2015 International Conference on Advances in ICT for Emerging Regions (ICTer) : 119-124

Creating a Domain Specific Modelling Method for Ambient Assistance

Judith Michael, Heinrich C. Mayr

Department for Applied Informatics/ Application Engineering
Alpen-Adria-Universität Klagenfurt, Austria
{judith.michael, heinrich.mayr}@aau.at

Abstract **– Domain specific modelling languages (DSMLs) have gained increasing popularity: they are convenient, support the productivity of modelling, and help to increase model quality and comprehensibility. Some work has been published about how to use or evaluate a DSML. In contrast to that, there is hardly any guideline for the DSML creation process and almost none for creating a Domain Specific Modelling Method (DSMM). This paper aims at contributing to fill that gap: it introduces a process for creating a DSMM. For illustration it uses a modelling language that has been created for the domain of Ambient Assistance domain.**

Keywords **- Domain Specific Modelling Language, Ambient Assistance, Method Engineering, E-Science in social sciences, Behaviour Modelling, Context aware systems**

I. Motivation

There is an on-going discussion about the pros and cons of domain specific modelling languages in comparison to the traditional generic languages like, for example, the Unified Modelling Language UML or the Business Process Model Notation BPMN. Certainly, generic languages have high merits due to their versatility in arbitrary domains as well as a broad body of experience and knowledge that has emerged from intensive use and research. On the other hand, such languages tend to follow the "law of logistic growth" [1]. I.e., they are continuously extended by researchers or standardization organizations up to point where their complexity and lack of concept orthogonality corrupts their transparency and makes them hardly manageable for practical use. Think for example of UML which grew from initially five "diagrams" up to 17 (standard) and 8 additional diagrams in the version 2.0 [2]. Such complexity may lead to misunderstandings and user demotivation.

In contrast to that a domain specific modelling language (DSML) is designed for exclusive use in a certain domain and there-in for specific purposes. Consequently they come (a) with a lean set of modelling concepts and explicit constraints that are tailored for the particular domain and purposes and (b) with lexical/graphical notations that are familiar and/or easy to understand by the users in that domain. These advantages result in an increasing popularity.

To use a DSML in practice requires, however, to embed it into a Domain Specific Modelling Method (DSMM), which features the procedure of how to apply the language as well as appropriate mechanisms to be used in such procedure.

This paper proposes a guideline for how to create such a DSMM. It illustrates the process steps by using the Human Cognitive Modelling Language (HCM-L) [3] [4] as a running example. This language was developed for modelling purposes in the domain of Ambient Assistance, and in particular within the framework of the Human Behavior Monitoring and Support (HBMS)[1] project, where it serves to represent and reproduce episodic knowledge of a certain person without any loss. With the approach presented here it should be possible to create am DSMM for any other domain, e.g., for modelling purposes in establishing a Social Life Network for farmers in Sri Lanka [5].

The paper is organized as follows: In section II we assemble the fundamentals for our approach, and discuss related work. Section II sketches the overall DSMM creation procedure which is then step by step explained in sections III – VII. Section VIII gives a short outlook on future work.

II. Fundamentals

This chapter shows how DSMLs fit into the (meta) model hierarchy, describes current work on DSML creation processes, gives an insight into the specialities of our target domain, and shortly introduces the domain itself.

A. DSMLs in the Modelling Hierarchy

The 4-level model hierarchy is widely used in academia and practice for explaining the intension/extension relationships between meta models and models and to settle the basis for meta modelling frameworks (e.g., see [6], [7], [8], [9]). Fig. 1 shows how a DSML fits into that hierarchy: On M2 we define the DSML using a meta modelling language provided on M3 (e.g., UML class diagrams). I.e. the DSML is an extension of M3 and a meta model for M1. On level M1 we use the DSML to create concrete models that are instantiated on level M0. of the instance provide a meta meta modelling language that is used The model defining a DSML is a meta-model for this language.

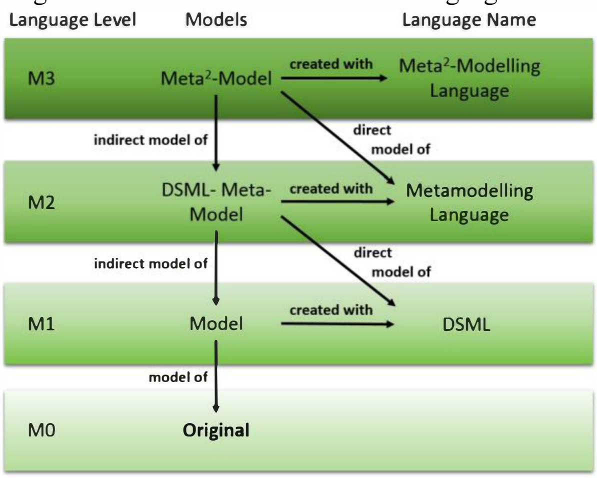

Fig. 1: Modelling Hierarchy for a DSML

[1] This work was funded by the Klaus Tschira Stiftung gGmbH, Heidelberg

978-1-4673-8270-0/15/$31.00 ©2015 IEEE

B. Creating a DSML

Whereas there is much work published on evaluating modelling languages ([10], [11]) or on how to use a DSML ([12], [13]), the process of DSML/DSMM design is rarely considered.

Frank [11] proposes a process which is intended to guide the development of a DSML for the domain of enterprise modelling. He suggests a sequence of "macro process steps", namely Clarification of Scope and Purpose, Analysis of Generic Requirements, Analysis of Specific Requirements, Language Specification, Design of Graphical Notation, Development of Modelling Tool, Evaluation and Refinement. For each of these steps again several "micro process steps" are defined.

Karagiannis and Kühn [12] define a modelling method to consist of (1) a modelling technique and (2) mechanisms and algorithms which work on the models (level M1). The modelling technique again is divided into a modelling language, in our case a DSML, and a modelling procedure, which defines the application of the language.

Our creation process is inspired by both approaches, but is to some extent more generic than [11], and more reflecting domain specific aspects than [12].

C. The example language HCM-L

When dealing with modelling in the domain of Ambient Assistance, the focus is on models as surrogates of human behaviour instead of artefacts that are used, e.g., as blueprints for database schema or software development (see [13]). HCM-L models (level M1) form a knowledge base for reasoning to optimally support a person: they are the core of the HBMS-System, and the central source of knowledge for other system components.

HCM-L is a lean modelling language for describing units of goal-driven human behaviour and its context. The key modelling concept 'Behavioral Unit (BU)' encapsulates possible sequences of actions ('Operations') that lead to a BU's goal. Concepts for context modelling cover the personal context (e.g., mental and physical restrictions of a person), the environmental context (e.g. furniture, resources), the social context (e.g. relatives), and the spatio-temporal context (e.g. location, time, frequency, duration of activities). For an in-depth description of HCM-L, the reader is referred to [3].

D. The Relevance of the Domain Ambient Assistance

The importance of assistance is obvious in view of the globally ageing population: the world population 60+ will increase from currently 12,2% to 21,2% in 2050 which will be more than 2 billion of older people [14]. Nearly 80% of these will live in the in the poorer countries of the world [15]. This global challenge forces researchers around the world to focus on Ambient Assisted Living (AAL) [16], a subdomain of Ambient Assistance. AAL aims at developing methods, tools and software systems that enable elderly people unobtrusively such that they can stay autonomous in their own homes as long as possible.

AAL was pushed substantially in 2004, when it became a Strategic Support Action (SSA) in the 6th Framework Program of the European Union, and when the *Ambient Assisted Living Joint Programme* (AAL JP) was implemented in 2008. More than 100 research and development projects have started since that time.

III. The Creation Process of a DSMM

We propose to divide the creation process in five main phases (see Fig. 2): Preparation, Modelling Language, Modelling Process, Modelling Tool and Evaluation. Each phase consists of several steps which are inspired by the work of [11] and [12] as mentioned before. The following chapters will motivate and explain each phase in detail:

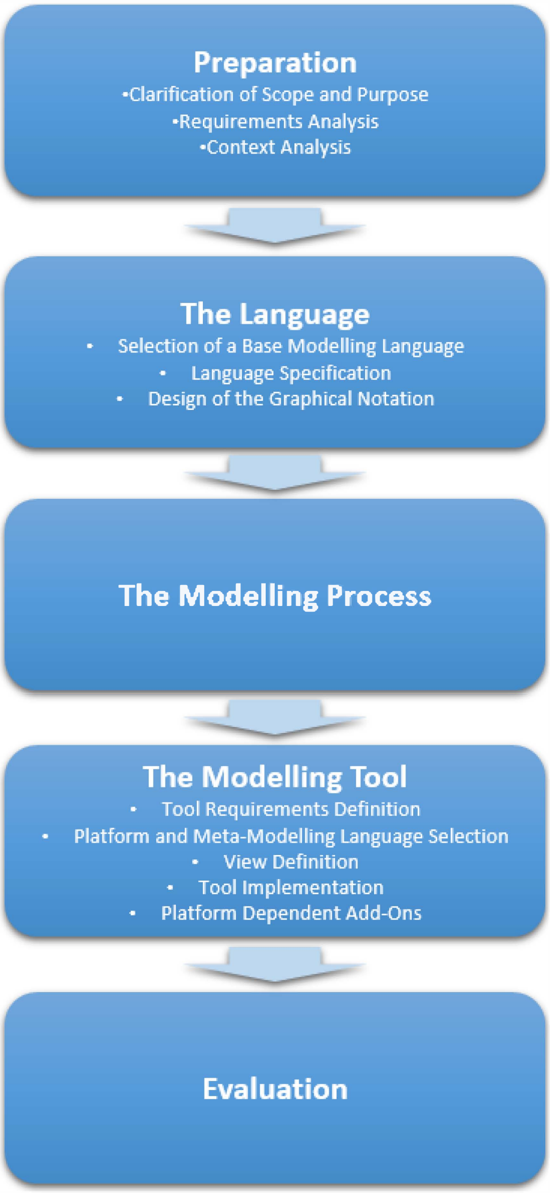

Fig. 2: The DSMM-creation process

IV. Preparation

The preparation phase ensures that all relevant facts of the Universe of Discourse (UoD), i.e. the domain in question, are known and well defined. Preparation is important for informing the subsequent phases with the knowledge needed for further development. We divide this phase in the steps *Clarification of Scope and Purpose*, *Requirements analysis* and *Context Analysis.*

A. Clarification of Scope and Purpose

A new DSMM should only be created if *need* and *purpose,* that *motivate* the endeavour, are clearly identified.

As well, the *scope* of the intended DSML has to be determined comprehensively, as it is the frame for the subsequent activities on levels M1 and M2: on level M2, the scope drives the definition of the modelling concepts to be provided as part of the meta-model, which again determines the models that can be created on level M1.

Another important aspect to clarify is the *profile* of future DSML *user groups*: users of the modelling tool, users who have to understand the models on M1, e.g., doctors in the case of AAL.

Running Example:

The main *motivation* for creating a DSMM for Ambient Assistance resulted from the *need* of a modelling language that (1) supports the goal of providing models which can be used as a knowledge base in the HBMS support system, (2) thus focuses on human behaviour and its context, and (3) is intuitively to understand by the relevant stakeholders of the AAL domain. A preceding analysis of common (generic) modelling languages revealed that these did only partly fulfil that requirements.

The *purpose* for creating the DSMM was to model human behaviour restricted to daily activities in the private home of one person. Such behaviour should be preserved in a knowledge base, and means for accessing that knowledge when needed for support had to be provided.

The *scope* is limited to the episodic knowledge [17] of a person: i.e., autobiographical events and contextual information. These are further restricted on level M1 to activities, which should be supported in the future HBMS-system. (This approach will allow the future users to benefit from the provision of own, but forgotten behavioural knowledge, independent of their age [18].)

B. Requirements Analysis

The main question in this step is: what should be modelled with the DSML? To answer that question, creating usage scenarios and exemplary diagrams is a good starting point. Another source of knowledge are domain specific standards and relevant literature, and even more important: stakeholder involvement is a must as knowing their expectations will improve the quality and acceptance of the DSMM. Clearly, requirements analysis has to be done iteratively, until a stable specification has been reached.

For further ideas what to take under consideration we refer the readers to the generic requirements and the micro process for specific requirements proposed in [12].

Running Example:

To identify the main scenarios, we went through publications of AAL projects as well as existing or upcoming standards. The so-called Activities of Daily Life (ADLs) [19], [20], a standard from health care, gave us a good starting point for our endeavour.

User involvement in this stage was ensured by two workshops and an empirical study. We thus identified four relevant areas of daily life: use of technical devices, electronic processes, daily activities and use of devices in smart homes. As the field is highly agile, new or former unknown requirements led to new scenarios and an iterative evolvement of the HCM-L-meta-model itself.

C. Context Analysis

Although context analysis also brings requirements to light, we propose to treat it as a step on its own, as it leads to a deep understanding of the given domain. Kofod-Petersen and Mikalsen, e.g., introduced a context model that focuses on a persons' surroundings, such as things, services and information accessed by the person, mental and physical information about the person, social aspects like friends or relatives, context about what a person is doing and spatio-temporal information. [21]. In the context analysis step all relevant information should be collected and reflected regarding their possible usage in the model and typical use cases. Bettini et al. [22] present a good overview of current context modelling approaches.

Running Example:

The HBMS System should adopt itself regarding the users' context, as well as regarding cognitive and physiological prerequisites of the person. This means that modelling concepts for context aspects are crucial for the language creation. Therefore, we carried deep analysis of current AAL projects, research concerning smart homes, pervasive and ubiquitous systems as well as activity and behaviour recognition. As general standards for the domain are under development but not currently used in projects, e.g., by universAAL [23], we could not use such.

V. LANGUAGE CREATION

The next phases concentrates on the language definition on level M2. As there are already many powerful (generic) modelling languages "on the market", it makes sense to evaluate these and to select one as a basis from which one can derive the intended DSML (step 1). The language definition is done by developing a meta-model (step 2) and an appropriate graphical notation (step 3).

In general a modelling language is specified by its syntax, semantics and notation. Karagiannis and Kühn [12] present an overview of current approaches to syntax and semantics definition, e.g., graph grammars or meta-models, and algebraic semantics or textual descriptions, respectively.

As regards the graphical notation of the concepts defined in the metamodel, it is worthwhile to follow Moody's nine principles for designing cognitively effective visual notations [24]. Another interesting idea is presented in [25]: novices outperform experts in designing symbols that are comprehensible to novices. So this might be a promising way of defining an effective notation. Clearly, again an iterative approach is necessary, informed by experiments involving the relevant stakeholders.

Note, that some language constructs might be too complex for graphical representation as for example, logical expressions representing conditions. For these, other appropriate forms of representation have to be defined, e.g., supported by interactive means.

Running Example:

As it was quite clear from requirements analysis, that we would need modelling language integrating dynamic (behaviour) and structural (context) aspects, we decided to use former experiences in conceptual modelling (KCPM, [26]), a lean, user-centred language for software requirements modelling, and use this as the basis for HCM-L. Consequently, the KCPM concepts were evolved and adapted to cognitive modelling. Thus we could benefit from adopting the thing concept (a more intuitive way of abstracting from classes and attributes), as well as from the KCPM approach to relate resources and actions.

In step 2, the HCM-L meta model was created in several iterative steps using a UML class-diagram-like representation. Fig. 3 shows the evolution of this meta model. After a first unstructured version, we have skipped version 1 and started in version 2 with KCPM as a basis. Furthermore, the meta model was based on ideas from Activity Theory [27], and thus, for example, provides a hierarchical structure. Other aspects added were more concrete elements, specialization of things, e.g., location things or persons. Whereas there were some structural changes in the first versions, the meta-model was stable from version 5 on. Changes till the current version 8.2 were only slight additions. The semantics of the language are defined in natural language.

As our goal was to create an intuitively understandable language, some of the (needed) complexity was displaced to a textual language, e.g., for defining conditions, pre- and post-condition expressions as well as instructions. A formal language for these part is under construction.

For the same reason, in step 3 as we decided to provide few graphical elements as possible. Some sample diagrams were created with this first draft set of modelling elements. The elements were revised in several iterations based on feedback of end users, colleagues and because of findings from [24], e.g., icons were added, the colour and thickness of different connections was changed for a higher visual distance.

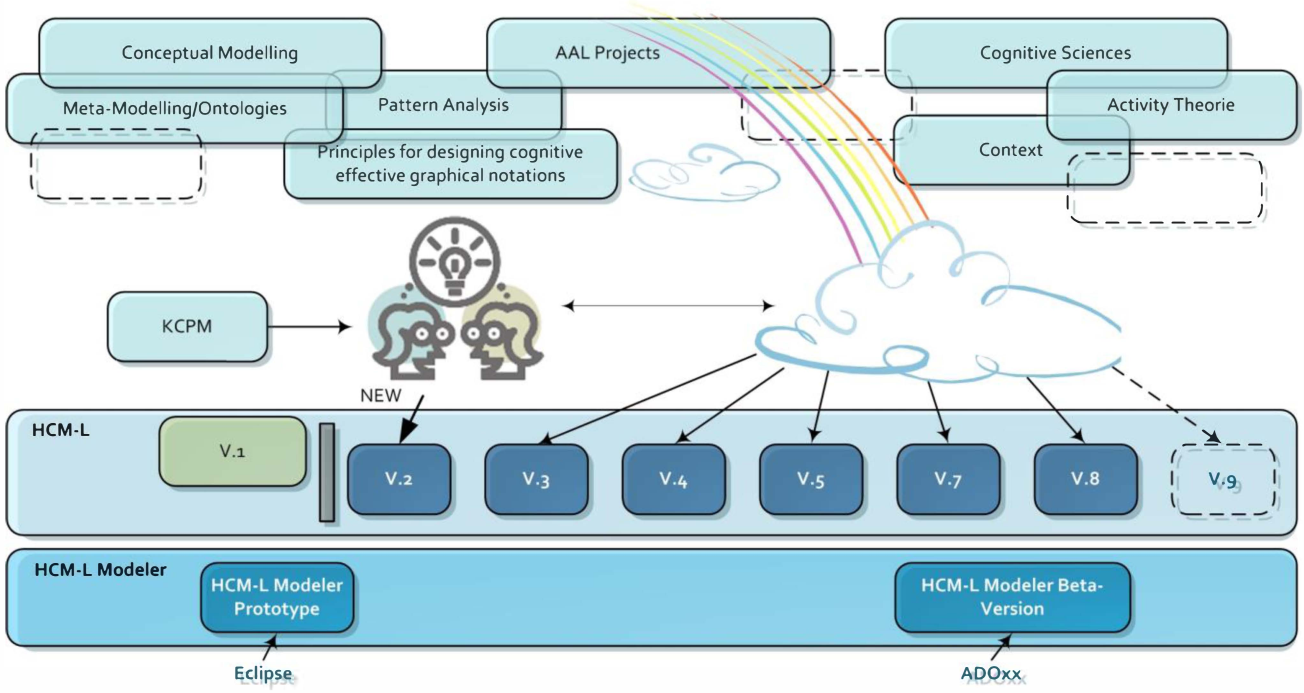

Fig. 3: The evolution of the HCM-L Meta-Model

Fig. 4 shows an example of a semantical transparent element and relation. The icon shows more intuitively that a person is modelled, the things “inside” the person box show the part-of relation in a more intuitive way than using the usual association connector (diamond shape and edge).

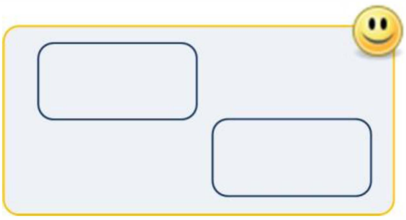

Fig. 4: Elements are part-of a person

VI. Modelling Process Definition

The modelling process definition should provide a stepwise procedure of how a particular model may be systematically built using the given DSML: with aspects should be modelled first, which view a modeller should start with (if there is more than one). The procedure should cover all modelling elements to provide a comprehensive insight for the modeller.

Becker [28] provides some (general) useful rules in his process model for creating business information models in the context of standard modelling principles (Grundsätze ordnungsmäßiger Modellierung, GOM): to model only relevant parts, to leave irrelevant parts of the UoD, or to take care about naming conventions.

Running Example:

Although, in the long run, models will be created in HBMS (semi-) automatically based on activity recognition, model transformation and integration, a modelling process has been defined for the use of HCM-L. It explains which view to start with (task context), and which elements should be created step by step in this view.

VII. Creation of the Modelling Tool

A modelling language without tool support is useless in practice. As for a newly created DSML there inherently is no ready to use modelling tool, such tool has to be created from scratch or by adaptation of an existing one. Again, several steps are to be performed in order to end up with an appropriate solution: (1) Tool Requirements Definition, (2) Selection of a Platform/Framework & a Meta-Modelling Language, (3) View Definition, (4) Tool Implementation and (5) Platform Dependent Add-Ons.

A framework for step 1 can be found in [29]. It is based on a set of tool characteristics that are grouped in four categories: methodology support, general software characteristics, documentation and environment. These characteristics can be used for a systematic requirements elicitation.

For step 2, we propose an approach opposite to that of [Frank]: instead of starting with selecting a meta modelling language (level M2), an appropriate framework/platform for tool generation should be selected for economic reasons. The decision then includes the choice of the meta modelling language. Several such frameworks or platforms [12], [30] or the ADOxx[2] Meta-Modelling Platform.

As humans have perceptual and cognitive limits, it is important to provide different views on the content and not only one complex model. Moody [24] provides some ideas which views might be helpful (step 3): to reduce the complexity, a model should be divided into smaller parts, as reducing the amount of information presented at a time facilitates the understanding; to support the cognitive integration of different diagrams, overall cognitive maps can help the reader to assemble information into a coherent mental representation of the models.

Depending on the chosen framework, the meta model of the DSML should be formulated using the framework's meta modelling language. Based here-on, the tool implementation (step 4) is generative to the extent supported by the chosen platform.

Step 5 consist in exploiting platform specific features, e.g., interfaces or coupling possibilities to external software, components for model checking, simulation, analysis, transformation or generation of documentation. These functionalities have to be checked with regard to the requirements defined for the modelling tool.

Running Example:

After exploring the requirements by use of a first prototype we performed a systematic collection of requirements for the future tool along the characteristics listed in [29]. Based here-on we decided to test the ADOxx Meta-Modelling Platform and finally have chosen it for implementation. The crucial factors for this decision were:

- The possibility of fast prototyping: a first version was completed after a month,
- Ease of changes in the meta-model and automatic tool adaptation,
- Availability of a simulation component
- Easy to combine with external software, e.g. for reasoning [Al Machot],
- Availability of analysis functionalities,
- Consistency checks are easy to realize,
- The general software and developer support works highly professional.

Concerning the views, following [21] we created integrated and aspect oriented ones: For example, to cope with the complexity of large models, there are three views on the structural context: Environmental, Spatial and Personal-Social Context. In the view of behavioural units, only relevant elements of the structural context are shown together with their links to the BU's operations. Furthermore, to provide mechanisms for cognitive integration, an Overview Map presents all BUs in the system, and there are integrated views on User Context and Structural Context (see also [3]).

Concerning step 4, the HCM-L meta model elements are derived from classes of the ADOxx Meta-Model [31], and described using the proprietary ADOxx Library Language (ALL). ALL uses the constructs defined in the ADOxx meta2model (level M3) [32].

Creating the modelling language and describing the modelling process is not enough for implementing a tool (step 4). Therefore, the general structure of (model)-documents had to be defined as well as the documents to be created and the order of creation. ADOxx uses notebooks for additional information of each element, so that we had to define the tabulators to cluster the meta-attributes meaningfully.

Some additions were implemented using the AdoScript language, e.g., automatic renaming of the element identifiers.

The ADOxx platform provides several features that are useful for our domain, e.g., functionalities for external coupling, acquisition of external data or the definition of queries using the analysis component.

Using the external coupling component made it possible to define interfaces to external reasoning mechanisms [33]. We have used AQL (ADONIS Query Language) to define queries for, e.g., checking modeling constraints.

VIII. EVALUATION

Basically, the evaluation has to be carried out against the goals and requirements determined in the preparation phase. This has to be done in co-operation with the stakeholders. In addition that, the model quality on levels M1 and M2 has to be evaluated. Frameworks for categorizing model quality, e.g., [34], [35], may serve as candidates to concentrate an evaluation on.

On level M2 if everything relevant for the given domain is included in the meta model. Ways to do this are for example to use a domain ontology or, if it is a language for behaviour modelling, to check it against the domain relevant workflow patterns [36].

On level M1 the models are to be checked against the requirements and use-cases, i.e., if all requirements and relevant use-cases are included in the models.

Running Example:

In several studies we have evaluated HCM-L regarding the goal of intuitive understandability: The results of the first study can be found in [33]. Also, a pattern based analysis has been carried out in order to verify if it is possible to model all human behaviour variants that are relevant for ambient assistance [37].

The evaluation on level M1 is still going on. We have successfully created pilot models for all relevant Use Cases; further evaluations for this level, however, will be performed.

IX. OUTLOOK

This paper showed a systematic approach to creating a DSMM and presented examples for every stage of development. The principles of that approach are not restricted to the AAL domain but can be applied to other domains as well: this has been proven by using it for the development of a DSML for a completely different domain, namely for site and quality modelling, and knowledge

[2] ADOxx is a commercial product and trademark of the BOC AG

representation in the context of Issue Management Systems [38].

Also, the first period of the HBMS project resulted very successfully in a prototype consisting of the modelling, tool, components for model management and reasoning, and a user interface for the explication of support steps.

Nevertheless, there is still a lot of work to do. Some needed add-ons are under implementation now, e.g., a model transformation in an ontology language or the integration of sequence-models. Another important development is defining the AdoScript triggers as 'transient triggers', which means the initiation of an operation at a certain time and creating an editor for the textual language.

Improvements of the DSML itself are under consideration too, e.g., for an enhanced modelling of goals and for a more intuitive way of modelling the context. Furthermore, based on the idea of [25] to let end users create elements of the graphical notation, we will provide the means for allowing multiple, user specific notations for the same language (defined by its meta model).

.

REFERENCES

[1] W. Feller, *On the logistic law of growth and its empirical verifications in biology*, Acta Biotheoretica 1940, Vol.5/2, pp 51-66.

[2] UML 2.5 Diagrams Overview, [Online] http://www.uml-diagrams.org/uml-25-diagrams.html.

[3] J. Michael and H. C. Mayr, *Conceptual Modeling for Ambient Assistance*. In *Conceptual Modeling - ER 2013*. LNCS 8217. Springer 2013, pp. 403-413.

[4] J. Michael: *Kognitive Modellierung für Assistenzsysteme*. PhD Thesis, Universität Klagenfurt, 2014.

[5] P. Di Giovanni, et.al., *Building Social Life Networks Through Mobile Interfaces: The Case Study of Sri Lanka Farmers*. In *Organizational Change and Information Systems*, P. Spagnoletti, Ed. Lecture Notes in Information Systems and Organisation. Springer 2013, pp. 399-408.

[6] International Organization for Standardization: Information technology -- Information Resource Dictionary System (IRDS) framework. ISO/IEC 10027:1990.

[7] (2002) Object Management Group: Meta Object Facility (MOF) Specification. [Online] www.omg.org/cgi-bin/doc/?formal/02-04-03.pdf .

[8] C. Atkinson, T. Kühne: Model-Driven Development: A Metamodeling Foundation, 2003.

[9] J. Bézivin: On the unification power of models, 2005.

[10] U. Frank, *Domain-Specific Modeling Languages: Requirements Analysis and Design Guidelines*. In *Domain Engineering*, I. Reinhartz-Berger, A. Sturm, T. Clark, S. Cohen and J. Bettin, Eds. Springer, 2013, pp. 133-157.

[11] U. Frank. Outline *of a method for designing domain-specific modelling languages*. ICB-Research Report 42. University Duisburg-Essen, 2010.

[12] D. Karagiannis and H. Kühn, *Metamodelling Platforms*. In *E-Commerce and Web Technologies*, K. Bauknecht, A. M. Tjoa and G. Quirchmayr, Eds. LNCS, Springer 2002, pp. 182-197.

[13] J.-P. Tolvanen and S. Kelly, *Defining Domain-Specific Modeling Languages to Automate Product Derivation: Collected Experiences*. In *Software Product Lines*, D. Hutchison et. al., Eds. Springer 2005, pp. 198-209.

[14] United Nations Population Division, World Population Prospects, the 2012 Revision, Indicator Population 60+, World, medium variant, [Online] Available: http://esa.un.org/wpp/unpp/panel_indicators.htm.

[15] United Nations, Department of Economic and Social Affairs, Population Division. World Population Ageing 2013. ST/ESA/SER.A/348, 2013.

[16] H. Steg et al.: *Europe Is Facing a Demographic Challenge. Ambient Assisted Living Offers Solutions*. VDI/VDE/IT, Berlin, 2006.

[17] G. Strube: *Wörterbuch der Kognitionswissenschaft*. Klett-Cotta, Stuttgart, 1996.

[18] J. Michael, A. Grießer, T. Strobel, and H. C. Mayr, *Cognitive Modeling and Support for Ambient Assistance*. In *Information Systems: Methods, Models, and Applications* (Revised Selected Papers.). LNBIP 137. Springer 2013.

[19] Katz, S. *Assessing self-maintenance. Activities of daily living, mobility, and instrumental activities of daily living*. American Geriatrics Society, New York NY, 1983.

[20] M. P. Lawton and E. M. Brody, *Assessment of older people: self-maintaining and instrumental activities of daily living. Gerontologist* 9, 3, 1969, pp. 179-186.

[21] A. Kofod-Petersen and M. Mikalsen, *Context: Representation and Reasoning. Special issue of the Revue d'Intelligence Artificielle on "Applying Context-Management"*, 2005.

[22] C. Bettini, O. Brdiczka, K. Henricksen, J. Indulska, D. Nicklas, A. Ranganathan, and D. Riboni, *A survey of context modelling and reasoning techniques. Pervasive and Mobile Computing* 6, 2, 2010, pp. 161-180.

[23] E. Ferro, M. Girolami, D. Salvi, C. Mayer, J. Gorman, A. Grguric, R. Ram, R. Sadat, K. M. Giannoutakis, and C. Stocklöw, *The UniversAAL Platform for AAL (Ambient Assisted Living). Journal of Intelligent Systems*, 2015.

[24] D. Moody, *The "Physics" of Notations: Toward a Scientific Basis for Constructing Visual Notations in Software Engineering*. IEEE Trans. Software Eng. 35, 6, 2009, pp. 756-779.

[25] P. Caire, N. Genon, P. Heymans, and D. L. Moody, Visual notation design 2.0: Towards user comprehensible requirements engineering notations. In IEEE 21st Int. Requirements Engineering Conference (RE), 2013, pp. 115-124.

[26] H. C. Mayr and C. Kop, *A User Centered Approach to Requirements Modeling. In: M. Glinz, G. Müller-Luschnat (Hrsg.): Modellierung 2002 - Modellierung in der Praxis - Modellierung für die Praxis*, 2002, pp. 75-86.

[27] Leont'ev, A. N. *Activity, Consciousness, and Personality*. Prentice-Hall, Englewood Cliffs, NJ, 1978.

[28] J. Becker, *Die Grundsätze ordnungsmäßiger Modellierung und ihre Einbettung in ein Vorgehensmodell zur Erstellung betrieblicher Informationsmodelle. Informationssystem-Architekturen,* Rundbrief des Fachausschusses 5.2 Informationssystem-Architekturen der Gesellschaft für Informatik, 5(2), 1998, pp. 56-62.

[29] R. Kaschek and H. C. Mayr, *A characterization of OOA tools*. In *IEEE International Symposium on Assessment of Software Tools*, 1996, pp. 59-67.

[30] Steinberg, D. *EMF. Eclipse Modeling Framework*. The eclipse series. Addison-Wesley, Upper Saddle River, NJ, 2009.

[31] J. Michael, F. Al Machot, and H. C. Mayr: ADOxx based Tool Support for a Behavior Centered Modling Approach. In: CAISE 2014, CEUR Workshop Proceeding, vol. 1164, Springer, 2014.

[32] H.-G. Fill and D. Karagiannis, *On the Conceptualisation of Modelling Methods Using the ADOxx Meta Modelling Platform*. In Enterprise Modelling and Information Systems Architectures, 2013, pp. 4-25.

[33] F. Al Machot, H. C. Mayr, and J. Michael, *Behavior Modeling and Reasoning for Ambient Support: HCM-L Modeler*. In *Proc. Int. Conf. on Industrial, Engineering & Other Applications of Applied Intelligent Systems (IEA-AIE 2014)*. LNAI 2014.

[34] Krogstie, J., Lindland, O. I., and Sindre, G. *Defining quality aspects for conceptual models*. [Chapman & Hall], [London], 1995.

[35] Batini, C., Ceri, S., and Navathe, S. B. *Conceptual database design. An entity-relationship approach*. The Benjamin/Cummings series in computer science. Benjamin/Cummings, Redwood City, Calif. [u.a.], 1991.

[36] W. van der Aalst, A. ter Hofstede, B. Kiepuszewski, and A. Barros, *Workflow Patterns. Distributed and Parallel Databases* 14, 1, 2003, pp. 5-51.

[37] H. C. Mayr and J. Michael, *Control pattern based analysis of HCM-L, a language for cognitive modeling*. In *Int. Conf. on Advances in ICT for Emerging Regions (ICTer2012)*. IEEE, 2012, pp. 169-175.

[38] V.Shekhovtsov, H.C.Mayr and C.Kop: Facilitating effective stakeholder communication in software development processes. I in: Nurcan, S., Pimenidis, E. (eds.): Lecture Notes in Business Information Processing 204, Springer 2015.