DSML4PTM

A Domain-Specific Modelling Language for Patient Transferal Management

Emanuele Laurenzi[1,2,3], Knut Hinkelmann[1,3], Ulrich Reimer[2], Alta van der Merwe[3], Pascal Sibold[1] and Rainer Endl[2]

[1]*FHNW University of Applied Sciences and Arts Northwestern Switzerland, Von-Roll Srasse, 10, 4600 Olten, Switzerland*
[2]*IPM-FHSG, Institute of Information and Process Management, University of Applied Sciences St. Gallen, Rosenbergstrasse, 59, 9001 St. Gallen, Switzerland*
[3]*Department of Informatics, University of Pretoria, Lynnwood Rd, 0083 Pretoria, South Africa*

Keywords: DSML4PTM, Domain-specific Modelling Language, Domain-specific Conceptual Modelling, Metamodeling, Model-driven Approach, Modelling Languages Extension.

Abstract: This paper presents a domain-specific modelling language for patient transferal management (DSML4PTM). To foster reusability within the modelling community, existing modelling languages were taken into account as far as possible and then extended as was needed by the application domain. The language was developed through iteration following the design science research methodology. For requirements elicitation purposes domain expertise and healthcare standards were taken into account. The new modelling language was evaluated first with respect to the elicited requirements and then through the creation of two models reflecting a reference process and an application scenario. Next, an evaluation on the perceived usefulness and cognitive effort of the language was performed using a focus group with modelling and domain experts.

1 INTRODUCTION

Recent statistical findings released by Eurostat (2017) revealed that healthcare is still a main expenditure in developed countries. Germany has the lead among EU members with a current healthcare expenditure equivalent of 11.0% gross domestic product (GDP), while Switzerland has an even higher expenditure (11.4%). These percentages have the tendency to even increase according to The World Bank (2014) and governments are reacting by imposing pressure on healthcare providers, especially on hospitals to lower cost. However, hospitals should still provide a high quality of service (Lenz et al. 2012), which require processes and activities to be as optimal as possible. This, however, results in a huge challenge due to the complexity of the domain. In fact, many structured and ad-hoc processes that involve a broad range of crucial decisions typically can take place across organizations and among actors with different expertise. One process that reflects such a complex environment is the so-called transferal management process, which is also called *transitional care* or *hospital discharge management/planning*. Parry et al. (2008) defined transferal management as "a set of actions designed to ensure the coordination and continuity of care received by patients as the transfer between different locations or levels of care". This set of actions, also called administrative pathways, includes medical information and excludes the treatment of the patient, which rather refer to clinical pathways (Lenz and Reichert 2007).

The Patient Radar Project (Reimer and Laurenzi 2014) addresses the above-mentioned issues by enabling intersectoral collaboration between acute hospitals and rehabilitation clinics, where (a) rehabilitative expertise is brought early into the acute somatic treatment loop, and (b) demand for rehabilitation treatment is considered as early as possible. Such a collaboration takes place within the complex settings of the transferal management domain, where many domain experts are involved, i.e. from acute hospitals, rehabilitation clinics, and finally health insurances for cost reimbursements.

In our work, the main idea is to adopt a model-driven approach that provides all the relevant concepts and decision types of the transferal management domain in a form that is familiar to the

Laurenzi, E., Hinkelmann, K., Reimer, U., Merwe, A., Sibold, P. and Endl, R.
DSML4PTM - A Domain-Specific Modelling Language for Patient Transferal Management.
DOI: 10.5220/0006388505200531
In *Proceedings of the 19th International Conference on Enterprise Information Systems (ICEIS 2017) - Volume 3*, pages 520-531
ISBN: 978-989-758-249-3

domain experts. The ultimate goal is to enable domain experts designing and/or adapting models in a meaningful and less error-prone way, and therefore to help produce quality domain models.

For this, a domain-specific modelling language for patient transferal management (DSML4PTM) is proposed.

The paper is organized as follows: Section 2 provides the motivation for using DSMLs. Section 3 introduces the most relevant DSMLs for this work. Next, the adopted methodology is described in section 4. In section 5 we will describe the new metamodel as well as the procedure we followed. Section 6 presents the implementation of the new metamodel in the ADOxx metamodeling toolkit. Finally, evaluation and conclusion are described in section 7 and 8, respectively.

2 MOTIVATION FOR USING DOMAIN-SPECIFIC MODELLING LANGUAGES

In order to ensure a common understanding on the term "modelling language" we refer to the framework developed by Karagiannis and Kühn (2002) - A *modelling language* is defined by *notation*, *syntax* and *semantics*. The notation specifies the visual representation of the modelling language, while the semantics define the meaning of the syntactic element by establishing a mapping to a semantic schema. The *modelling procedure* determines how to apply a modelling language to create models.

Models are defined in the metamodel, which is created by a metamodeling language. The metamodel itself is defined in a metametamodel (see Fig. 1). According to different authors (Karagiannis and Kühn 2002, Atkinson and Kuhne 2003, Laarman and Kurtev 2010) these 3 levels are sufficient. The Object Management Group (OMG 2015), for instance, proposed metamodeling framework (MOF) for model-driven engineering (OMG 2015) with exactly this number of levels.

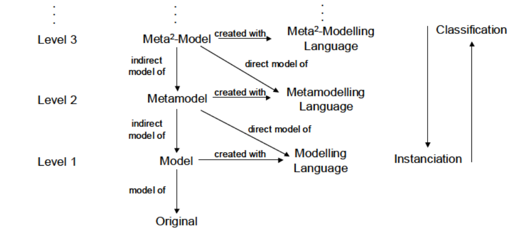

Figure 1: Metamodelling Hierarchy (Strahringer 1996).

There is a distinction between domain-specific modelling languages (DSML) and general-purpose modelling languages (GPML), An example for a GPML are UML class diagrams. A GPML provides the user with a high degree of freedom, and thus might lead to inconsistency and representations that do not make sense. A way to overcome these problems is by extending the semantics of the metamodel (Pérez and Porres 2013, Lodderstedt et al. 2002, Felfernig et al. 2000) by adding constraints that restrict the varieties of models. For this purpose, OMG extended the UML metamodel introducing the Object Constraint Language standard (OMG 2014). This approach, however, increases the complexity of the modellers' task as they either have then to know all the constraints, or they are hindered by the system to make certain modifications to a model when they would violate a constraint.

Domain-specific modelling languages (DSMLs), shift this complexity to the metamodel level, where constraints are introduced in order to ease the modelling on level 1 (see Fig. 1) (Fowler 2011, Frank 2010, Gray et al. 2008, Mernik et al. 2005, van Deursen et al. 2000). For this, domain-specific modelling languages (DSMLs) promise to enable domain experts designing models in a meaningful and less error-prone way, and therefore support producing quality models (Kelly and Tolvanen 2008). Moreover, allowing domain experts dealing directly with language constructs they are familiar with, leads to significant impact on their learning curve. It was already observed by Hudak and Paul (1996) that, domain experts can quickly learn the language and its applicability improves. In other words, DMSLs bring the benefit of high understanding of models (hence, the underlying reality) among domain experts, fostering not just productivity in design-time, but also the optimization phase, where paint points are rapidly identified and actions can be taken accordingly.

A common approach to develop a DSML consists of several iterative phases until a version of the language is delivered that is mature enough. The order of these phases is typically as follows: (1) capturing domain requirements, (2) defining concrete syntax, abstract syntax and related semantics (metamodel) (3) testing the language with end users. The latter can lead to further iterations in case some requirements of the language are not fulfilled. The main complexity resides in phase "2", which is according to Cho et al. (2012) is crucial even for language development experts.

In the following section, we present some of the most relevant DSMLs in healthcare.

3 DSMLS IN HEALTHCARE

Shenvi et al. (2007) created a DSML to model an electronic Patient Care Report (ePCRs). The metamodel includes domain-specific modelling constructs such as patient demographics, vital signs and medications. Each of these constructs has several attributes that define the visualization and behavior of that element. Additionally, connections between constructs are defined.

Mathe et al. (2009) have designed a visual DSML (Clinical Process Management Language) for capturing treatment protocols. They argued that at that time there existed no widely accepted visual language for capturing treatment protocols, and generic software modelling languages, such as UML, are not designed to capture medical knowledge.

Burwitz, et al. (2013) have evaluated modelling languages for their suitability to model clinical pathways. They started with defining domain-specific requirements and found out, that existing languages such as BPMN do not fulfil all of them. The available languages mainly fail to represent variable flows and evidence-based decisions. The authors therefore decided to create their own DSML (called CP-Mod), based on the Clinical Algorithm. CP-Mod needed to be extended to be able to model complex health care processes and to reduce the existing deficits by addressing the requirements that are only partly or not at all met.

Heß et al. (2015) have done a similar work, which can be seen as an advancement of the work from Burwitz, et al. (2013). Also Heß, et al. (2015) came up with an extended list of domain-specific requirements. They recognized that no language fully covers these extended requirements to model clinical pathways. They argue: "to realize the potential benefits of CPs (clinical pathways), a comprehensive modelling method accounting for peculiarities of hospitals' action system and IS is required. The core of such a method should be a Domain-Specific Modelling Language for CPs". They further mention that widely accepted business process modelling techniques to model CPs lack in fostering communication between stakeholders and in quality support from tools. Thus, they decided to extend the metamodel of the language MEMO OrgML to create a DSML for the modelling of clinical pathways. The newly created language DSML4CPs reconstructs the professional terminology from the medical domain with a special focus on oncology.

Braun, et al. (2015) have also noticed the need for domain-specific concepts to model clinical pathways. However, they propose to use BPMN for the extension as it is widely used for modelling business processes. In their work, they use the language CP-Mod to derive requirements for the extension of BPMN and to create the language BPMN4CP. One advantage of the solution is that several perspectives on a clinical pathway, such as resource, process or document perspective, are possible to be modelled. They have revised their BPMN4CP approach based on its practical application. This led to an evolution of BPMN4CP from version 1.0 to version 2.0 to cover additional business requirements (Braun et al. 2016).

Neumann et al. (2016) also developed a domain-specific extension of the BPMN modelling language. Their objective was to model and execute interoperative surgical workflow in the integrated operating rooms.

It is becoming a common practice to support experts by utilizing modelling languages that cover the requirements of the underlying domain (Neumann et al. 2016, Braun, Burwitz, et al. 2015, Braun, Schlieter, et al. 2015, Burwitz et al. 2013). However, to the best of our knowledge, so far research on DSMLs in healthcare mainly focused on clinical pathways (CPs). On the contrary of administrative pathways, CPs refer to the treatment activities of patients. Therefore, we can state that there is a gap in the literature to provide a domain-specific modelling language for the patient transferal management.

4 RESEARCH METHODOLOGY

In this work, we developed the DSML by adopting the design science research (DSR) approach of Vaishnavi and Kuechler (2004):

1. Awareness of the problem: In this phase, we elicited requirements from the transferal management domain. For this, we could mainly rely on the activities performed in the research project Patient Radar. Namely, they carried out interviews and workshops with modelling and domain experts, from which a reference model and an application scenario were created. Additionally, health-related standards and documentation were taken into account together with literature review.

   Next, a categorization was made to differentiate procedural from declarative information as it is recommended by von Halle and Goldberg (2010). That is, we distinguished between process logic (i.e. which activities take place – including prescribed flow and more ad hoc activities) and business logic (i.e. how to make decisions – including structured and unstructured logic). Additional declarative categorizations concerning data representation, documentation and organization management were included. Table 1 shows one of the 50 requirements.

Table 1: Excerption from list of requirements.

#	Requirement	Description	Category
R3.1.4	The DSML should accommodate constructs to model decision criteria according to the DefReha© standard.	Entry, exit, inclusion and exclusion criteria for the different rehabilitation types should be defined according to the DefReha© standard.	Business Logic (Structured)

2. Suggestion: In this phase, we took into account all the requirements elicited in the previous phase and suggested a new metamodel accordingly.
3. Development: In this phase, we implemented the metamodel as well as the graphical notation for the concrete syntax in the ADOxx Development Toolkit.
4. Evaluation: In this phase, we first evaluated the new language with respect to the elicited requirements. Then, we investigated its applicability by modelling both the reference model and application scenario created in the awareness of problem. Finally, a focus group session was performed to evaluate the perceived usefulness and cognitive effort of the language.

All phases were used to continuously refine the artifact.

5 A METAMODEL FOR DSML4PTM

The design approach for our DSML is based on the two principles proposed by Karagiannis et al. (2016):

- Considering already existing standard modelling languages, with related applications and lessons learned. Emmenegger et al. (2016), for example, combined several modelling languages to enhance workplace learning.
- Specialization of language constructs according to requirements elicited from a specific domain. Hinkelmann et al. (2016), for example, extended BPMN to allow specifying both business requirements for the cloud service discovery and cloud service descriptions.

The procedure used to define the new language was the following:

According to the gathered requirements we (a) identified the most suitable existing modelling languages; (b) removed the unneeded modelling elements (as also Silver (2011) suggests) from the selected languages; (c) added the domain-specific elements; (d) integrated the language constructs that belong to different modelling languages, and finally (e) added constraints among the remaining language constructs. Steps (b) and (c) can also be inverted.

These steps were embedded into a two-tier approach that refers to the metamodeling hierarchy introduced in Fig. 1, see also (Reimer and Laurenzi 2014). (a), (b), (c) and (d) are all part of the abstract syntax, which resides in level 2 of the metamodel stack (see Fig. 2). They all contain semantics. The identified metamodels in (a), for example, already contain their own semantics. Moreover, while for (b) semantics is removed, it is added for (c). Integrating model elements among each other also imply to additional semantics, (see (d) in Fig.2). Additional semantics are added in terms of new relations among metamodel elements, (see (e) in Fig. 2). Modelling elements added in (c) are then represented in the form of concrete syntax together with the remaining metamodel elements (see level 1 of Fig. 2). In level 1, new models can be created and existing ones can be adapted to reflect different scenarios.

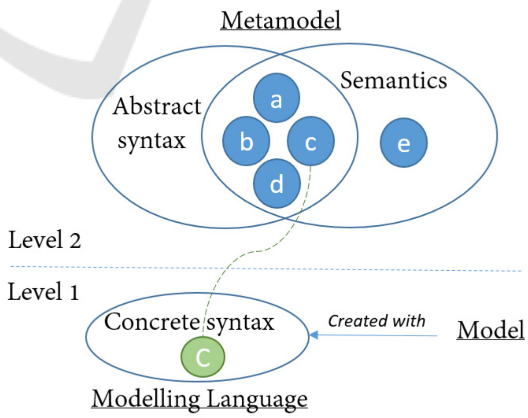

Figure 2: Suggested approach.

The two-tier approach allows the following:

- On Level 1, administrative pathways from acute hospitals and rehabilitation clinics are combined into a coherent discharging

processes. Moreover, new processes can be accommodated or the existing ones adapted as the discharging process that fits one acute hospital might differ for another clinic.
- On Level 2, the new metamodel can be extended further to cover new domain requirements, e.g. transferring patients from acute hospitals to nurse facilities.

About 100 concepts and 300 attributes were added in the metamodel. Due to space restrictions, we do not list all the elements, but we rather motivate the selection for each metamodel with respect to the most relevant requirements. For each of the metamodels, we then describe the rest of our procedure by mentioning some of the extended elements as well as the integration among metamodel elements.

5.1 Process Modeling

Requirements for structured and unstructured processes
It is necessary to combine predefined, clearly structured process parts with more ad hoc process steps, as some activities and conditions are known in advance while the execution of others depends on human judgment or external events. For instance, before performing the patient's disposition in the rehabilitation clinic, the transferal manager should release all the necessary data and documents related to the patient (i.e. sequence flow). Conversely, the responsible physician from the rehabilitation clinic might discuss with the main physician from the acute hospital about the patient's therapy. The activity execution of this activity is up to the responsible physician rather than dictated by a sequence flow.

BPMN
BPMN 2.0 (OMG 2011) is the most widespread notation for modelling business processes. Therefore it was selected to model activities and conditions, which are known in advance and their flow can thus be modelled. However, we removed unnecessary BPMN elements, and extended it with new elements to satisfy our list of requirements (see concepts in light blue bubbles in Fig. 3).

The requirement for the process progress was addressed by adding a new concept *"Status"*. This includes the percentage attribute that reflects the actual state of the process (see the related concrete syntax in the point 6 of figure 8).

CMMN
To model those activities and conditions that cannot be embedded in the process flow, we chose CMMN (OMG 2016a) the OMG standard for case management modelling. Again, we removed the unneeded elements as well as extended new ones. "Sentry" and "Entry Criterion" were kept, while for "Discretionary Task" specialisation like *Update Disposition* and *Perform Rehab Conference* were introduced.

Insights
The integration with the BPMN elements took place by connecting the sentry to the task. According to our proposed semantics, a task can be performed either as a subsequent activity as part of a flow, or as soon as

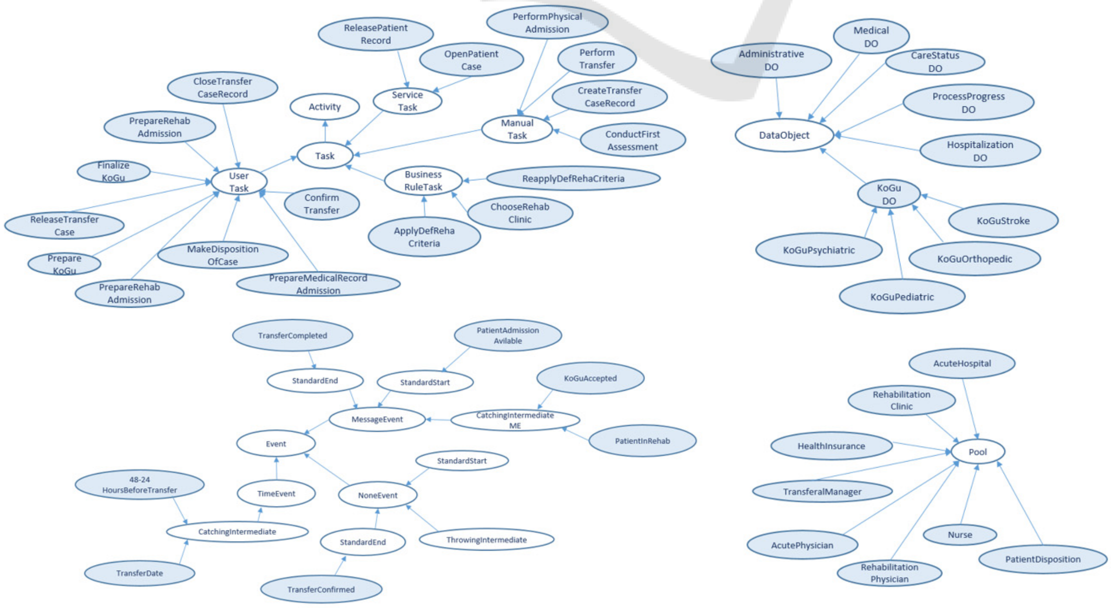

Figure 3: Extended elements of BPMN.

the sentry evaluates to true. Moreover, a task can have one or more sentries. The latter expresses the "OR" condition. The same applies also in presence of an input flow and at least one sentry.

The discretionary task concept is a subclass of the manual task (see bubbles with blue outline in Fig. 4). At run-time, discretionary tasks that are involved in the sequence flow are skipped if none of the attached sentry evaluates to true. For example, the *PerformRehabConference* is a discretionary task that can be executed by the rehab if the patient case is complex or simply if he/she wants to discuss the case with the physician in the acute hospital.

5.2 Control Element Model

Requirements for Decisions on the Process Flow
Complex decisions are made along the discharging process. For example, the acute hospital decides on whether to start performing the admission for the incoming patient (e.g. preparing all the needed resources). This is only possible after the transferal date has been agreed upon between the acute hospital and rehabilitation clinic and after the cost reimbursement form has been sent to the health insurance. Another example is on decisions that need to be taken if the patient's situation worsens.

Control Element Model

As it is already claimed by Hinkelmann (2016), sentries can be specified in the so-called Control Element Model in order to enable reuse of conditions and events. The metamodel elements that are taken into account from the Control Element Model are "On-Condition" and "If-Condition". Figure 4 shows the extended elements in the light orange bubbles as well as their integration with the sentry element of the CMMN.

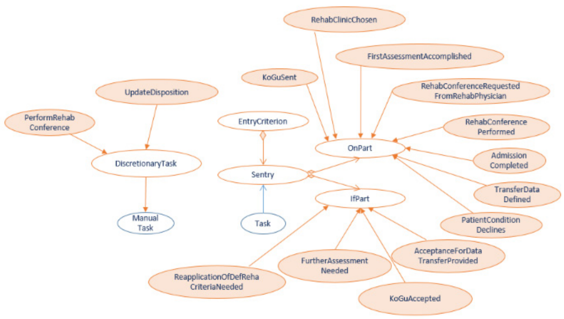

Figure 4: Some extended elements from CMMN, Control Element Model and their integration.

5.3 Decision Modeling

Requirements for Decisions on the Activity Level
The decision logic for complex decisions along the discharging process has to be modelled on the activity level. For example, the application of the right discharging criteria permits to derive the most suitable rehabilitation type and interface (e.g. from acute hospital to inpatient neurological rehabilitation, rather than to a rehabilitation with compulsory medical monitoring). In Switzerland, discharging criteria are specified in the DefReha© standard issued by the organization H+ Swiss Hospital (H+ 2016).

DMN
In order to model decision logic we selected the DMN standard (OMG 2016b). The metamodel concerning the Decision Requirements and the Decision Logic were reused and integrated into the metamodel of the new DSML. The analysis of the DefReha© document has shown, that the decision to determine the rehabilitation category and relevant inclusion and exclusion criteria are of central importance for the domain in Switzerland. Figure 5, shows the extended elements in light green bubbles (due to space limitation, we could not list all of them). The integration took place via the decision element, which refers to the business rule task and the discretionary task.

Additional constraints were added among the extended elements. In section 6 (*arrow 5* in Figure 8) we describe an example on how the DMN elements are related with each other.

5.4 Documents and Knowledge Model

Requirements for Structured Documents
The modelling has to comply with healthcare standards like the International Classification of Functioning, Disability and Health (ICF) standard (World Health Organization 2016).

Representing the relevant process documents is also a need: The cost reimbursement form (abbreviated as KoGu) is the most important document in the transferal management process. In case it is definitely rejected, the discharging process most likely comes to an end due to the lack of financial support.

Another important document is the hospitalization form, which contains all the needed information for the transferal manager to start the process. The process progress along the patient hospitalization should provide visibility to crucial events such as case and patient accepted by the clinic, first assessment performed, discharging date agreed, etc. Then, the

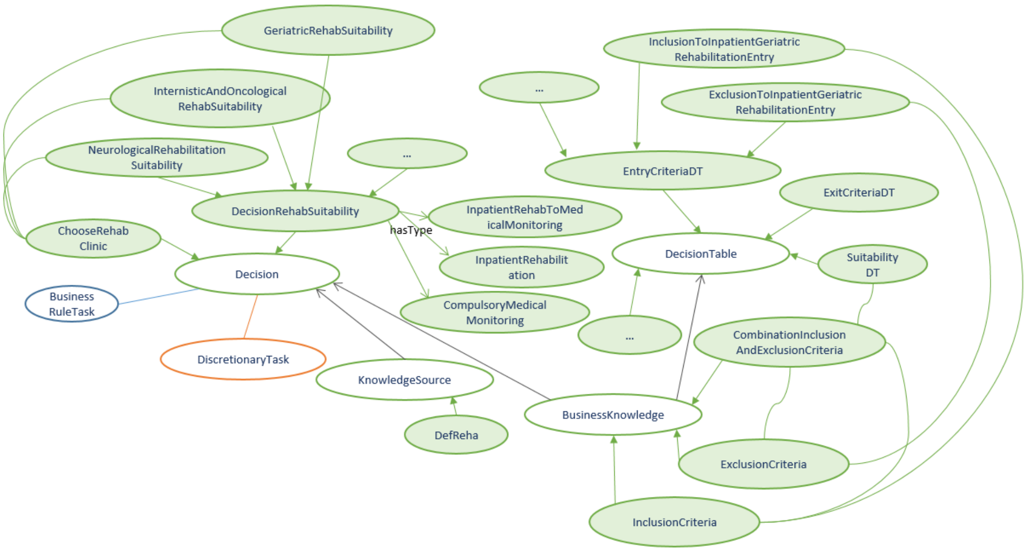

Figure 5: Some extended elements from DMN, their integration and additional constraints.

medication list form that has to be updated after the transferal is confirmed.

Finally, each document belongs to a category. For instance, the hospitalization document belongs to the administrative data, the medication list belongs to the medical data, while the assistance list belongs to the care status data.

Requirements for Document Representation

There is a need to represent the status and versions of document. For example, the KoGu document should include the status (i.e. ready, sent, rejected or accepted), while the ICF document should include both the trend and the status of each category.

Documents and Knowledge Model

Complying with many standards as well as accommodating many relevant documents were addressed by selecting the Documents and Knowledge Model adopted in the European project LearnPAd (De Angelis et al. 2016). This provides a way to structure documents and knowledge representations. For example, we introduced the new concept "KoGu data object" as a specialized data object. The required status of the KoGu data object was modelled by means of attributes (Figure 6 shows the concrete syntax of the KoGu data object with related status as well as the hospitalization document with related version on top of the icon).

Constraints for the defining the structure of data and documents were also added. Due to the very large collection of data and documents, we grouped them into medical data, administrative data, care status data and process progress. Each group contains several

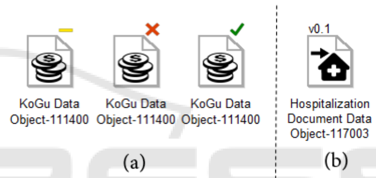

Figure 6: Example of status and version on extended data objects.

data and documents (e.g. see assistance data and special medication data connected with the care status in Fig. 7).

Additional constraints addressed the need of indicating the status of data/documents. For instance, the metamodel contains a connection between the ICF standard to the ICF qualifier status, and between the KoGu document and its status (see Fig. 7). The same applies on other four elements, i.e. process status, physical transfer status, medication list status, and acceptance status. These "Status" elements are determined based on the progress of the data collection and are used to aggregate the overall "Status" in the DSML4PTM model.

Bubbles with light yellow in Fig. 7 depict some of the specialized elements, while connections to other colored bubbles show their integration with other metamodel elements, i.e. from BPMN (in light blue) and from DMN (in light green).

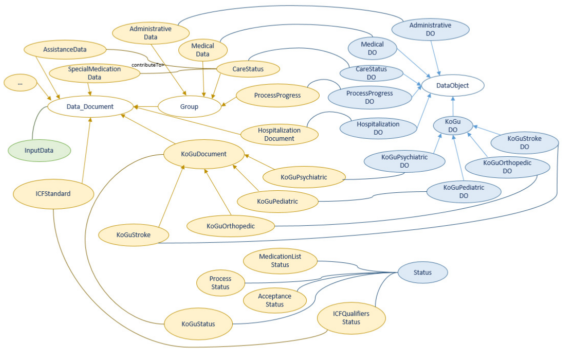

Figure 7: Some extended elements from Document and Knowledge Model, their integration and additional constraints.

5.5 Organizational Model

Requirements for a structure on roles and care units
Making the structure of the organization explicit in a model supports the domain expert to identify quickly and easily who is performing what role in which care unit or rehab clinic.

Organization Model
The Organizational Model as proposed in Emmenegger et al. (2016) provides constructs for assigning personnel to roles, which in turns are assigned to organization units. This structure ensures that a user can easily browse through an organization and find a suitable person or business unit.

We extended the organizational metamodel as follows. The role element was specialized as: *Administrative Staff*, *Nurse*, *Acute Physician*, *Rehabilitation Physician*, *Patient*, *Patient Disposition* and *Transferal Manager*. Next, the organizational unit was specialized as: *Acute Hospital*, *Rehabilitation Clinic*, *Care Unit*, *Non-intensive Care Unit*, *Intensive Care Unit*, *Emergency Room*, *Site of Care* and *Health Insurance*.
The organizational metamodel was then integrated with the process model as follows:

- Extended organizational unit elements are associated with the extended pool elements from BPMN.
- Extended role elements are associated with the extended lane elements from BPMN. For example, the role transferal manger (subclass of role) is associated with the transferal manager (subclass of Lane).

6 IMPLEMENTATION

The new metamodel was implemented in ADOxx Development Toolkit (https://www.adoxx.org).
Figure 8, shows part of the application scenario modelled in DSML4PTM. The figure also shows how model elements that belong to the above described metamodels are related to each other.

Arrow 1 shows the connection of the acute hospital from the process to the organizational unit.

Arrow 2 shows the connection of the data object "Hospitalization" with the related document, which is modelled in the Document and Knowledge Model.

Arrow 3 shows the notebook with all data values of the KoGu document.

Arrow 4 shows the connection going from the sentry attached to the discretionary task "Re-apply DefReha Criteria" to the related Control Element Model. A Sentry contains optional events (ON part) and conditions (IF part). According to the semantics borrowed from CMMN, the execution of the task is possible if both ON-part and IF-part evaluate to true.

Arrow 5 shows the connection of the business rule task "Apply DefReha Criteria" with the decision construct "Decide on Rehabilitation Suitability" modelled in the Decision Requirements Diagram. The latter construct takes the input "patient data" and "medical information" (both modelled in the Document and Knowledge model). It refers to the knowledge source "DefReha©", which is an external PDF. Additionally, rules for selecting the rehabilitation type are expressed in decision tables. Hence, all the sub-decisions that reflect each rehabilitation type are modelled and used as input for the "Decide on Rehabilitation Suitability" concept. For example, the neurological rehabilitation type has three interfaces: Inpatient Rehabilitation, Compulsory Medical Monitoring and from Medical Monitoring to Inpatient Rehabiliation. These are modelled as further decision concepts that in the metamodel are connected with the "Neurological Rehabilitation Suitability" concept.

Arrow 6 depicts the connection from the process status element to all those attributes for which the status is determined, e.g. data released, first assessment done, Rehab conference done, transfer date, KoGu ready, etc.

Additionally, the integration of CMMN and BPMN allows specifying the actor who would perform a discretionary task by placing the task on the appropriate lane.

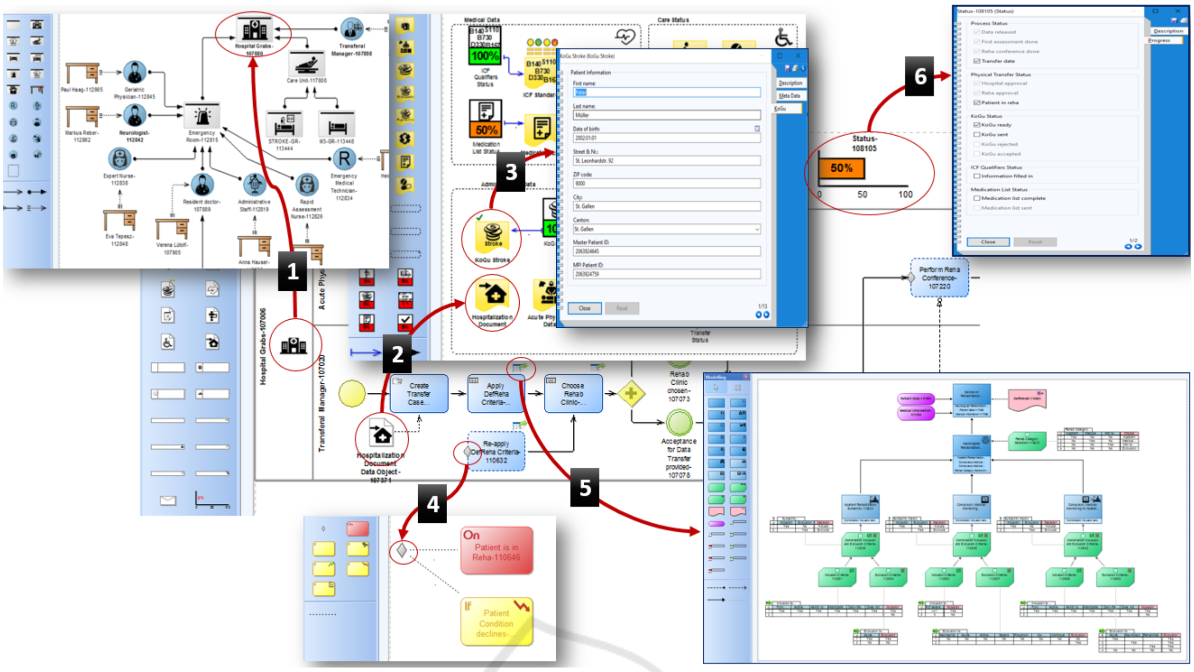

Figure 8: Part of the reference model implemented in ADOxx.

7 EVALUATION

The new modelling language was evaluated on its completeness with respect to the elicited domain requirements. DSML4PTM was implemented in ADOxx and both a reference process and an application scenario were successfully modelled after the implementation. These models were then used in a two-hour focus group session with four modelling experts and one domain expert for an evaluation on the perceived usefulness of the language and its cognitive effort. In the first hour, the new modelling language was introduced to the participants and then a walk-through session explained how the new language can be applied. In the next hour, the participants had a hands-on session and they were asked to extend the reference process with two new scenarios:

1. DefReha criteria should be re-applied in case the KoGu document is rejected.
2. KoGu document needs to be revised as some information is missing.

Finally, a questionnaire was provided to perform a qualitative evaluation. The questionnaire was developed based on the guidelines proposed by Tullis and Albert (2013). It was conducted on individual basis and within a time frame of at least half an hour. The questionnaire contained 5 sections, i.e. four questions on general background; eleven questions on modeling background; three open questions on the perceived usefulness of the new modelling language; five open questions on the cognitive effort of the new modelling language; and three open questions as a feedback for future improvements.

In the following, we summarize the outcome of the questionnaire with respect to the perceived usefulness and cognitive effort.

Perceived usefulness: All participants agreed on the high potential of the DSML in fostering communication and collaboration among domain experts. Modelling experts stated that the new language simplifies the modelling process, which helps improve model quality. They also emphasized that the language can be seen as a basis for the integration of different information systems (e.g. health information systems, patient administration systems, health records) and automated verification.

Cognitive effort: All participants agreed on the ease-of-use of the new language, for which a metric is the amount of time that is required to learn the usage of a language. All the participants stated that the usage of the language can be learned within a short period of time. This was backed by the fact that participants could propose meaningful models for the two new scenarios within less than half an hour. Of course, to apply the language in real settings would need more time to get acquainted with it. Further metrics we looked at are the appropriate abstraction of language elements and their default values, the graphical notation of the concrete syntax, and the support at design time in creating meaningful models. The latter was mentioned as one of the main strength as during the hands-on session participants perceived they could not connect model elements with each other randomly. In line with this, modelling experts mentioned the reduction of modelling mistakes as a strength of the new language. One agreed upon drawback of the language is the difficulty to understand when to use a sentry rather than a gateway. We noticed that during the hands-on session, those with a strong BPMN background tended to use gateways only. One suggestion was to encourage the usage of sentries as a best practice while using gateways only if really needed, e.g. in case the visualization of the flow is fundamental

The evaluation was also done quantitatively by comparing the reference model created in DSML4PTM with the equivalent one created in BPMN. The latter contained approximately 15% more elements. In fact, in complex scenarios (e.g. when KoGu is rejected) the specification of most of the events and conditions that would be modelled with gateways can be replaced by sentries (that is also valid for tasks that are embedded in a flow).

8 CONCLUSIONS

Based on the elicited requirements from the awareness phase, a domain-specific modelling language for transferal management was developed. The development procedure is based on principles that foster the reusability within the modelling community. Hence, several metamodels were considered to cover as far as possible the application domain, i.e. BPMN, DMN, CMMN, Control Elements Model, Organization Model and Documents and Knowledge Model. Next, unneeded modelling elements were removed while others were added to fulfill the requirements of the transferal management domain. Then, a metamodel integration among the remaining modelling elements was performed. Finally, semantics was mainly borrowed from the selected metamodels and new one added by means of constraints among the remaining language constructs.

Future work goes towards a methodology that provides semantically enhanced recommendations while developing a DSML. Another possible research direction goes towards the modelling support at design time. Namely, the retrieval of best practices modelling patterns according to the syntax and semantics of models. These patterns will then be proposed to the modeler for quality improvements.

REFERENCES

De Angelis, G. et al., 2016. Modeling for Learning in Public Administrations—The Learn PAd Approach. In *Domain-Specific Conceptual Modeling*. Cham: Springer International Publishing, pp. 575–594. Available at: http://link.springer.com/10.1007/978-3-319-39417-6_26 [Accessed March 11, 2017].

Atkinson, C. & Kuhne, T., 2003. Model-driven development: a metamodeling foundation. *IEEE Software*, 20(5), pp.36–41.

Braun, R. et al., 2016. BPMN4CP Revised – Extending BPMN for Multi-Perspective Modeling of Clinical Pathways. Available at: https://www.computer.org/csdl/proceedings/hicss/2016/5670/00/5670d249.pdf [Accessed March 8, 2017].

Braun, R., Burwitz, M., et al., 2015. Clinical processes from various angles - amplifying BPMN for integrated hospital management. In *2015 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE, pp. 837–845. Available at: http://ieeexplore.ieee.org/document/7359794/ [Accessed February 16, 2017].

Braun, R., Schlieter, H., et al., 2015. Extending a Business Process Modeling Language for Domain-Specific Adaptation in Healthcare. In *12th International Conference on Wirtschaftsinformatik*. Osnabrück, Germany.

Burwitz, M., Schlieter, H. & Esswein, W., 2013. Modeling Clinical Pathways - Design and Application of a Domain-Specific Modeling Language. *Wirtschaftsinformatik Proceedings 2013*. Available at: http://aisel.aisnet.org/wi2013/83 [Accessed November 20, 2015].

Cho, H., Gray, J. & Syriani, E., 2012. Creating visual Domain-Specific Modeling Languages from end-user demonstration. In *2012 4th International Workshop on Modeling in Software Engineering (MISE)*. IEEE, pp. 22–28. Available at: http://ieeexplore.ieee.org/document/6226010/ [Accessed March 9, 2017].

van Deursen, A., Klint, P. & Visser, J., 2000. Domain-specific Languages: An Annotated Bibliography. *SIGPLAN Not*, 35(6), pp.26–36. Available at: http://doi.acm.org/10.1145/352029.352035.

Emmenegger, S. et al., 2016. Workplace Learning - Providing Recommendations of Experts and Learning Resources in a Context-sensitive and Personalized Manner. In *MODELSWARD 2016, Special Session on Learning Modeling in Complex Organizations*. Rome.

Eurostat, 2017. *Healthcare expenditure statistics*, Available at: http://ec.europa.eu/eurostat/statistics-explained/index.php/Healthcare_expenditure_statistic.

Felfernig, A., Friedrich, G.E. & Jannach, D., 2000. UML as Domain Specific Language for the construction of Knowledge-Based Configuration System. *International Journal of Software Engineering and Knowledge Engineering*, 10(4), pp.449–469. Available at: http://www.worldscientific.com/doi/abs/10.1142/S0218194000000249?src=recsys&journalCode=ijseke [Accessed May 20, 2015].

Fowler, M., 2011. *Domain-specific languages*, Upper Saddle River: Addison-Wesley.

Frank, U., 2010. Outline of a method for designing domain-specific modelling languages. , (42). Available at: http://hdl.handle.net/10419/58163.

Gray, J. et al., 2008. DSLs: the good, the bad, and the ugly. In *Conference on Object Oriented Programming Systems Languages and Applications archive*. Nashville and {É}tats-Unis: ACM. Available at: http://hal.inria.fr/inria-00402566.

H+, 2016. DefReha© Version 1.01. Available at: http://www.hplus.ch/de/dienstleistungen/branchenloesungen/defrehac/ [Accessed March 11, 2017].

von Halle, B. & Goldberg, L., 2010. *The Decision Model: A Business Logic Framework Linking Business and Technology*, CRC Press Auerbach Publications.

Heß, M. et al., 2015. A domain-specific modelling language for clinical pathways in the realm of multi-perspective hospital modelling. *ECIS 2015 completed research papers*. Available at: https://bibliographie.ub.uni-due.de/servlets/DozBibEntryServlet?mode=show&id=64975&XSL.ListKey=iqs6j63u&XSL.PageNr=&lang=en [Accessed March 8, 2017].

Hinkelmann, K. et al., 2016. A Semantically-Enhanced Modelling Environment for Business Process as a Service. In *4th International Conference on Enterprise Systems ES2016*. Melbourne, Australia. Available at: https://www.researchgate.net/publication/308396155_A_Semantically-Enhanced_Modelling_Environment_for_Business_Process_as_a_Service.

Hinkelmann, K., 2016. Business Process Flexibility and Decision-Aware Modeling—The Knowledge Work Designer. In *Domain-Specific Conceptual Modeling*. Cham: Springer International Publishing, pp. 397–414. Available at: http://link.springer.com/10.1007/978-3-319-39417-6_18 [Accessed August 11, 2016].

Hudak, P. & Paul, 1996. Building domain-specific embedded languages. *ACM Computing Surveys*, 28(4es), p.196–es. Available at: http://portal.acm.org/citation.cfm?doid=242224.242477 [Accessed March 9, 2017].

Karagiannis, D. et al., 2016. Fundamental Conceptual Modeling Languages in OMiLAB. In *Domain-Specific Conceptual Modeling*. Cham: Springer International Publishing, pp. 3–30. Available at: http://link.springer.com/10.1007/978-3-319-39417-6_1 [Accessed August 11, 2016].

Karagiannis, D. & Kühn, H., 2002. Metamodelling Platforms. In *In Proceedings of the 3rd International Conference EC-Web 2002 -- Dexa 2002, Aix-en-Provence, France, 2002, LNCS 2455*. Springer-Verlag, p. 182.

Kelly, S. & Tolvanen, J.-P., 2008. *Domain-specific modeling: Enabling full code generation*, Hoboken: Wiley.

Laarman, A. & Kurtev, I., 2010. Ontological Metamodeling with Explicit Instantiation. In M. Brand, D. Gašević, & J. Gray, eds. *Software Language Engineering*. Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 174–183. Available at: http://dx.doi.org/10.1007/978-3-642-12107-4_14.

Lenz, R., Peleg, M. & Reichert, M., 2012. Healthcare Process Support: Achievements, Challenges, Current Research. Available at: http://dbis.eprints.uni-ulm.de/784/ [Accessed March 6, 2017].

Lenz, R. & Reichert, M., 2007. IT support for healthcare processes – premises, challenges, perspectives. *Data & Knowledge Engineering*, 61(1), pp.39–58. Available at: http://linkinghub.elsevier.com/retrieve/pii/S0169023X06000784 [Accessed March 6, 2017].

Lodderstedt, T., Basin, D. & Doser, J., 2002. SecureUML: A UML-Based Modeling Language for Model-Driven Security. In Springer, pp. 426–441.

Mathe, J.L. et al., 2009. A Model-Integrated, Guideline-Driven, Clinical Decision-Support System. *IEEE Software*, 26(4), pp.54–61. Available at: http://ieeexplore.ieee.org/document/5076459/ [Accessed March 8, 2017].

Mernik, M., Heering, J. & Sloane, A.M., 2005. When and how to develop domain-specific languages. *ACM Computing Surveys*, 37(4), pp.316–344.

Neumann, J. et al., 2016. BPMNSIX – A BPMN 2.0 Surgical Intervention Extension: Concept and Design of a BPMN Extension for Intraoperative Workflow Modeling and Execution in the Integrated Operating Room. In *7th Workshop on Modeling and Monitoring of Computer Assisted Interventions (M2CAI) - 19th International Conference on Medical Image Computing and Computer Assisted Interventions (MICCAI 2016), At Athens, Greece*.

OMG, 2011. *Business Process Model and Notation (BPMN) Version 2.0*, Needham, MA: Object Management Group OMG.

OMG, 2016a. Case Management Model and Notation (CMMN V 1.1). Available at: http://www.omg.org/spec/CMMN/1.1/PDF/.

OMG, 2016b. Decision Model and Notation. Available at: http://www.omg.org/spec/DMN/1.1/PDF/.

OMG, 2015. *Meta Object Facility (MOF) Core Specification Version*, Object Management Group. Available at: http://www.omg.org/spec/MOF/2.5/.

OMG, 2014. Object Constraint Language. Available at: http://www.omg.org/spec/OCL/2.4 [Accessed May 27, 2015].

Parry, C. et al., 2008. Assessing the Quality of Transitional Care. *Medical Care*, 46(3), pp.317–322. Available at: http://www.ncbi.nlm.nih.gov/pubmed/18388847 [Accessed February 20, 2017].

Pérez, B. & Porres, I., 2013. Reasoning about UML/OCL Models using Constraint Logic Programming and

MDA. In *ICSEA 2013, The Eighth International Conference on Software Engineering Advances*. pp. 228–233. Available at: http://www.thinkmind.org/index.php?view=article&articleid=icsea_2013_8_10_10352 [Accessed May 21, 2015].

Reimer, U. & Laurenzi, E., 2014. Creating and maintaining a collaboration platform via domain-specific reference modelling. In *EChallenges e-2014 Conference : 29-30 October 2014, Belfast, Ireland*. IEEE, pp. 1–9.

Shenvi, R. et al., 2007. Generation of Context-Specific Electronic Patient Care Reports (ePCR) using Domain-Specific Modeling. Available at: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.131.5832 [Accessed March 8, 2017].

Silver, B., 2011. *BPMN Method and Style, Second Edition* Second Edi., Aptos, CA: Cody-Cassidy Press.

Strahringer, S., 1996. Metamodellierung als Instrument des Methodenvergleichs: Eine Evaluierung am Beispiel objektorientierter Analysenmethoden. *Publications of Darmstadt Technical University, Institute for Business Studies (BWL)*. Available at: http://ideas.repec.org/p/dar/wpaper/988.html [Accessed September 9, 2015].

The World Bank, 2014. Health expenditure, total (% of GDP). Available at: http://data.worldbank.org/indicator/SH.XPD.TOTL.ZS?end=2014&start=1995&view=chart [Accessed January 12, 2017].

Tullis, T. (Thomas) & Albert, B. (William), 2013. *Measuring the user experience : collecting, analyzing, and presenting usability metrics*, Elsevier.

Vaishnavi, V. & Kuechler, B., 2004. Design Science Research in Information Systems. *January 20, 2004*.

World Health Organization, 2016. *WHO | International Classification of Functioning, Disability and Health (ICF)*, World Health Organization.