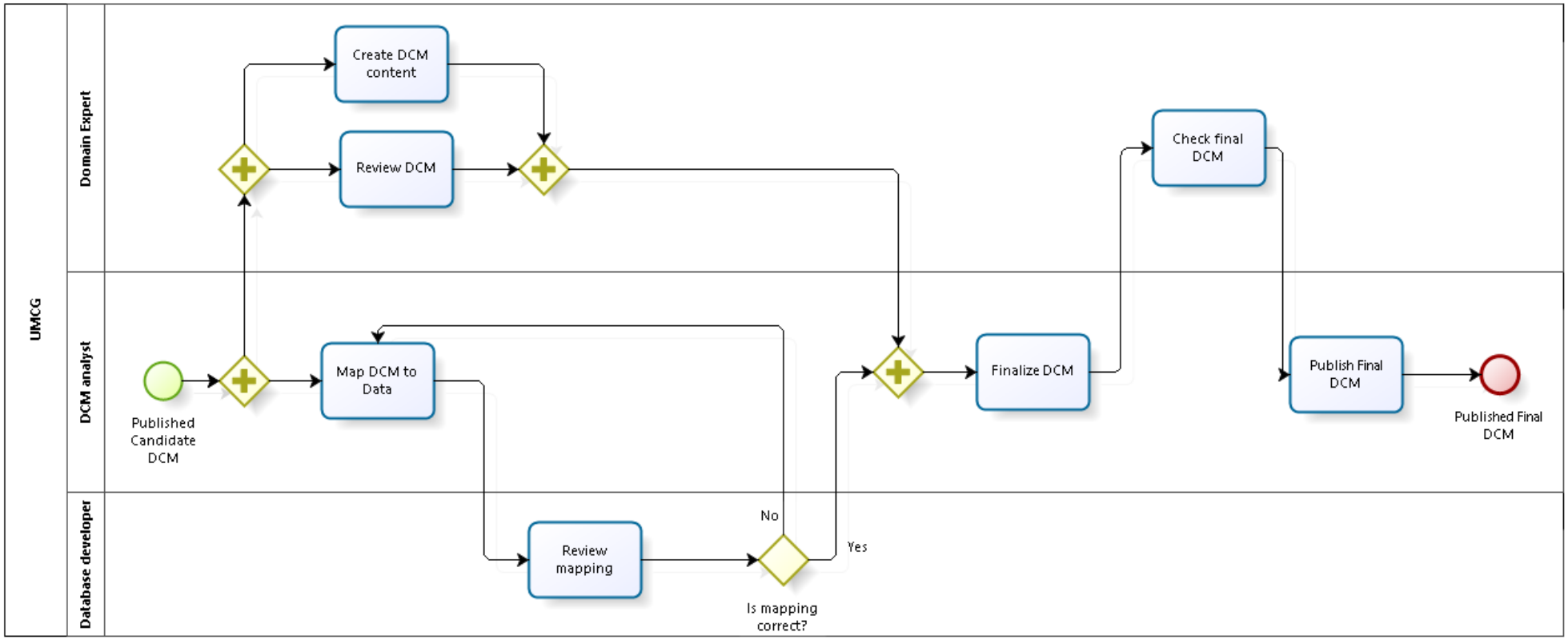
d. Analyze attribute request



e. Propose candidate DCM

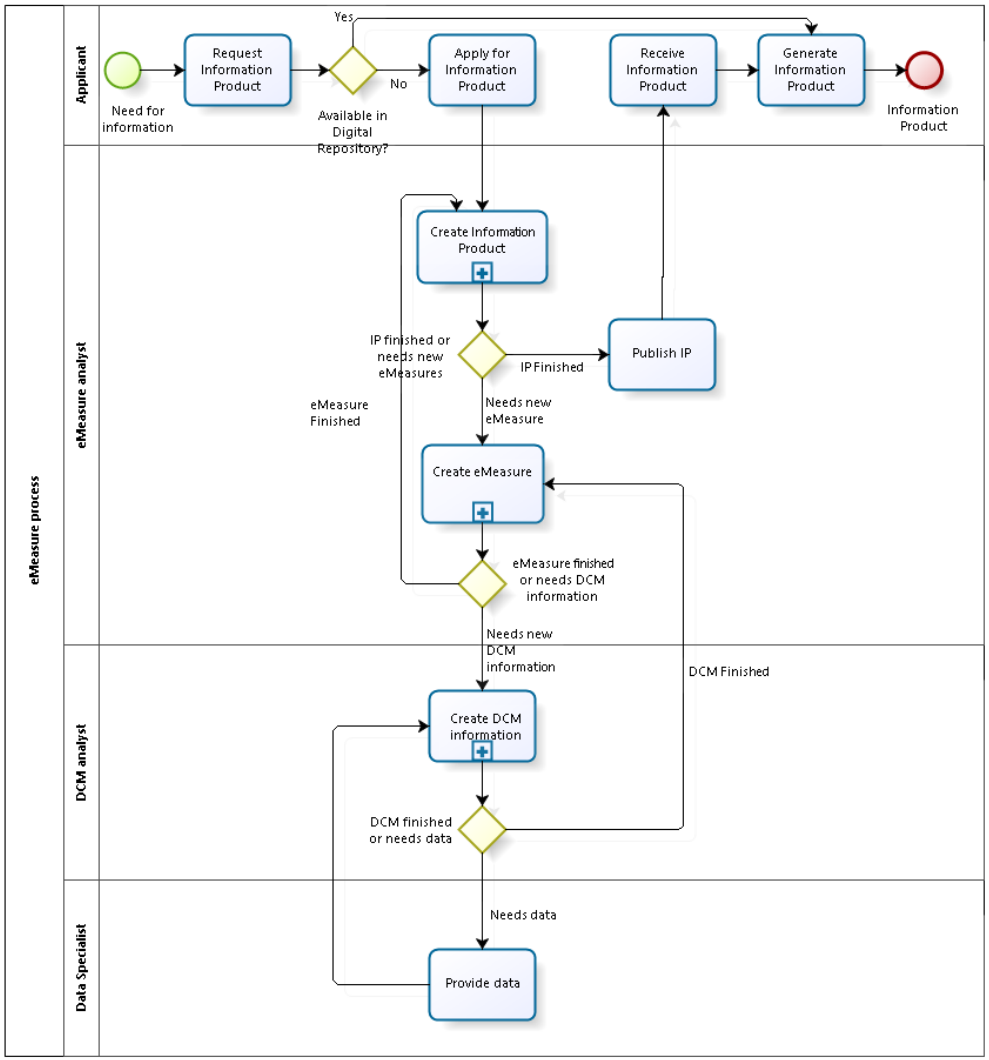


f. Make Final DCM

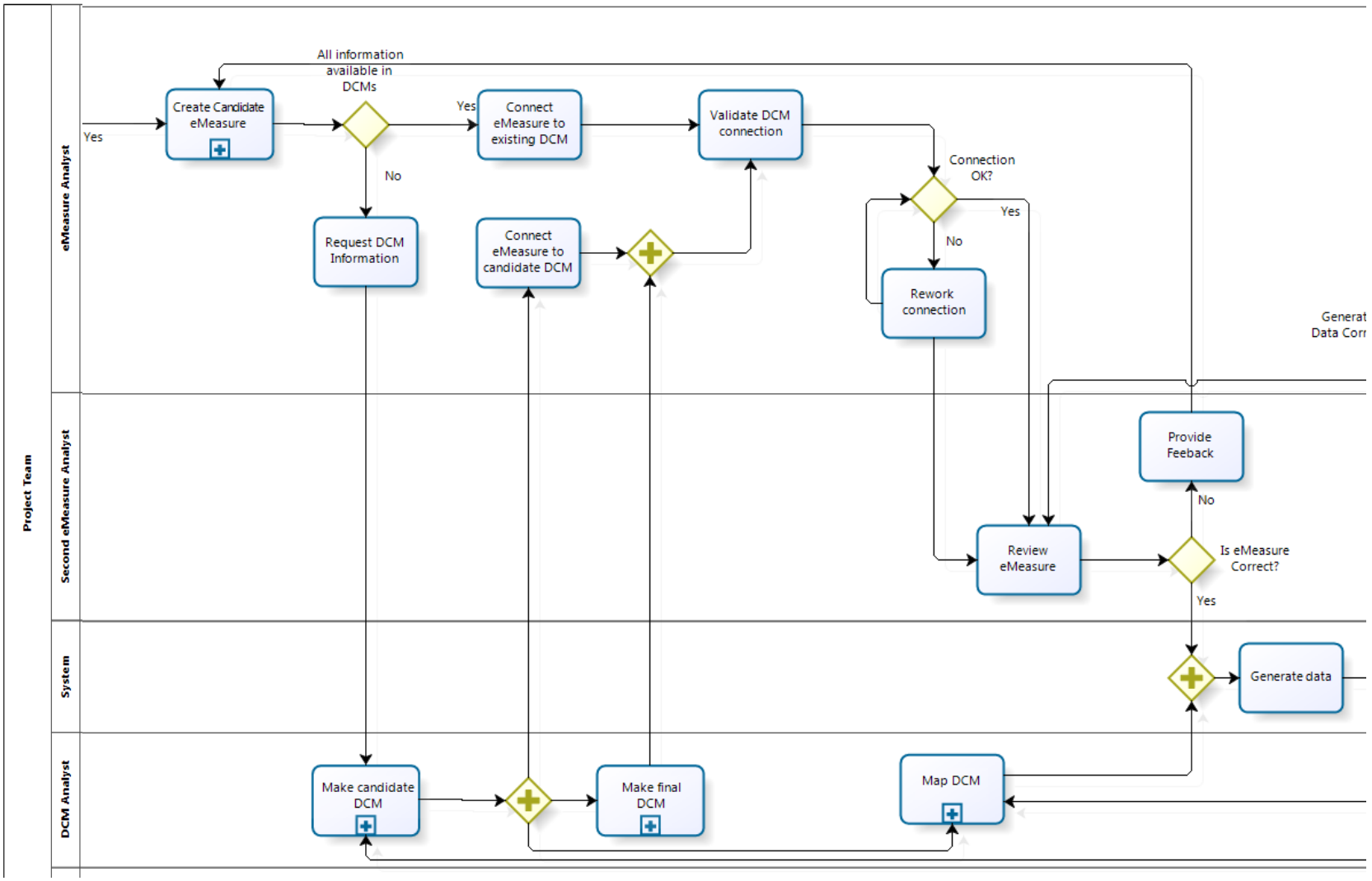


ii. PRELIMINARY PROCESS MODELS

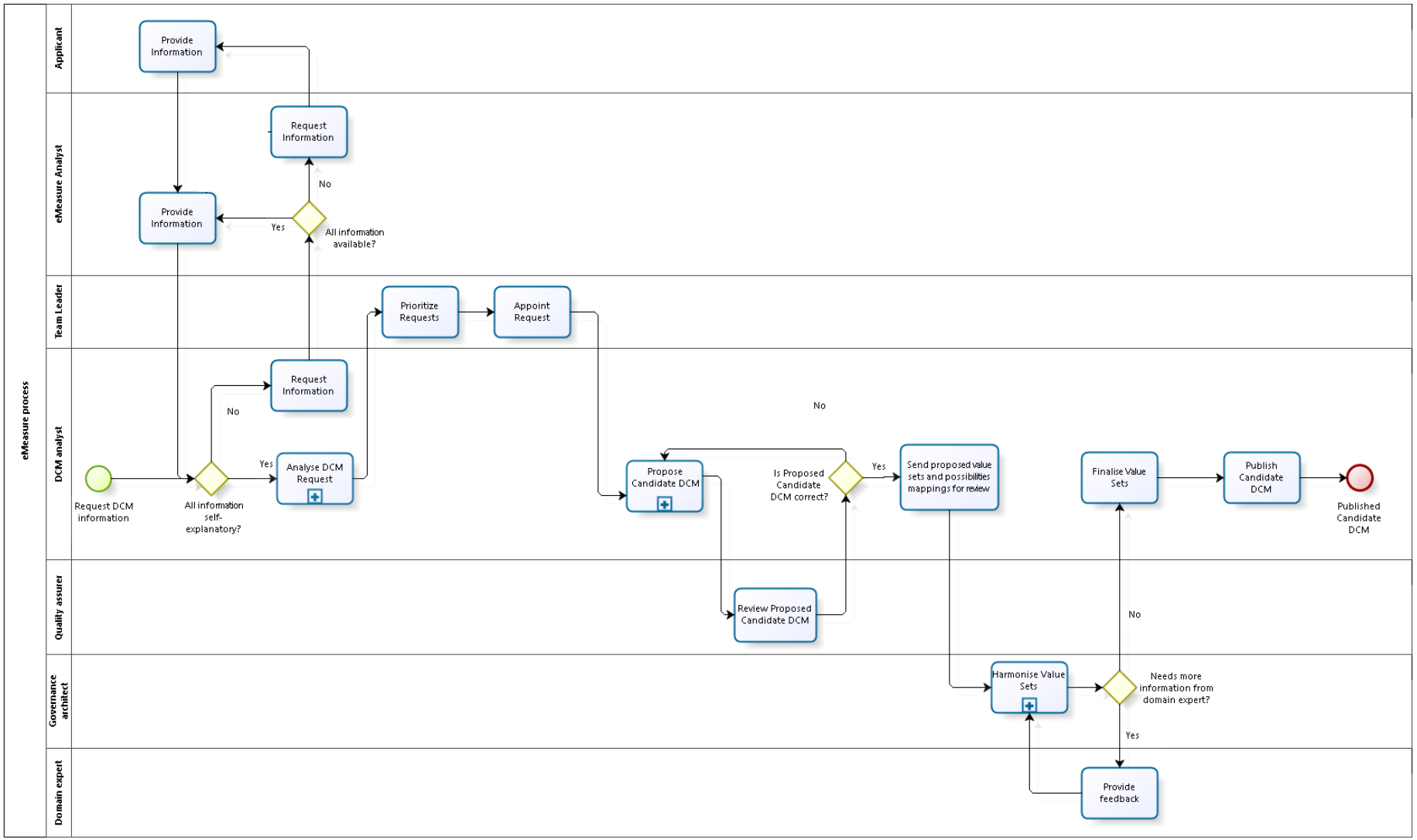
a. Process overview



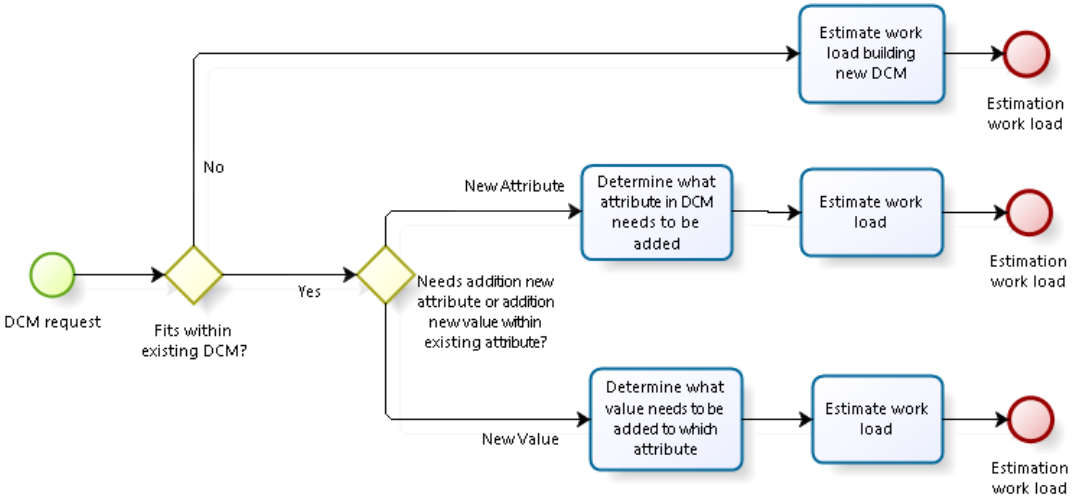
b. Decoupling point between eMeasure development process and DCM development process: Create eMeasure



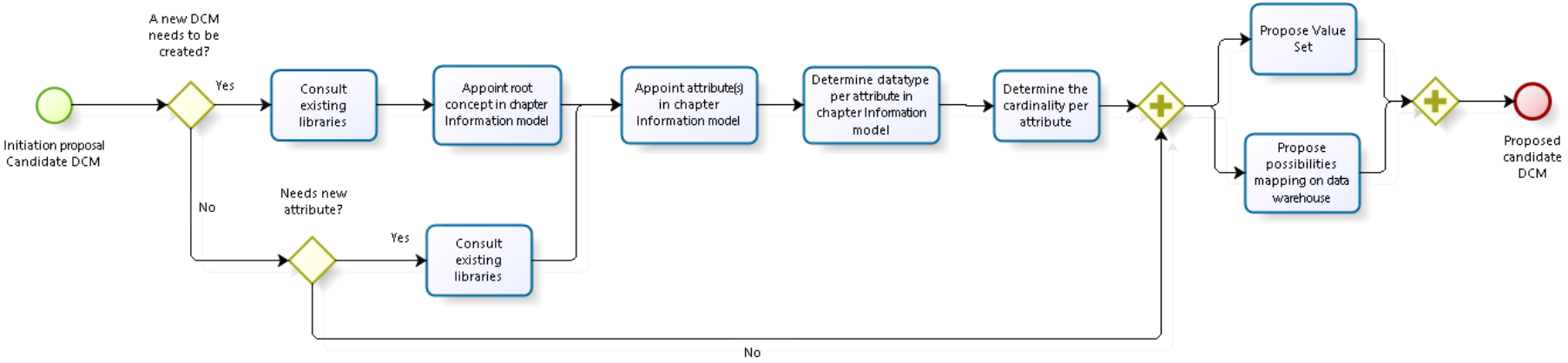
c. Make Candidate DCM



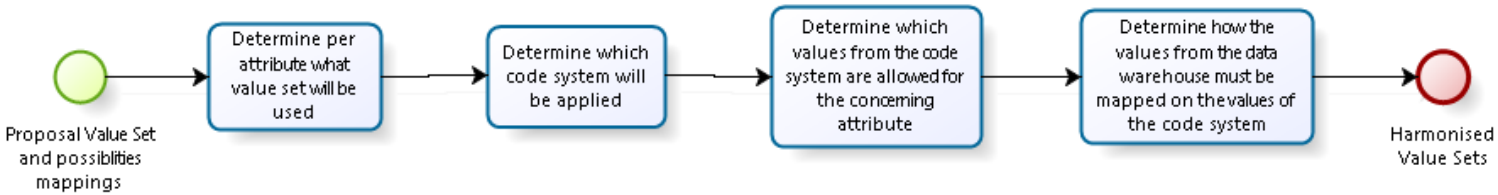
d. Analyze DCM Request



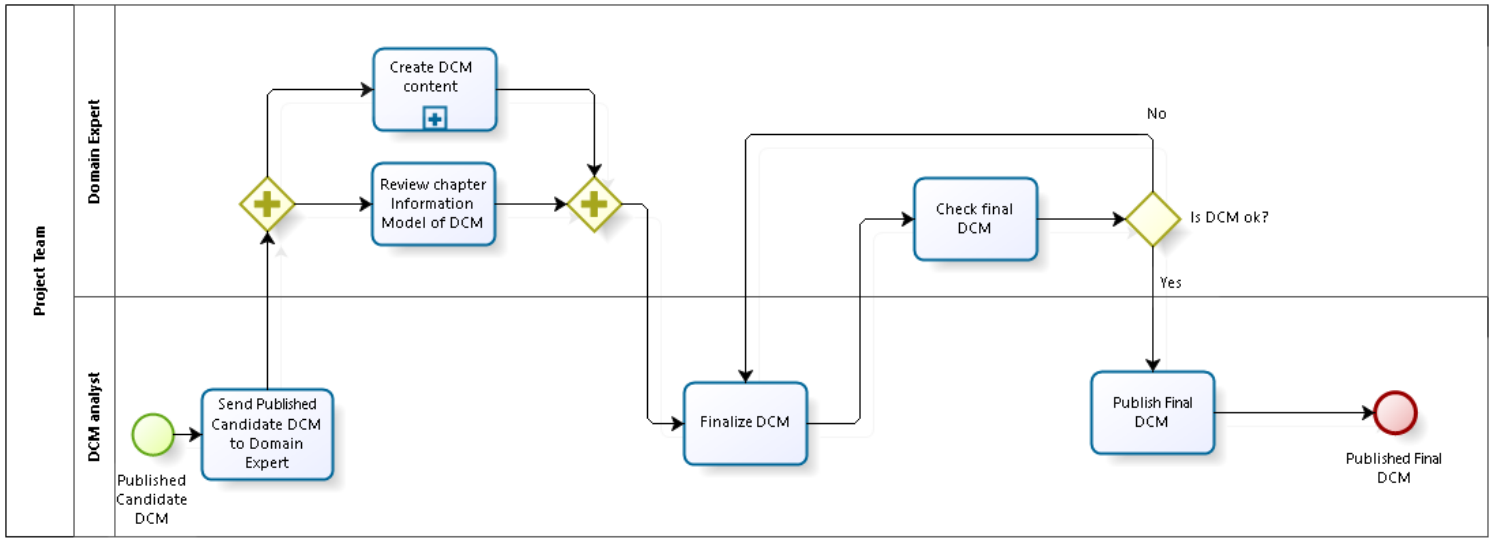
e. Propose Candidate DCM



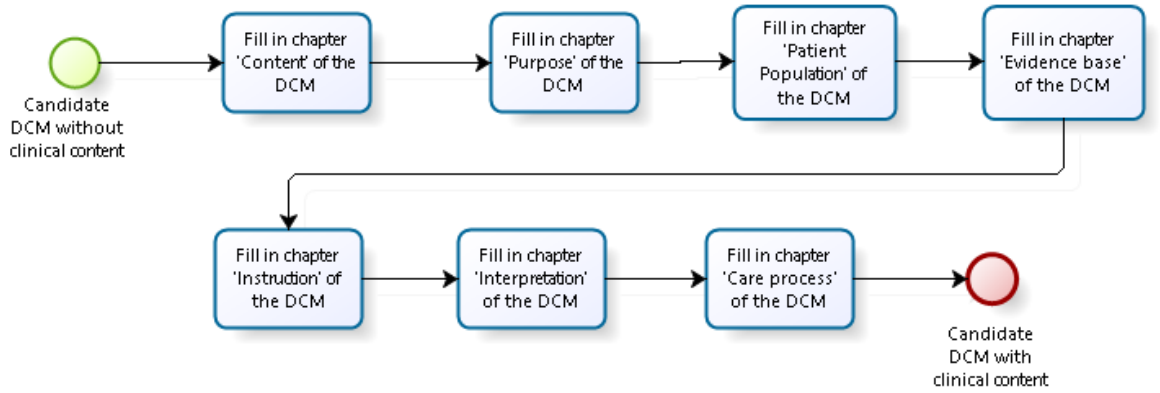
f. Harmonize Value Sets



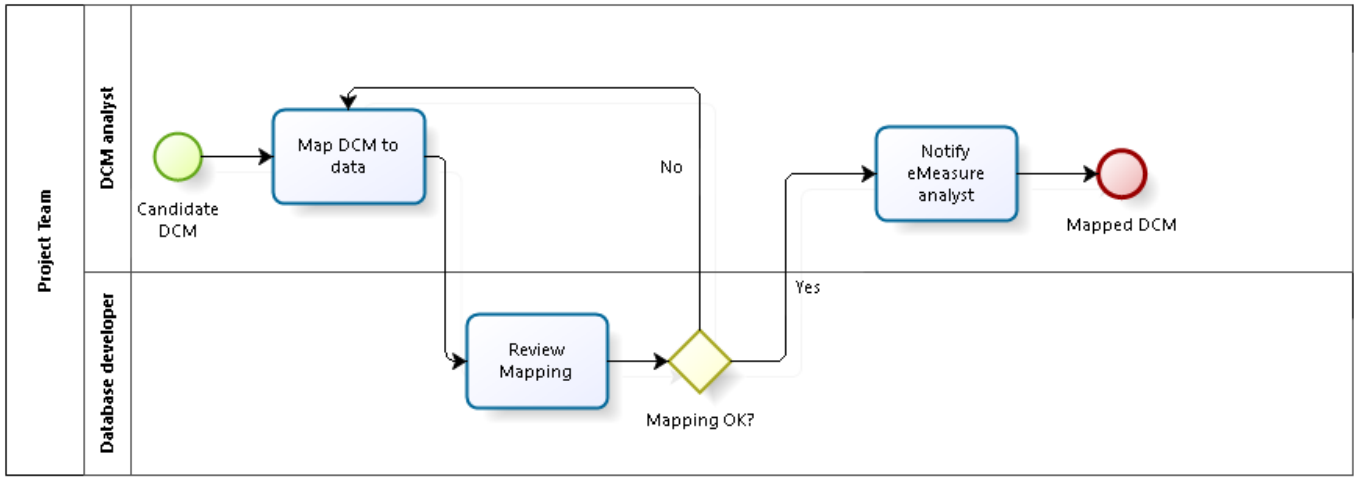
g. Make Final DCM



h. Create DCM Content

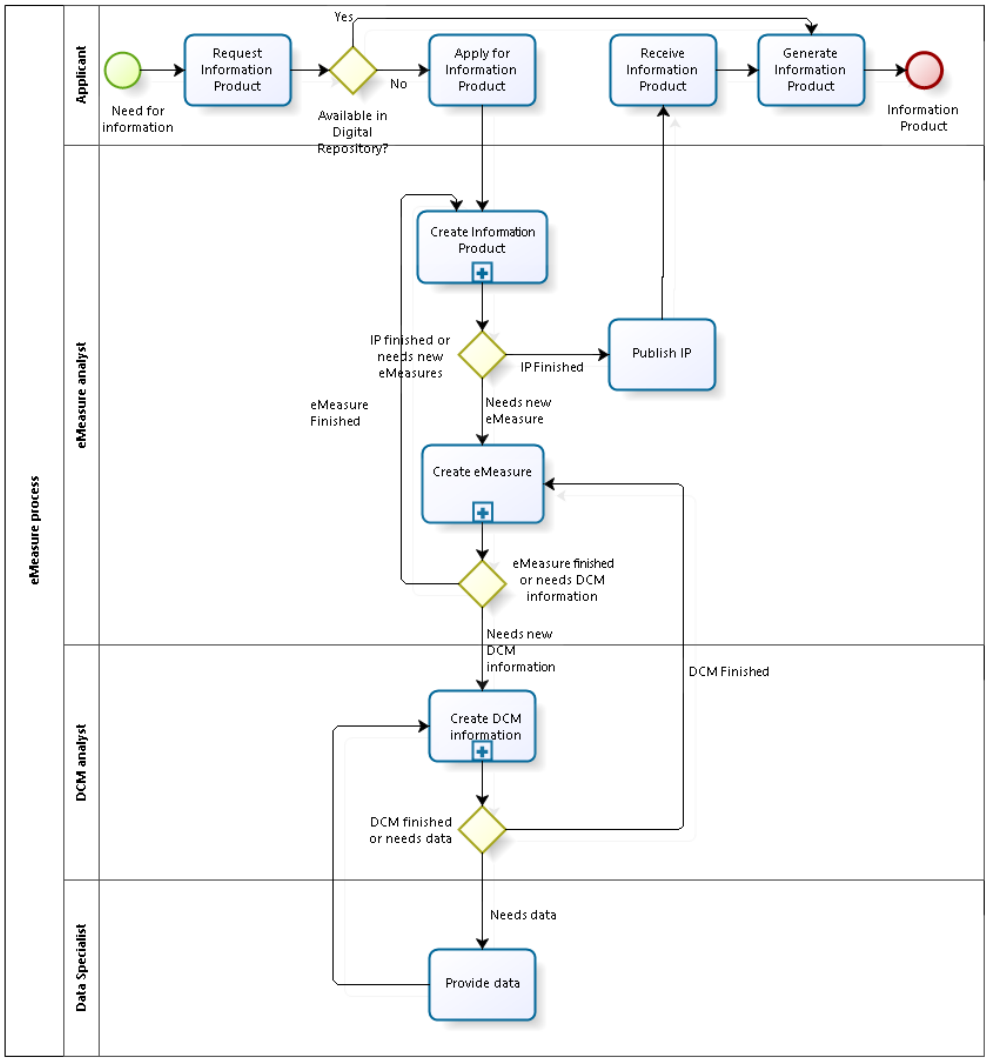


i. Map DCM

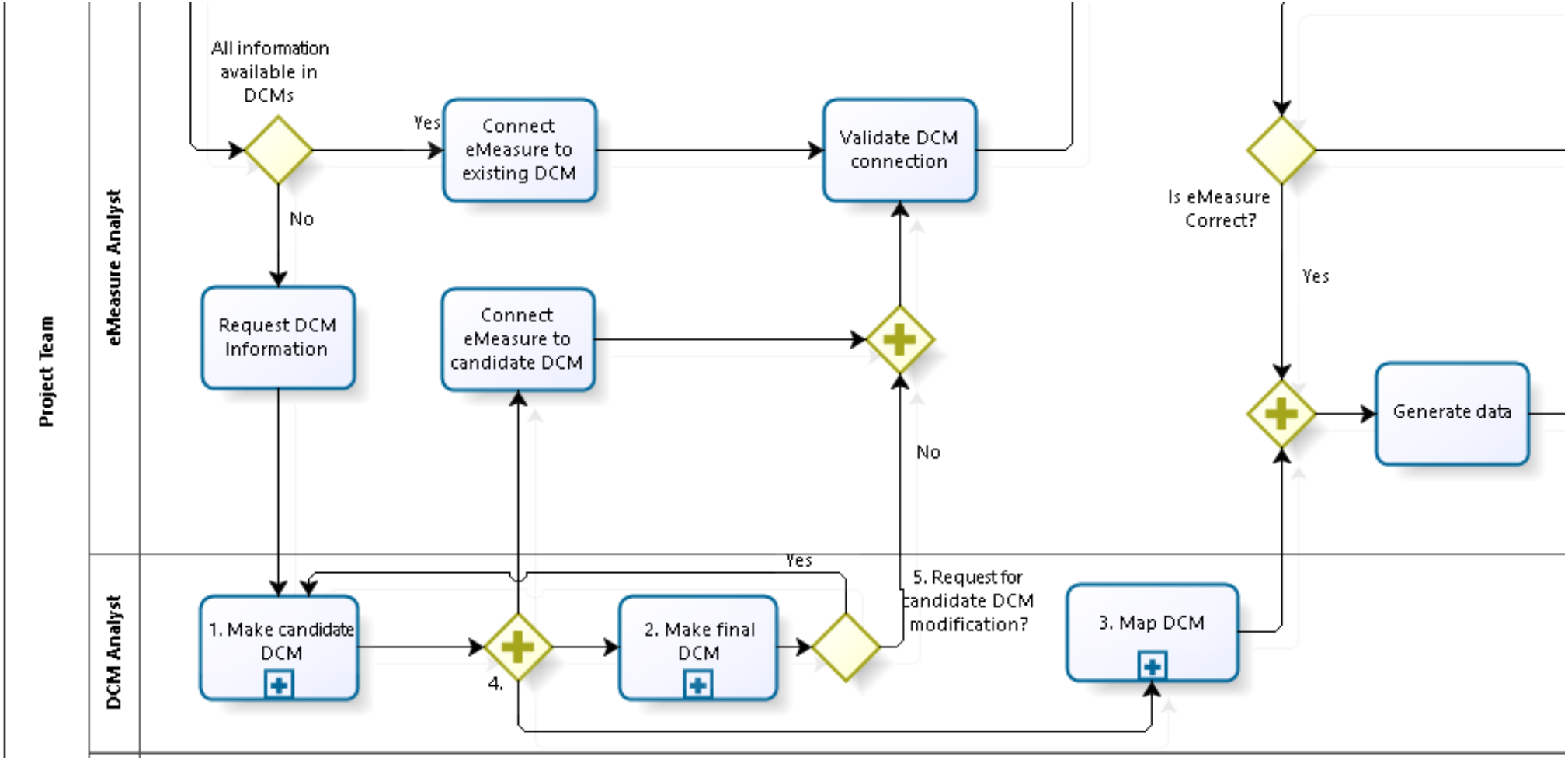


iii. FINAL PROCESS MODELS

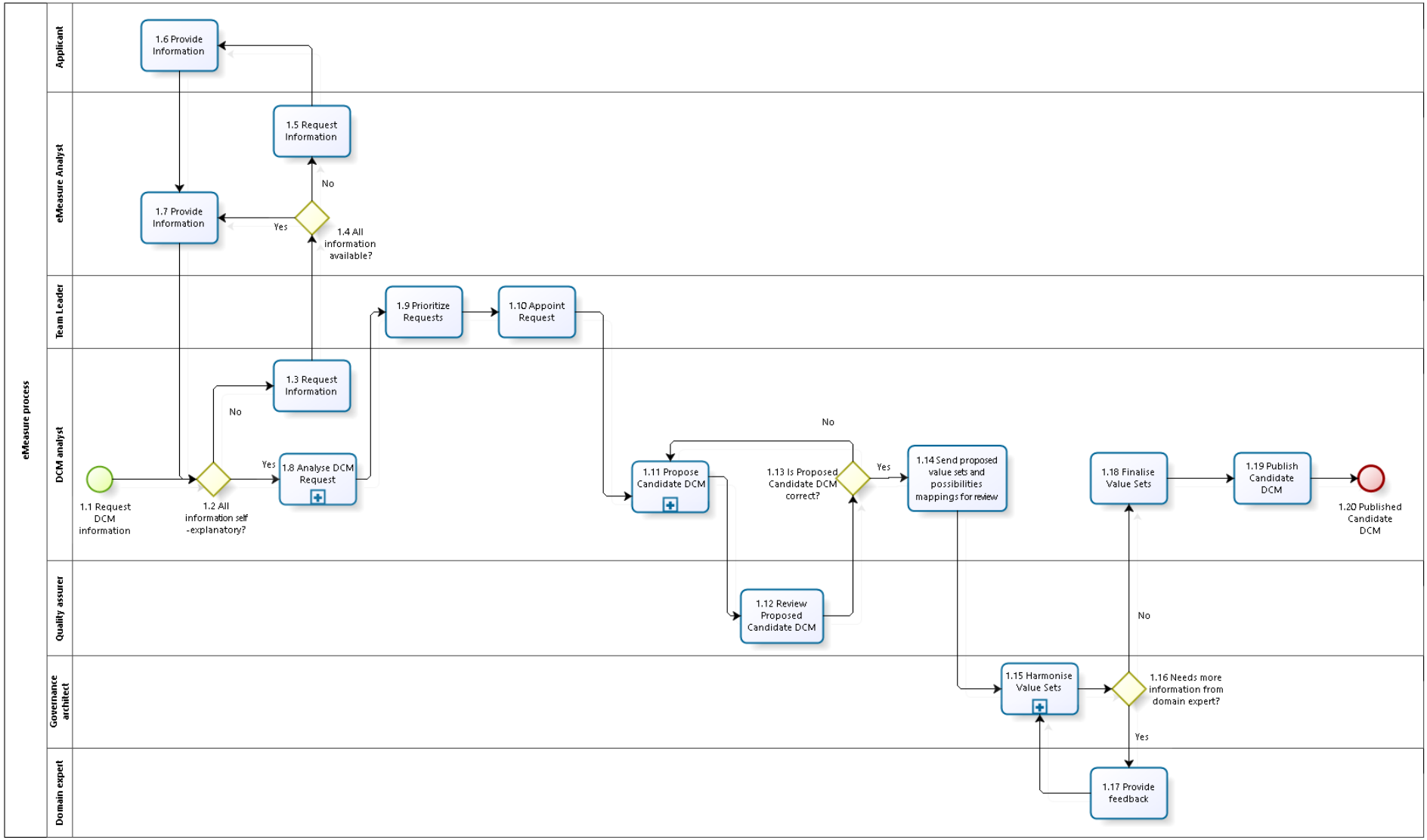
a. Process overview



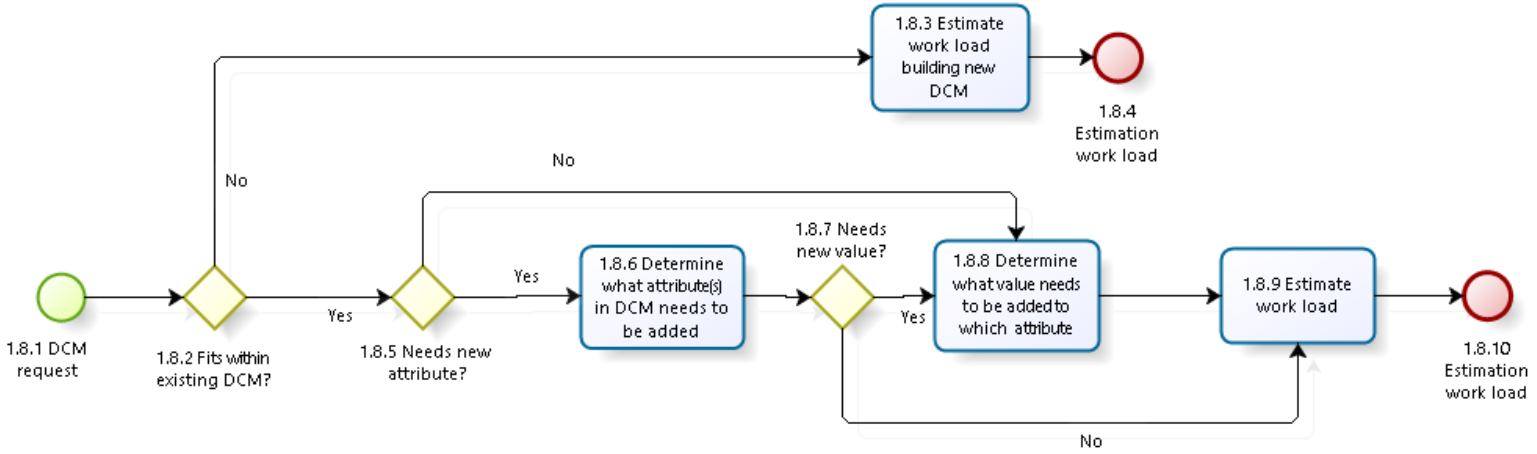
b. Decoupling point between eMeasure development process and DCM development process: Simplified 'Create eMeasure'



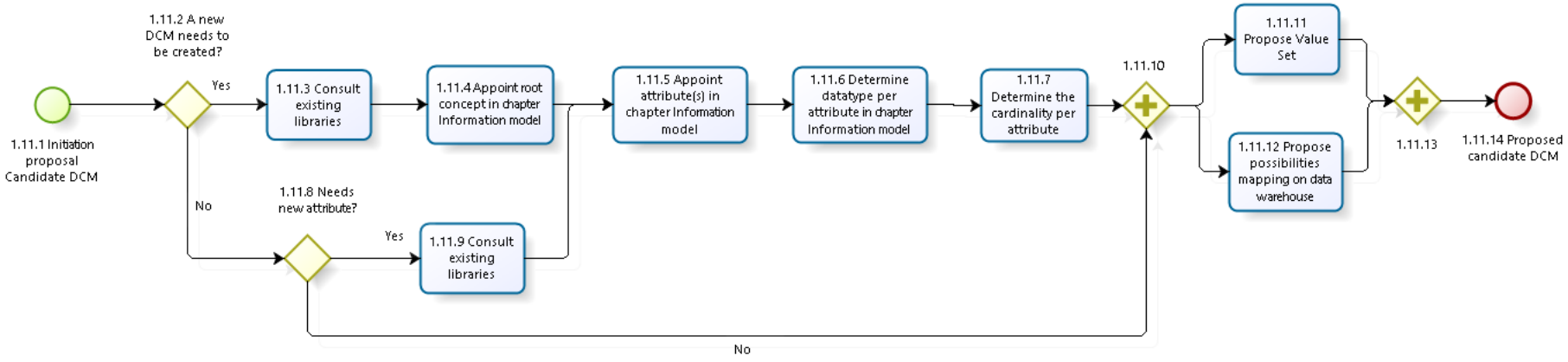
c. Make Candidate DCM



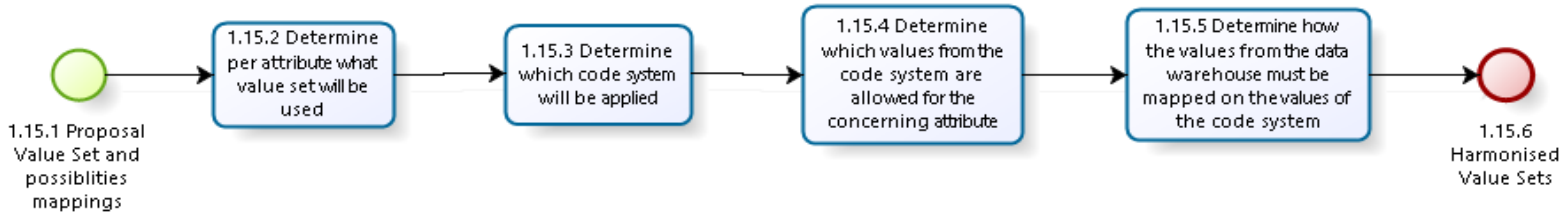
d. Analyze DCM Request



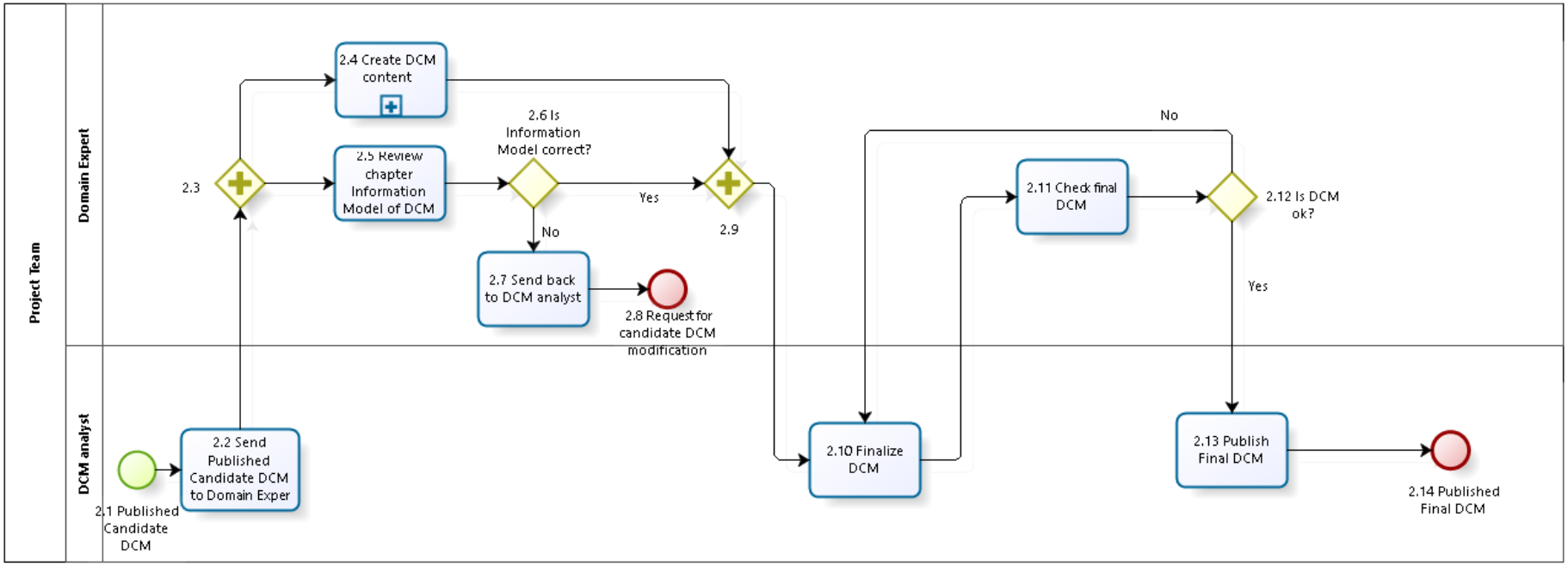
e. Propose Candidate DCM



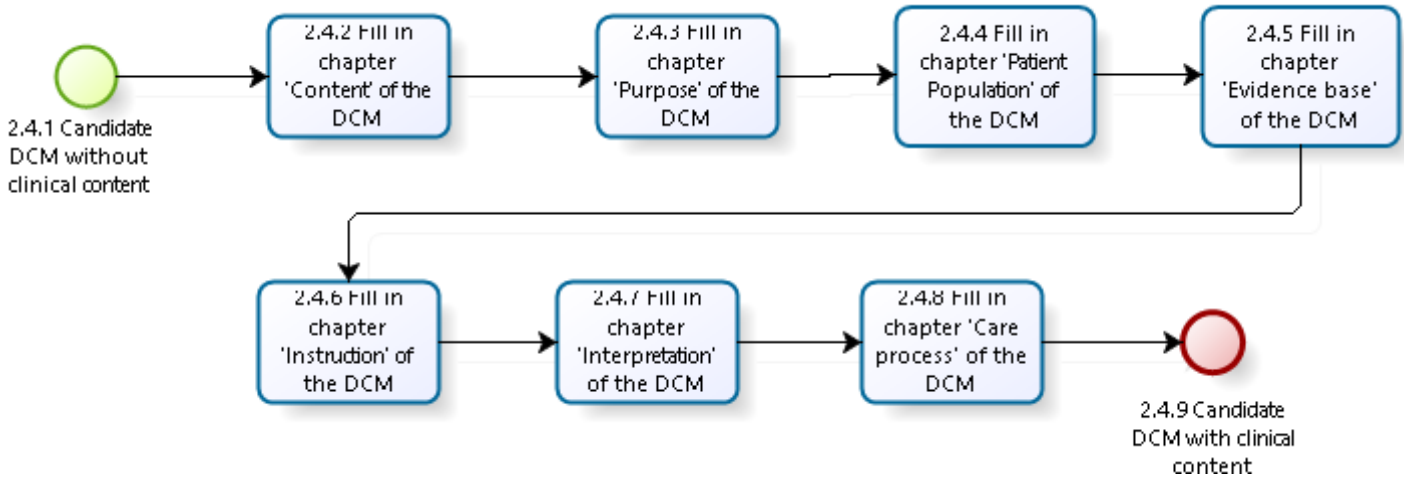
f. Harmonize Value Sets



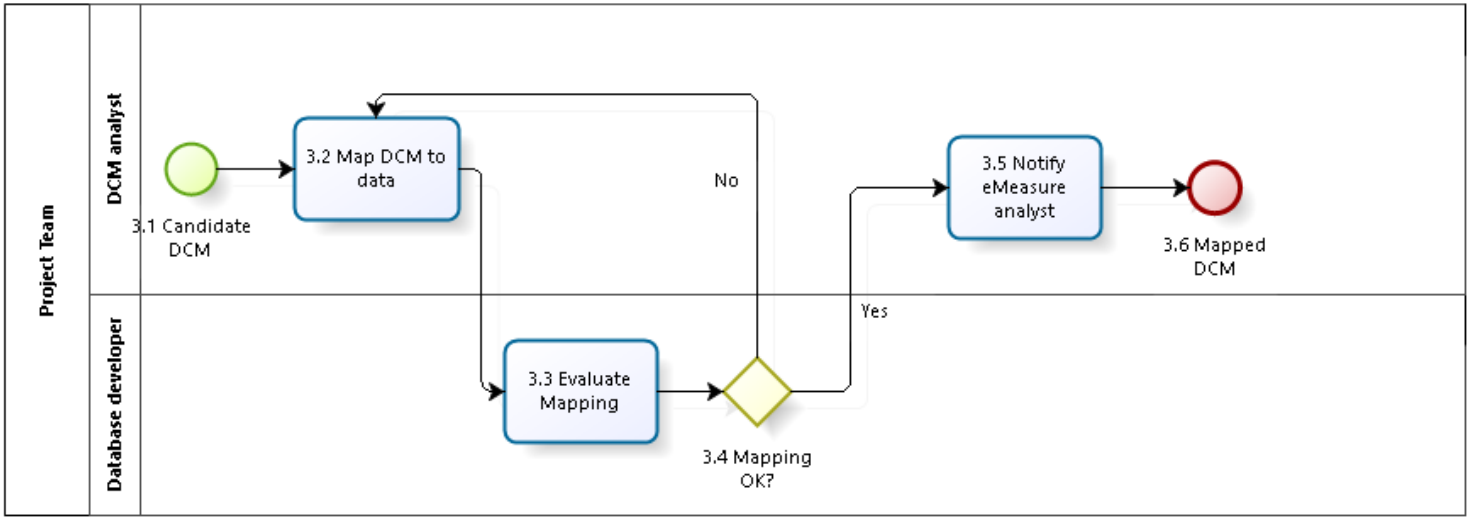
g. Make Final DCM



h. Create DCM Content



i. Map DCM



APPENDIX V: PRACTICAL EXAMPLE THROUGH PROCESS

This appendix describes how an example walks through the designed process to create clearer insights of how the process looks like in a more practical way. The information request chosen is from an instance from the government (Zichtbare Zorg) which concerns a knee replacement indicator set 1 with respect to the reporting year of 2013. The information request is shown below in Dutch (since this is how it was received by the LTHN):

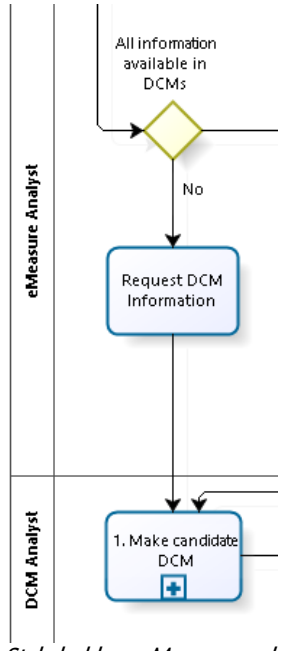
Indicator 1: Antibiotische profylaxe	
Relatie met kwaliteit van zorg	Antibiotische profylaxe is bewezen effectief in de preventie van diepe wondinfecties bij plaatsing van een totale knieprothese. Het optreden van diepe wondinfecties wordt mede beïnvloed door aanwezige co-morbiditeit (o.a. hypertensie, diabetes, obesitas). Deze antibiotische profylaxe moet wel op het juiste ogenblik wordt gestart: 60 tot 15 minuten vóór de incisie of vóór het opwekken van bloedleegte kan als het optimale tijdsinterval worden beschouwd. Het percentage wondinfecties dat na opereren optreedt weerspiegelt de kwaliteit van de geboden profylaxe. Overigens moet er rekening mee worden gehouden dat ook patiëntenkenmerken zoals de gezondheidsstatus het optreden van wondinfecties kunnen beïnvloeden.
Operationalisatie 1a	Is er een richtlijn of protocol beschikbaar voor antibiotische profylaxe in geval van een totale knieprothese? *Ja/Nee*
Operationalisatie 1b	Percentage operaties waarbij de patiënt peri-operatief antibiotica toegediend heeft gekregen, in geval van een totale knieprothese.
Teller 1b	Aantal operaties waarbij de patiënt peri-operatief antibiotica toegediend heeft gekregen, in geval van een totale knieprothese
Noemer 1b	Aantal operaties waarbij de patiënt een totale knieprothese heeft ondergaan
Operationalisatie 1c	Percentage operaties waarbij de patiënt 60 tot 15 minuten vóór de incisie of vóór het opwekken van bloedleegte antibiotica toegediend heeft gekregen, in geval van een totale knieprothese.
Teller 1c	Aantal operaties waarbij de patiënt 60 tot 15 minuten vóór de incisie of vóór het opwekken van bloedleegte antibiotica toegediend heeft gekregen, in geval van een totale knieprothese
Noemer 1c	Aantal operaties waarbij de patiënt peri-operatief antibiotica toegediend heeft gekregen, in geval van een totale knieprothese
Operationalisatie 1d	Percentage diepe wondinfecties in geval van een totale knieprothese
Teller 1d	Aantal diepe wondinfecties tot zes weken na de operatie bij patiënten in geval van een totale knieprothese
Noemer 1d	Aantal operaties waarbij de patiënt een totale knieprothese heeft ondergaan
Definities	Peri-operatief: Gedurende de klinische opname De volgende definitie (WIP) van een diepe wondinfectie is van toepassing. De infectie is ontstaan binnen 1 jaar na operatie en de infectie lijkt het gevolg te zijn van de operatie en betreft de diepliggende weefsels van de incisie (zoals fascie en spier) en voldoet bovendien aan één of meer van de volgende bevindingen: 1) Purulente afscheiding uit een diepe incisie maar niet van de organen en anatomische ruimten van het operatiegebied. 2) Spontane wonddehiscentie of wond geopend door de chirurg terwijl de patiënt koorts (>38°C) en/of lokale pijn of gevoeligheid heeft tenzij een wondkweek negatief blijkt.

	3) Abces of ander teken van infectie van het gebied van de diepe incisie gezien bij directe observatie, tijdens heroperatie of histopathologisch of radiologisch onderzoek. 4) Diagnose 'diepe infectie van het operatiegebied' door de chirurg of behandelend arts. NB: Infecties die zowel oppervlakkig als diep zijn worden geclassificeerd als diepe postoperatieve infecties van het operatiegebied.
In-/exclusiecriteria	1d: Exclusie: Patiënten met ASA-klasse > 3
Bron	1a: Richtlijnen of protocollen 1b: Datamanagementsysteem anesthesiologie, anesthesielijst in patiëntendossier (teller), ZIS, DBC- en verrichtingenregistratie (noemer) 1c: Datamanagementsysteem anesthesiologie, anesthesielijst in

Initially this request is received by the hospital and forwarded to the eMeasure analyst. Since the process of an information product request and the development of it are studied by Van de Laar (2015), his thesis gives further details on this matter. However, from the process that is part of Van de Laar (2015), it turns out that most DCM's are available such as 'Patient', 'Operation' (in Dutch: verrichting), 'Medicines' to derive the necessary eMeasures, except for one that includes 'deep-wound infection'. That is where the process of this thesis comes in, where the eMeasure analyst requests the DCM analyst to come up with a DCM that includes this necessary value.

On the next pages the designed process will be walked through step by step, by showing the BPMN model on the left, explained with a general description in the middle and a more specified description related to the example on the right. This is done, so the relatively abstract process comes more to life and creates a better understanding. The numbering in the BPMN models gives guidance to both the sequence of activities and gateways as the level of depth of a nested activity.

BPMN model



Stakeholders: eMeasure analyst, DCM analyst

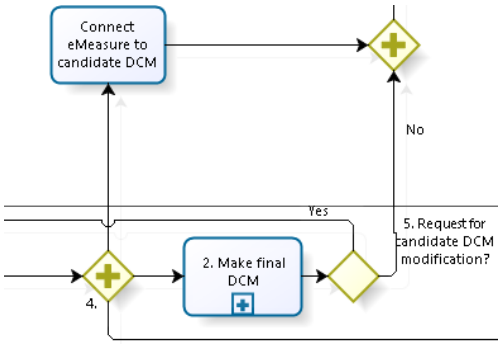
General description

The process described in this thesis only comes in place, when the eMeasure analyst does not have all information available in the already existing DCM's. When something is missing, he/she requests this information to the DCM analyst.

The DCM analyst receives this request and is responsible to ensure this missing information will be implemented in a DCM. In order to this, he first makes a candidate DCM, i.e. he does a proposal for the missing information. This will be discussed more in detail later, since it is a nested activity.

Specified on example

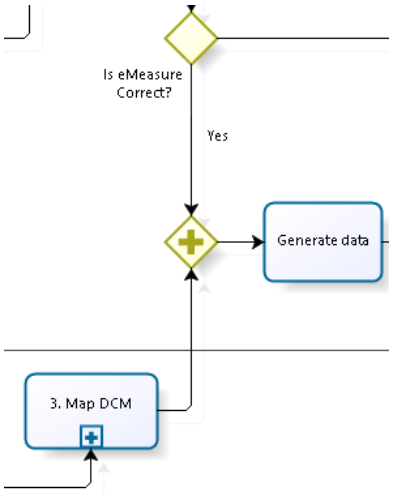
In the case of the information request of an instance from the government, all necessary information is available in DCM's, like 'Patient', 'Operation' (in Dutch: verrichting), 'Medicines', except for a DCM containing something regarding a 'deep-wound infection'. Within the DCM 'Patient' it is possible to exclude patients with ASA-class > 3 (urgency level). Within the DCM 'Operation' one can filter on only 'knee replacement surgery' and within DCM 'medicines' the treatment of antibiotic prophylaxis can be found. For the 'deep-wound infection' criteria a candidate DCM needs to be made.



Stakeholders: eMeasure analyst, DCM analyst

After a candidate DCM is made, three activities will be performed in parallel. One is that the eMeasure analyst already connects its eMeasure to the candidate DCM. Another one is that a final DCM is made by the DCM analyst. In case within this activity it turns out that a modification is needed in the candidate DCM, this will be send back to initial stage of making a candidate DCM.

The candidate DCM containing 'deep-wound infection' can be used to already connect to eMeasure 1d so time savings can be realized. Furthermore, the DCM needs to be made definite/final, which will be explained more in detail later. When no modification needs to be made in the candidate DCM after making the DCM final, the eMeasure can continue its procedure, explained by Van de Laar.

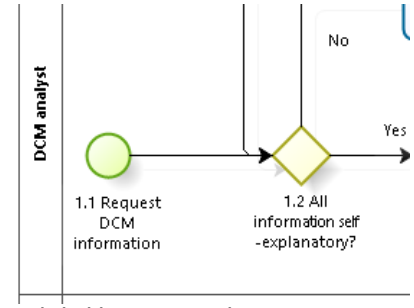


Stakeholders: eMeasure analyst, DCM analyst

The third parallel activity after the candidate DCM is made available, is the mapping of the DCM onto the data warehouse. This means that the DCM, which represents the desired ideal data from the total rough data, is actually linked with the existing database. When this mapping is done, the eMeasure analyst can generate the data for further actions.

Once the candidate DCM is made, the DCM containing the information about 'deep-wound infection' needs to be transformed from an empty shell to a data-generating infrastructure. Therefore correct linkages need to be made from the DCM to the existing data warehouse which contains this information. This will probably be, inter alia, in the database that is filled by the orthopedic surgeon who executes knee replacements.

BPMN model



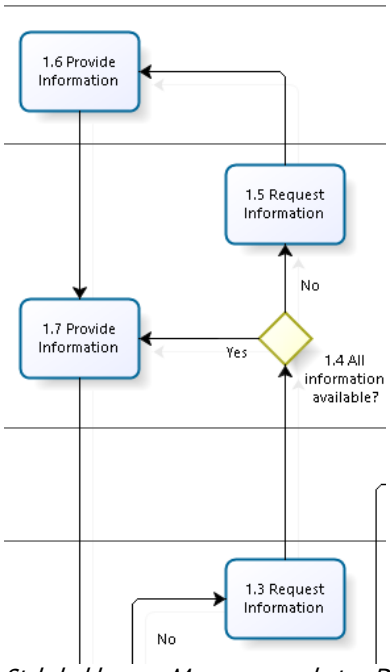
Stakeholder: DCM analyst

General description

From the numbering it can be seen that this entails a more detailed look on 'Make candidate DCM'. It starts with the request of an eMeasure analyst for DCM information that is not yet available. In order to come up with the correct information, this information request needs to be crystal clear to the DCM analyst.

Specified on example

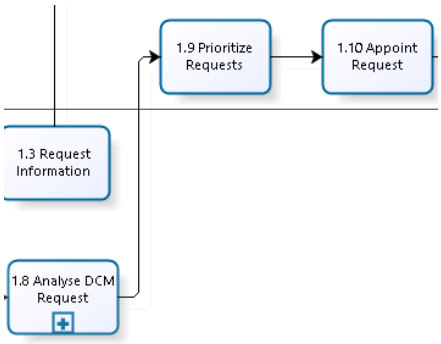
A request comes in for the DCM analyst to realize 'deep-wound infection' within some DCM, since this is missing. First, the DCM analyst wonders whether he understands exactly what this request means. Since he is not an expert on this field, he may not fully understand what is asked for.



Stakeholders: eMeasure analyst, DCM analyst, applicant

In case the DCM analyst does not fully understand what the request entails, he request explanation/information from the eMeasure analyst. She, on its turn, first checks whether she can answer this question, since she received the information request from the applicant. In case of any ambiguities, she contacts the applicant to provide more information about the concerning unclarity. The eMeasure analyst functions as the bridge between DCM analyst and applicant to sustain existing relationships.

For our example, the DCM analyst wants to know what is exactly meant by 'deep-wound infection'. He therefore requests the eMeasure analyst information about the definition of infection and what is exactly meant by 'deep'. Does this relate to number of millimeters or layers of skin etc.? In case the eMeasure analyst cannot answer to these questions, she forwards these to the applicant.

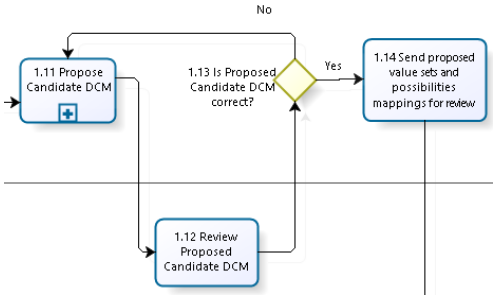


Stakeholders: DCM analyst, Team leader

When all information related to a DCM information request is clear to the DCM analyst, he analyses the request by estimating what the work load would be. He sends this information to the team leader, who will on basis of the estimated work load and due date of the eMeasure prioritize the requests and appoint them again to his team of DCM analysts.

When the DCM analyst knows what is exactly meant by 'deep-wound infection', he can analyze the request and estimate its workload. Since this is a nested activity, this will be explained more in detail later. From the estimated workload and the due date of this eMeasure on March 1st, the team leader will mark this as a top priority request and appoints it to its most experienced DCM analyst.

BPMN model



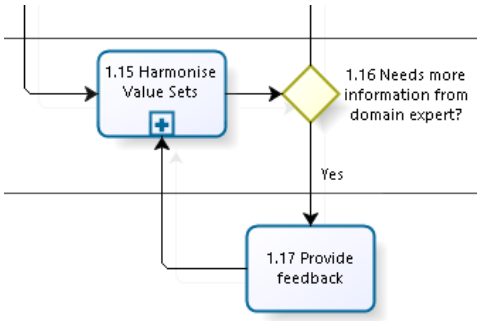
Stakeholder: DCM analyst, Quality Assurer

General description

The DCM analyst proposes a candidate DCM, which will be discussed more in detail later (nested activity). This proposal will be reviewed by a quality assurer on structure and whether it follows the standards of HL7 and Nictiz. When incorrect, this is send back with feedback to the DCM analyst. When correct, the proposed candidate DCM will be sent to a data governance architect.

Specified on example

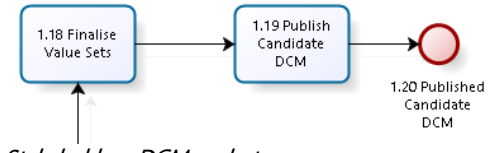
A proposal of a new DCM is made containing 'deep-wound infection' as one of the values within a certain attribute within the DCM 'infection'. More explanation is given in the sub-processes of this nested activity. This proposal is send to the quality assurer who finds no inadequacies related to structure and the standards of HL7 and Nictiz and approves. Next, the DCM analyst sends his proposed value sets and possibilities for mapping to the data governance architect for review.



Stakeholders: Data governance architect, Domain expert

The data governance architect is responsible for harmonizing the value sets and mappings. From the proposed candidate DCM, he determines what is eventually used and what is not. Since it often concerns a complex clinical context, a domain expert is consulted in case of ignorance or unclarities.

The data governance architect harmonizes proposed values by for example, eliminating 'long infection' and 'blatter infection' from the value set, since these already occur in another DCM or are not relevant for this information request. An orthopedic surgeon will be consulted for the value sets, whereas a database developer is consulted that knows where mappings of 'deep-wound infections' can be found.

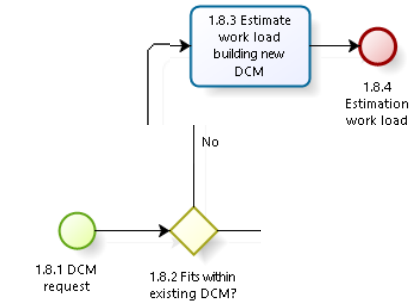
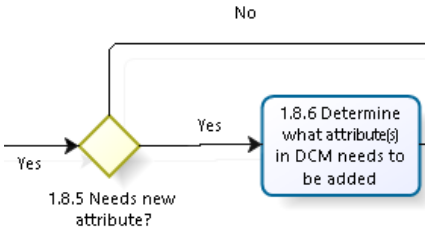
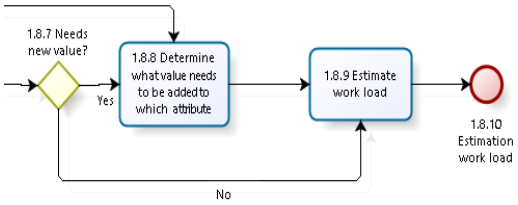
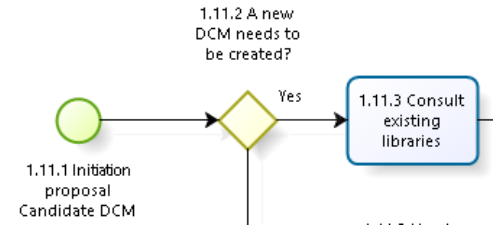
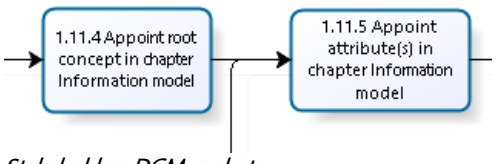
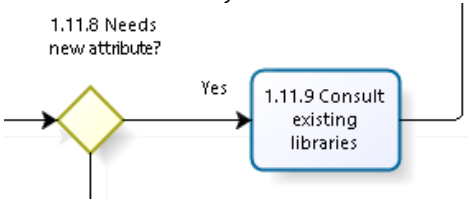


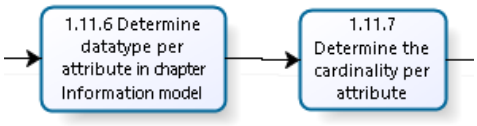
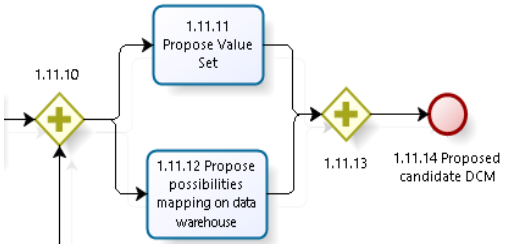
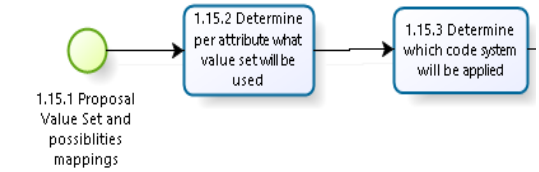
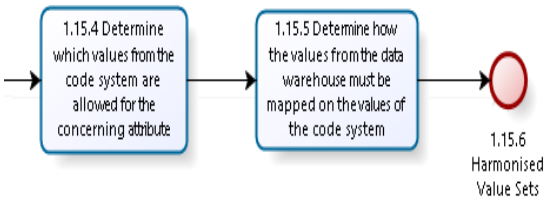
Stakeholders: DCM analyst

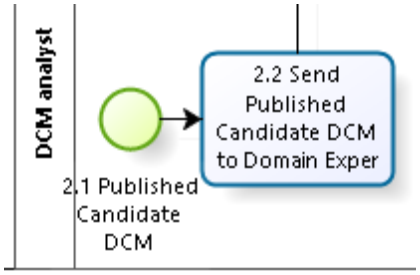
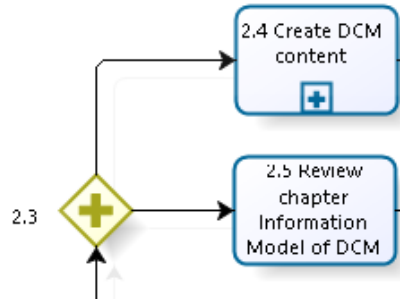
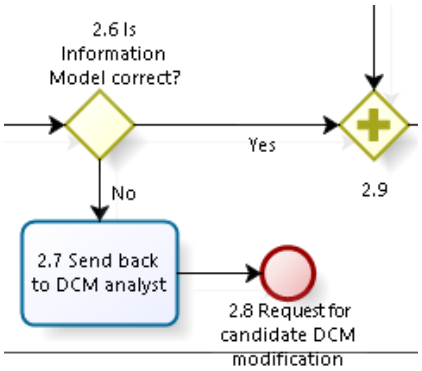
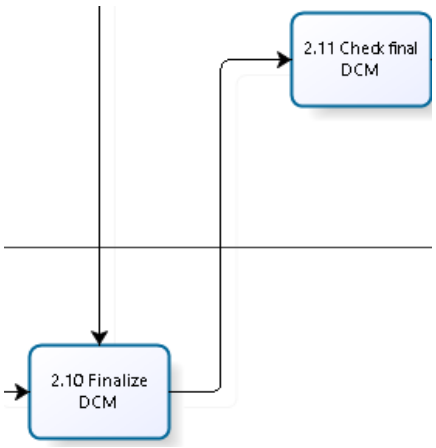
When the value sets are harmonized by the governance architect, the DCM analyst finalizes these by adding metadata.
Then the candidate DCM is ready to be internally published for further actions.

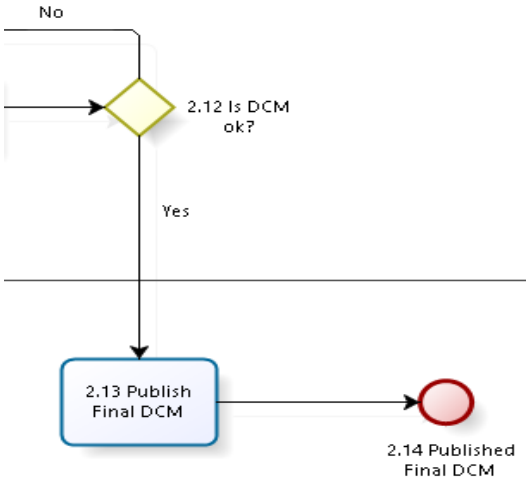
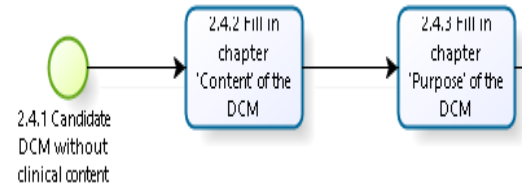
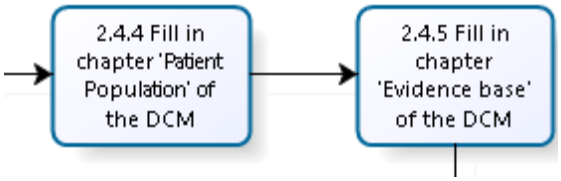
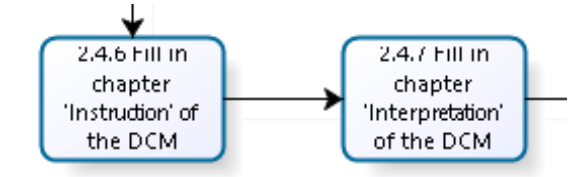
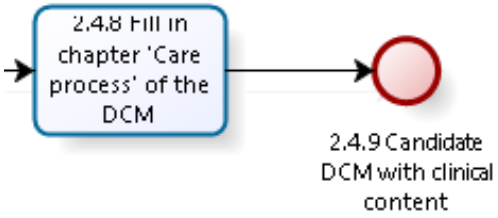
The DCM analyst puts its name, Mr X. on the candidate DCM, update it to version 1.1 and order the values in descending alphabetic order.

He then publishes the candidate DCM on the internal Wiki of the LTHN.

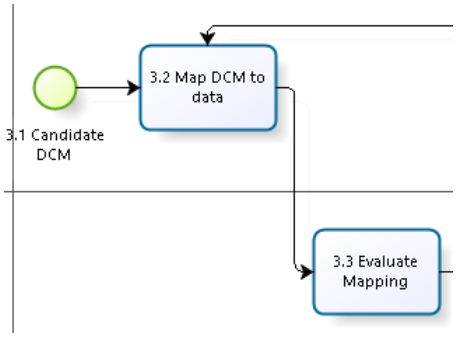
BPMN model	General description	Specified on example
 *Stakeholder: DCM analyst*	When analyzing an incoming request for DCM information, it is first checked whether this fits within an existing DCM. If this is not the case, an estimate is made of the work load of building a new DCM.	For the request for 'deep-wound infection' it is seen that this does not fit within an existing DCM and therefore a work load of a few weeks is given. This is done because a new DCM generally takes up more time than adding an attribute or value.
 *Stakeholder: DCM analyst*	In case a DCM information request does fit within a new DCM, it is checked whether it needs a new attribute or not. If so, it is determined what attribute it is to be added.	Not applicable to our example, since it does not fit in within an existing DCM.
 *Stakeholder: DCM analyst*	It is then checked whether a new value needs to be added. If so, it is determined what value this may be and subsequently, the work load will be estimated for this DCM information request.	Not applicable to our example, since it does not fit in within an existing DCM.
 *Stakeholder: DCM analyst*	For the nested activity 'Propose candidate DCM' it is first checked whether a new DCM needs to be created. If so, existing DCM libraries are consulted.	In our example a new DCM needs to be created and therefore existing libraries (PSI, GenOGeg, eOverdracht, CEM, CDE and OpenEHR) are checked in what DCM 'deep-wound infection' could be placed in.
 *Stakeholder: DCM analyst*	From the consultation of existing DCM libraries, a choice is made what the root concept of the DCM will be. Additionally, the attributes are appointed.	The root concept of the DCM is chosen to be 'Infections'. Some attributes are 'Start date', 'End date' and 'Sort of infection'.
 *Stakeholder: DCM analyst*	In case not a new DCM is needed, it is checked whether a new attribute is needed. If so, existing libraries are consulted.	Not applicable to our example, since it does not fit in within an existing DCM.

BPMN model	General description	Specified on example
 *Stakeholder: DCM analyst*	The DCM analyst needs to determine per proposed attribute what data type it concerns. Moreover, he determines the cardinality per attribute.	In the case of attribute 'Start date' and 'End date', a date, i.e. a number (DD-MM-YYYY) is needed to fill in. This could be done with the help of a calendar. The cardinality of a start/end date with regard to an infection is 0…*
 *Stakeholder: DCM analyst*	It is then time to propose value sets for the given attributes, whereby at the same time possible mappings are proposed for these on the data warehouse. This is done, because proposed values need to be able to be mapped, thus checked if they are available in the database.	Possible values for our example under the attribute 'sort of infection' are 'upper-skin infection', 'deep-wound infection' or 'internal infection'. It is checked that these values are able to be mapped. When this is done, we have a proposed candidate DCM with root concept 'Infections'.
 *Stakeholder: Data governance architect*	The governance architect determines per attribute what value set will be used. He thereby tries to eliminate open spaces as much as possible, i.e. reduce the possibilities to answer 'other' within a value set. Furthermore he determines which code system to be applied.	From the proposed value sets, the governance architect discusses with the orthopedic surgeon that 'internal infection' does not fit within this DCM, so should not be incorporated. The code system that is going to be applied for 'sort of infection' looks for example like OID 2.16.840.1.113883.2.43.11.60.40.2.7.4.1. (this is actually the code system for 'sort of drugs', but this is for illustrative purposes)
 *Stakeholder: Data governance architect*	Then, the governance architect determines which values from that code systems are allowed per attribute. Additionally, he determines how the values from the data warehouse must be mapped on the values of the code system.	For 'deep-wound infection' an allowable value from the code system looks for example like SNOWMED CT:387341002 (This is actually the code value for 'heroin', but this is again to illustrate). Moreover, the possible mappings for this value are determined in collaboration with the database developer who is responsible concerning this piece of data.

BPMN model	General description	Specified on example
 *Stakeholder: DCM analyst*	To make a final DCM, the DCM analyst has as input an internally published candidate DCM and sends this to an expert regarding the domain of the DCM.	With respect to knee replacements, an orthopedic surgeon can be regarded as the domain expert, since he is the one performing this kind of operations.
 *Stakeholder: Domain expert*	A domain expert receives the published candidate DCM and is responsible for both adding clinical content to the DCM and reviewing the information model of the DCM. This is done simultaneously, because during the review, it is seen what essential clinical content is needed to create unambiguity in the DCM.	The orthopedic surgeon checks on the information model and adds necessary clinical content. This is more specified in the sub-processes of this nested activity on next page(s).
 *Stakeholder: Domain expert*	Depending from the outcome of the review of the information model of the DCM, it is either sent back to the DCM analyst for a candidate DCM modification or gets approval and together with the clinical content sent back to the DCM analyst for further actions.	It is assumed that our information model of 'Infection' is complete and correct. Therefore it is sent together with the clinical content to the DCM analyst by the orthopedic surgeon.
 *Stakeholders: Domain expert, DCM analyst*	The DCM analyst receives the completed and validated DCM back from the domain expert and can finalize it. This means that he checks whether the domain expert has put all clinical content under the correct chapter and adds metadata. The DCM analyst then sends it to the domain expert for a final check.	The DCM analyst adds his name, mr. X, to the final DCM, updates it to version 1.1, since it is the first version of 'Infection' and adds a disclaimer. He then sends its back to the orthopedic surgeon for a final check.

BPMN model	General description	Specified on example
 *Stakeholder: DCM analyst*	From the final check, the DCM is either approved or not. If not, it is sent back to the DCM analyst to adjust it correctly otherwise he/she publishes it as a definite/final DCM on a platform which is publicly accessible.	The DCM 'Infection' is completely validated and approved and therefore publicly published on Wiki.
 *Stakeholder: Domain expert*	A domain expert of the concerning DCM needs to add clinical content to the DCM by filling in the chapters 'Content' and 'Purpose' in plain text.	The orthopedic surgeon describes unambiguously what the DCM 'Infection' includes with i.a. clear definitions. On top of that he describes the purpose of DCM 'Infection'. One is of them is to answer the information request of the governance instance 'Zichtbare Zorg'.
 *Stakeholder: Domain expert*	A domain expert of the concerning DCM needs to add clinical content to the DCM by filling in the chapters 'Patient population' and 'Evidence base' in plain text.	The patient population for this DCM is everyone who got any kind of infection after a surgery. The chapter 'evidence base' describes related information of the DCM which is proven scientifically.
 *Stakeholder: Domain expert*	A domain expert of the concerning DCM needs to add clinical content to the DCM by filling in the chapters 'Instruction' and 'Interpretation' in plain text.	The orthopedic surgeon describes unambiguously the instruction of for example diagnosing every sort of infection. Moreover, he describes in plain text how the DCM needs to be interpreted.
 *Stakeholder: Domain expert*	A domain expert of the concerning DCM needs to add clinical content to the DCM by filling in the chapter 'Care Process' in plain text.	At last, the orthopedic surgeon describes how the care process looks like regarding infections and thereby completes the DCM with clinical content.

BPMN model



Stakeholder: DCM analyst, Database developer

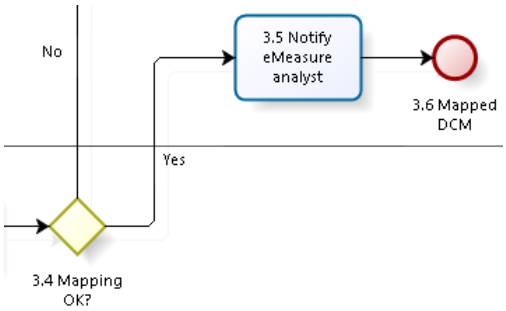
General description

For a DCM to be able to generate data, it needs to be mapped first onto the data warehouse. This is performed by the DCM analyst and needs to be reviewed by the database developer who is responsible for that part of the data warehouse.

Specified on example

The DCM analyst maps the DCM 'Infection' on the necessary fields regarding infections to the TBGM of the LTHN.

It is then evaluated by the database developer of the TBGM whether this is done correctly.



Stakeholders: DCM analyst, Database developer

After approval of the mapping, the eMeasure analyst is notified by the DCM analyst that the DCM can be used for data generation. If the mapping is not OK, the DCM analyst receives feedback of the database developer on how to map correctly.

It is assumed that the mapping of the DCM 'Infection' has been executed correctly. The DCM analyst then notifies the eMeasure analyst who is responsible for answering the information request of the government's instance 'Zichtbare Zorg'.