ADOxx based Tool Support for a Behavior Centered Modeling Approach

Judith Michael
Alpen-Adria-Universität Klagenfurt
Application Engineering Research Group
Austria
+4346327003734
judith.michael@aau.at

Fadi Al Machot
Alpen-Adria-Universität Klagenfurt
Application Engineering Research Group
Austria
+434632700
fadi.almachot@aau.at

Heinrich C. Mayr
Alpen-Adria-Universität Klagenfurt
Application Engineering Research Group
Austria
+434632700
heinrich.mayr@aau.at

ABSTRACT
Meta-modeling platforms that support the automatic generation of modeling tools open a new quality in information systems development for engineers: Emphasis can be put on the design and use of a modeling language that is customized to the particular needs and desired features. This may contribute to strengthen the information-system design phase as it helps to reduce the developers' aversion against overloaded modeling languages and inflexible or expensive modeling tools. Our demo paper introduces HCM-L Modeler, a modeling tool for the Human Cognitive Modeling Language (HCM-L), which has been implemented using the meta-modeling platform ADOxx. The modeler is component of an ambient assistance information system for supporting elder persons in mastering their daily life activities.

Categories and Subject Descriptors
H.1.2 **[Information Systems]:** User/Machine Systems - *Human factors, Human information processing*.
I.6.5 **[Computing Methodologies]**: Model Development - *Modeling methodologies.*

General Terms
Algorithms, Design, Human Factors, Languages.

Keywords
Conceptual Modeling, Individual Information System, Component based development, Prototype, Modeling Platform, Knowledge Management System, Meta-Modeling, Adaptive and Context-Aware System.

1. INTRODUCTION
When thinking of Information Systems (IS), most images in mind are related to the business domain: providing support for managers and their decisions, supporting business processes and thus assisting the employees in their job functions. Typically, though mostly realized based on an integrated standard system, IS are customized to the needs of respective enterprise and its users.

Ambient assistance information systems for individuals, however, require an even more personalized functionality, which leads the notion of *self-centered IS*: A system for one particular person, assisting she/him by providing information about and from herself/himself, in a way tailored to her/his abilities and needs. At a first glance, this might sound irrelevant. But think about getting older and forgetting how to use a technical device, how to use the online banking software or even how to dress yourself on or how to cook your favorite dish: then you might wish to have individual assistance for mastering your activities of daily life in order to be independent from others. This leads us to the domain of Ambient Assisted Living (AAL) [1].

The AAL-project HBMS[1] aims at saving relevant information about human behavior of a person in a cognitive model (HCM, Human Cognitive Model) and providing this information to the person when needed. To describe a person's individual HCM, the Human Cognitive Modeling Language HCM-L, i.e. a Domain-Specific Modeling Language (DSML), has been defined in order (a) to provide a user and use centered language and (b) to enable a mostly automatic model creation and integration out of sensor and/or tracking data. User centeredness should allow and simplify model validation and refinement when desired. As HCM-L is to describe behavioral ("episodic") knowledge, it can be called a conceptual cognitive modeling language.

This paper concentrates on the understandability of the modeling language as well as the modeling tool supporting HCM-L, which forms a component of our HBMS-System (an ambient assistance IS) together with reasoning modules and a web-based support tool. As the tool primarily served as a proof-of-concept for the modeling language, the novelty of the approach lies mainly in that language. The future HBMS-system users will be caregivers and the supported persons themselves.

Section 2 briefly introduces the HCM-L using an example and gives an introduction into the modeling procedure. Section 3 illustrates some features of the HCM-L Modeler. Section 4 presents the results of a study which aimed at evaluating the understandability of HCM-L for its future users. Section 5 outlines related work. Section 6 gives a resume and outlines future developments.

PETRA '15, July 01 - 03, 2015, Corfu, Greece
© 2015 ACM. ISBN 978-1-4503-3452-5/15/07…$15.00
DOI: http://dx.doi.org/10.1145/2769493.2769503

[1] The work is part of the HBMS project - Human Behavior Monitoring and Support: funded by Klaus Tschira Stiftung GmbH, Heidelberg - a research project in the field of Ambient Assisted Living

2. A MODELING LANGUAGE FOR AMBIENT ASSISTANCE

HCM-L was developed in order to provide a modeling language that is tailored to conceptualizing human behavior including all relevant context, uses approved concepts from other modeling languages, but excludes concepts that are not relevant for the AAL-domain.

When introducing a new modeling language, the question has to be answered, if this is really needed or at least justified with respect to the intended application domain. In our case, the answer to that question is definitely "yes". As natural languages evolve over time following social, economic or environmental changes, modeling languages do so, too. They have to meet given challenges as efficient and adequate as possible. Standardized languages have benefits due to their universal applicability and their wide range of concepts. However, exactly this wide range can be a handicap for the efficient and effective use of such a modeling language, in particular if non-experts – in our case doctors, caregivers or even end-users themselves – should be able to understand and validate models intuitively. Consequently, a lean modeling language for the domain of human behavior that comes with only few but appropriate concepts is justified. This is also affirmed, e.g., by the Open Model Initiative (OMI [2]) that encourages the development of domain specific modeling languages.

A modeling language without a supporting tool has not much practical value. OMI, therefore, proposes a meta-modeling platform that supports the tool implementation of any conceptual modeling language. Following this approach, we developed a HCM-L modeling tool using the meta-modeling platform ADOxx [3].

2.1 The Modeling Language HCM-L

Conceptual modeling usually starts with identifying and modeling relevant structural properties of a given Universe of Discourse: Objects (classes), relationships (associations) and properties (attributes) are determined by making use (intuitively) of the human ability to abstract and such master complexity. Based here-on, functional and dynamic aspects are modeled.

Conceptual modeling using HCM-L works the other way round, as dynamics are in focus: The dynamic aspects – the observed behavior – are modeled first, and only then, the relevant structural aspects related to the behavior, i.e. its static "context", are modeled. Activity theory [4] reinforces that point, as activities, actions and operations are in focus whereas "object orientation" (considering objects) is only one out of four more aspects.

We introduce the HCM-L only shortly using an example; more detailed information may be found in [5]. The HCM-L concepts were derived from analyzing the target AAL domain of (instrumental) activities of daily life [6] and their context [7]; the graphical notation considers the nine principles for designing cognitively effective visual notations [8].

Creating a HCM-L model starts from the most prominent elements in human behavior: activities. We call the resp. concept *Behavioral Unit (BU)*. Figure 1 shows a BU 'create a standing order'; i.e. our example stems from a personal business process and supposes that the user has already opened his access to an electronic banking portal.

Daily life activities usually have a goal, which is reached by performing a sequence of actions. These actions are captured by the HCM-L concept *Operation*, graphically drawn inside the resp. BU (expressing that a BU 'consists' of operations) and linked by *Flows*. Having executed an operation without outgoing flow means that the BU's goal is reached, i.e. in our example: 'new standing order is created'. There may be alternative actions like the three ways to receive a Transaction Authentication Number (TAN); *Pre- and Post-Condition Expressions* allow arbitrary granularity for the control flow (graphically simply by naming the logical operator, see XOR in Figure1).

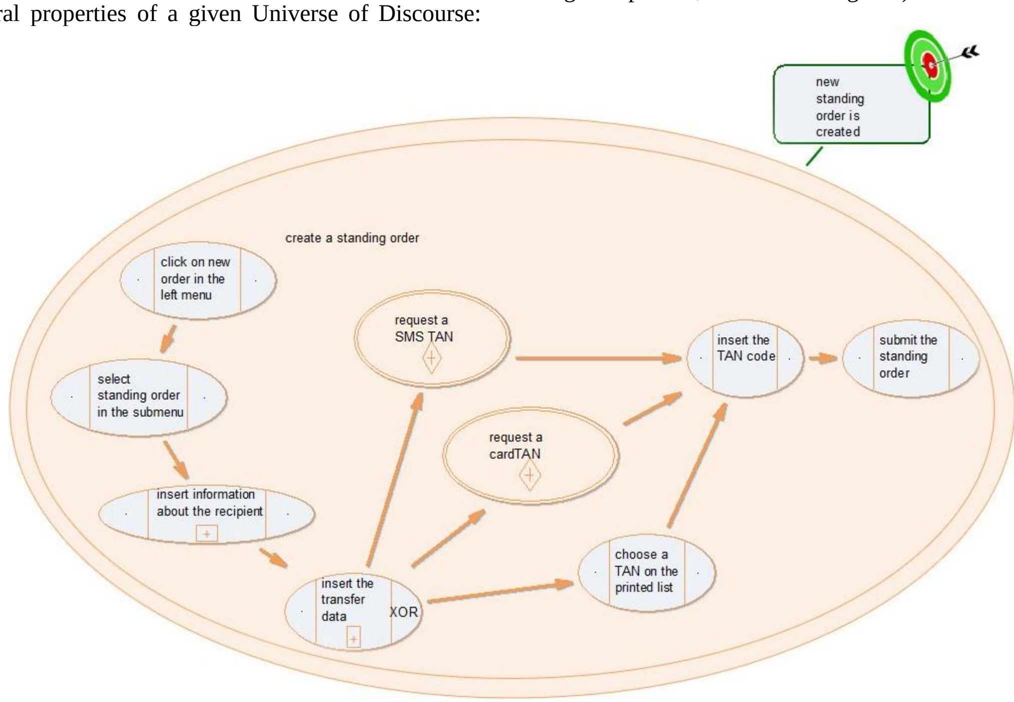


Figure 1. Example BU for creating a standing order with several Operations and a Goal

Create a standing order may be part of a larger BU 'use the online banking system'; as well, more detailed information about actions may be needed for support; e.g. to request a SMS TAN, again a sequence of actions might be necessary. Therefore, operations can be BUs, too. Thus, HCM-L allows for hierarchical structures.

As support information can not only be derived from dynamic structures HCM-L provides concepts for modeling structural, spatial, personal, social and temporal contexts as well based on the areas described in [7].

All these contexts are interrelated. Therefore, they have to be connected in a model as well in accordance with the principle of cognitive integration [8]. To support mastering of complexity, however, we provide three different views on a HCM-L model. As an example, the elements shown in Fig. 1 are a part of the view "Task Context", where the dynamic elements of one BU are displayed. In contrast to that, "User Context" and "Structural Context" are (overlapping) integrated views: Structural Context covers personal and social information about a person, the resources needed for an operation and the surrounding where the behavior takes place. User Context integrates the Task Context and, per operation, the Structural Context; i.e., in this all the structural information like calling, executing and participating elements for each operation is displayed. For further information, please refer to [5].

A comprehensive control pattern-based analysis [9] revealed, that all relevant semantics can be expressed using HCM-L when modeling activities of human behavior, their hierarchies, and the relevant context information.

2.2 From a modeling language to a modeling method

Following [13], a modeling method consists of three key components: (1) a modeling language, defined by its syntax, semantics and notation, (2) a modeling procedure, defining the steps to establish valid models, and (3) mechanisms and algorithms for model evaluation and exploitation.

In order to develop a modeling language further to a modeling method, we had to define the modeling procedure as well as mechanisms and algorithms.

The modeling process is oriented on the three development phases in the HBMS project, whereas phase 1 is the current development phase.

In phase 1, the behavior description is manually mapped to behavioral models and a person can get support by directly asking the HBMS-System for it. The creation and mapping of the models is done manually in this first phase, therefore a modeling tool is a support for creating the models. Single sequences of behavioral steps will be integrated manually (a semi-automatic integration is under progress) or a user already starts with building an integrated model. The view for that is the Task Context. In the Task Context view a BU is created and connected with a goal for that activity. Systematically each operation is modeled and connected with the structural elements, if they already exist. If some elements needed for an operation are not included in the Structural Context, they will be added stepwise and then connected with the operation as executing, calling or participating elements. In a last step, the expressions and temporal conditions are added.

In phases 2 and 3, where the behavior is observed using sensor data, the behavior sequences are generated based on these. The relevant structural elements are already included in the system, as they are defined during the installation of the system or automatically derived from AAL Systems. The sequences are connected with the structural elements and using the integration mechanism, a BU is created from several sequences. By analyzing heuristic rules, Pre- and Post-condition Expressions can be generated and their graphical realization (splits and joints) extracted from the Expressions. Already in the HCM existing sequences are not saved in the end, only their occurrence is recorded in the concerning operations and flows. The person under treatment or a caregiver can check the model and simulate the steps to validate the content.

Supplementary to the language, mechanisms and algorithms are developed. We had to define restrictions for the connection of elements in the graphical editor, to specify the graphically usable elements for each view, to create and use reasoning algorithms for gaining further information for the future users, and to make tool specific definitions, e.g., what attributes should be presented on what tab in the notebook of each element.

3. THE HCM-L MODELER

The HCM-L Modeler (see Figure 2) was developed using the meta-modeling platform ADOxx® [3] [14]. A main reason for choosing ADOxx was, that all basic modeling functions (drawing, linking and reorganizing elements, resizing, hierarchical arrangement, editing) could be implemented easily using the ADOxx Development Toolkit [3]. The HCM-L Meta-Model (also called user specific meta-model in the ADOxx context) inherits from the ADOxx Meta-Model. With ADOxx, it was possible to define and realize the graphical notation, the different context models (see [5]) and further attributes of the elements of our modeling method in a notebook-representation.

In what follows, we outline some further features that go beyond these basic ones: model stepping for an animated walk-through, querying, checking the consistency of a model, providing reasoning support, reading sensor data for complex scenarios, as well as media file management.

The HCM-L modeler is considered as a universal tool to model, manage and reason different AAL scenarios, because of the following reasons:

- It is easy to use and the provided syntax and semantic concepts are simple to understand.
- The modeler offers the possibility to import sensor data and to export the models in XML files that enables parsing, reading, accessing and writing the reasoning-data in a flexible form.
- The modeler offers the possibility to import sensor data and to export the models in XML files that enables parsing, reading, accessing and writing the reasoning-data in a flexible form.
- Different media files types are supported and can be used for different purposes either for user's recommendations or for user's understanding.
- It does not depend on a specific programming language or a specific library for reasoning. Reasoning approaches can be implemented in either ADOxx script language, JAVA, C or R that can be added as plugins into the HCM-Modeler.

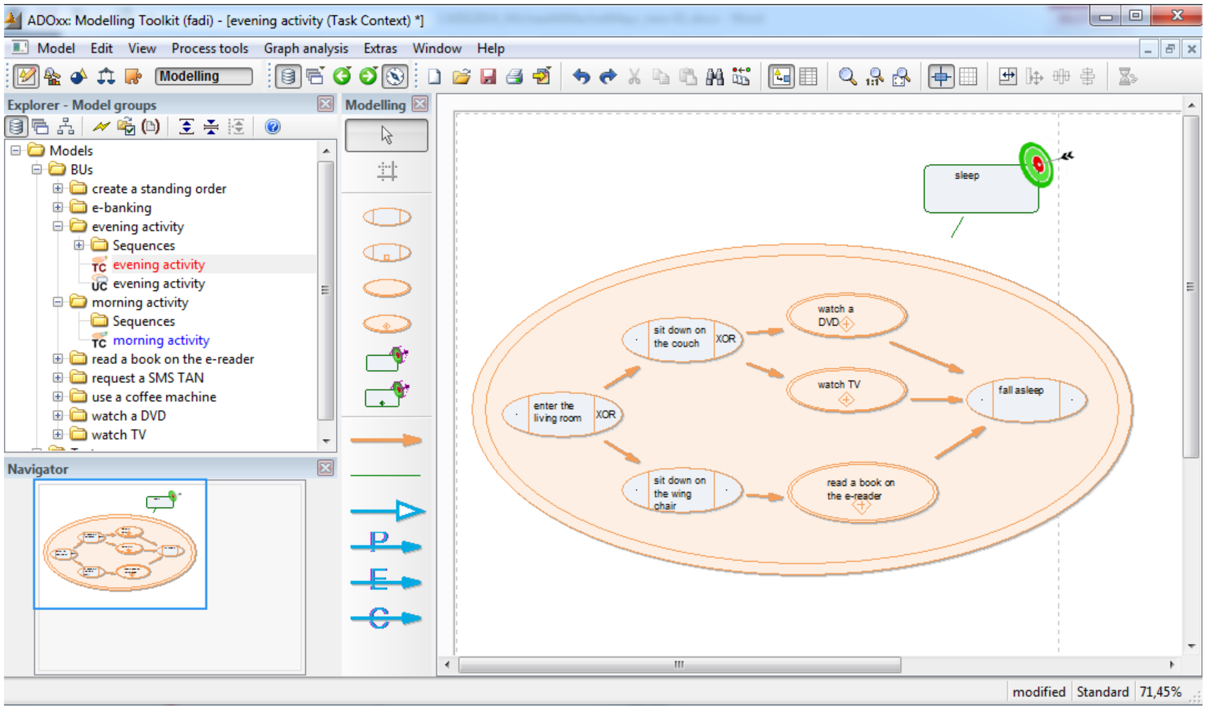


Figure 2. An overview of the proposed tool (HCM-L Modeler)

In simple words, the proposed tool is an open access lab for AAL tests and development scenarios. Fig. 2 shows an overview of the proposed tool.

3.1 Model Stepper

The stepper animates the succession of operations (of the active model) and allows a stepwise pass through a behavioral unit path based on users' decisions. This is achieved by highlighting the visited operation.

Once the stepper encounters the need of a user decision (evoked by a pre- or post-condition of the current operation), a selection window is opened where the user can choose the next step. In case of encountering a sub-unit (within a hierarchy of behavioral units), the stepper offers the choice between continuing on the current hierarchy level and walking through the sub-unit.

By visualizing the operation flows that are possible due to the model's structure, the stepper supports model understanding and validation. The long-term idea is to provide this stepper functionality to end users in order to make validation possible for them. If the model contains loops, for each loop, only one iteration will be considered and the stepper stops exactly at the final operation. It means as long as the selection of the next step is always based on the user's decision, loops are no problem.

3.2 Querying and Predefined Queries

Based on the ADOxx querying feature HCM-L Modeler supports model validation based on (predefined) queries that are formulated using the SQL-like language AQL. Such queries may concern checking the values of attributes, the coherence of elements, the compliance with predefined rules and restrictions as well as the timing of events. AQL queries can be ad hoc formulated by a user, or pre-defined by the meta-model developer in the Development Toolkit, e.g., a pre-defined query for event detection (information from sensor data). For ad-hoc formulation the HCM-L modeler provides an interactive assistant using an ADOxx basic functionality.

As an example, the following query unveils all BUs in the given model (figure 1) that should occur between 06:00am and 11:30am. The user can create the following domain specific AQL query using the AQL queries window of HCM-L Modeler:

```
(<"Behavioral Unit">[?"atTime" >=
"00:000:06:00:00"]) AND (<"Behavioral
Unit">[?"atTime" <= "00:000:11:30:00"]
```

Although this query is not terribly realistic in our running example, we show the HCM-L Modeler result in Figure 3 in order to give an impression on how the system operates: the tabular output consists of the IDs, descriptions and the titles of all behavioral units that should occur between 06:00 and 11:30.

Query results - Check the Post-Condition of Operation

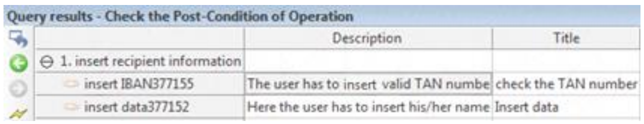
	Description	Title
1. Behavioral Units		
insert IBAN377155	Here the user has to create a standing order	create a standing order
insert data377152	Here the user has to transfer money to different recipients	perform a bank transfer

Figure 3. Answer to the AQL (possible behavioral units between 06:00 and 11:30)

Clearly, queries may be more complex by addressing value type restrictions for attributes or complex events in the sense of aggregations of simpler or atomic ones.

3.3 Consistency Check

A major issue in modeling processes is the fact that comprehensive consistency checks are difficult, in particular for inexperienced users. However, inconspicuous mistakes in the logic may affect the whole model: contradictory semantics reduce the performance of reasoning processes and yield invalid results. For the HCM-L modeler we considered three main consistency issues: (1) using the right syntax of logical operators, (2) consistent naming of model elements throughout the whole model and (3) the overall syntax check during modeling to allow the right connection between different types of classes and relation classes. Whereas (2) and (3) are automatically checked during the modeling process, (1) is accomplished using the AQL feature: After clicking on the button "pre-defined queries", HCM-L Modeler yields a menu of different consistency checks for every model and sub model, e.g., checking the correct syntax of the pre-condition label of an operation or operation-makro. Further consistency check-queries are in preparation.


Query results - Check the Post-Condition of Operation

	Description	Title
1. insert recipient information		
insert IBAN377155	The user has to insert valid TAN numbe	check the TAN number
insert data377152	Here the user has to insert his/her name	Insert data

Figure 4. Result of the condition check

Figure 4 shows the result of the consistency check "*Post-Condition of Operation with Pre Post and Suboperations*". Apparently, there were problems with the post conditions of two operations ("check the TAN number" and "Insert data").

3.4 Reasoning Support

Both model and rule based reasoning approaches for behavior modeling require the extraction of information out of the given overall model. HCM-L Modeler offers several functions for that purpose. As an example, it features the possibility to calculate the frequency of specific activities; this is based on the user history and results in a percentage value for each operation.

Another example is the calculation of the "importance value" and the "cost value" for each operation. This again is based on the user history, and on the similarity between the current user profile and other users.

Furthermore, HCM-L Modeler provides a function that determines all possible paths within a BU that lead, from a given operation, to the BU's goal (i.e. a valid end operation). Also, the length of these paths is calculated (for loops only one loop-iteration) so that the shortest one can be selected if appropriate. This function considers all sub-units. Figure 5 shows the path to the end from the operation "select valid from date" in the sub-unit "insert transfer data" (see Figure 1).

As ADOxx offers the possibility to import and export models in a generic XML format, all those reasoning attributes can be used, e.g. by external inference or reasoning tools.

In [15], we presented the reasoning approach in detail and described how the HCM-L modeler is using such XML files (ontologies) to apply inference on the exported model. The idea here is to maximize the probability of the user's currently intended behavior, so that a recommendation can be generated. In other words, the modeler determines the best fitting next operation when the user needs support.

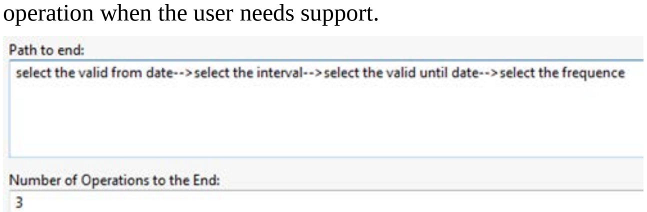


Figure 5. Path to the end (see Figure 1) from the current operation "select valid from date"

3.5 Reading Sensor Data

As already mentioned, the HCM-L Modeler is a part of a HBMS-System with different components. User monitoring will be provided through run-time by using sensors. This sensor data will be used to create the models using HCM-L (firstly simple sequences and after integration more generalized models).

ADOxx provides means to read content from files and databases to be included in the model (object or model level). It can read text, CSV, XLS, XML and DB formats. The HCM-L Modeler currently uses this feature for importing sensor data that are provided in XML.

The XML file should contain the ID of the sensor, the state of the action (true or false) and the time stamp of the selected activity. For user convenience, we included predefined AQL queries into the HCM-L Modeler to check simply active operations and their states.

Usually, sensors generate a huge amount of different measurements that must be saved and processed later for reasoning purposes, e.g., activity recognition and complex event detection. HCM-L modeler allows the export and the assign of such sensor data.

Figure 6 shows the flow of the HCM-L of how to read and how to use the imported sensor data.

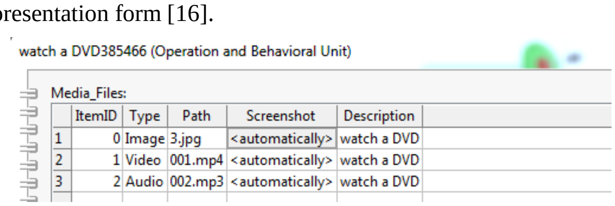
Figure 6. The overall flow of the HCM-L functionality "Reading Sensor Data"

3.6 Media Files

The HCM-L Modeler offers the possibility to upload media files (video, audio and images files in different formats) into the tool (see Figure 7). This feature allows using such files for visualizing complex issues and situations in the support phase (web-based support tool of the HBMS-System).

For example, if the user has to insert the card security code (CSC), sometimes called card verification data (CVD), the corresponding picture is presented to the user (automatically or after request) to show where this code is printed on the card.

The media files, the descriptions and the label of each operation is directly used in the support component to display a single behavioral step. This data is used to display each single step of the support information based on the selected type of support (pictures and text, videos and text, audio and text). Therefore, the existence of such data are highly required during the development of the GUI of the overall support system. In one of our end-user studies, we investigated the best way of presenting information to them by using a set of mock-ups. The results showed that a combination of pictures and audio information is the preferred presentation form [16].


watch a DVD385466 (Operation and Behavioral Unit)

Media_Files:

	ItemID	Type	Path	Screenshot	Description
1	0	Image	3.jpg	<automatically>	watch a DVD
2	1	Video	001.mp4	<automatically>	watch a DVD
3	2	Audio	002.mp3	<automatically>	watch a DVD

Figure 7. The different types of media files (image, video and audio) that are possible to be added using the HCM-L modeler

3.7 Experiences Using ADOxx

Generally, developing a modeling tool by use of a meta-modelling platform proved to be a good way for implementing a tool with basic functionalities in a short period of time. In particular, ADOxx turned out to be an appropriate platform for DSML modeling tool development.

It offers different types of predefined meta-models, a great graphical user interface, to design the elements of the modeling language.

Furthermore, the well architecture of the development tool kit, which enables handling different types of model classes and relations. Additionally, the user management system supported by the tool allows creating, editing, and documenting different versions of the desired model. The scripting language of ADOxx is a powerful language to implement complex algorithms for either reasoning or model checking. It provides a variety of libraries to ease the access to models names, models attributes, and models IDs.

In contrast to other similar tools, ADOxx supports the connection with other execution files, in the case of using external reasoning systems, e.g., DLL, JAR files and Microsoft office tools.

Moreover, the support provided by the ADOxx experts was helpful to implement the desired functionalities. They provided helpful examples additionally to the ADOxx standard tutorial.

Despite of the previous advantages, we still have more complex requirements that are not implemented yet because of limitations of the meta-meta model definition. E.g., a visualization of static and dynamic elements in one view with a stepping functionality or the generation of predefined model element instances for a certain scenario (instances of living room, dining room and kitchen for the AAL domain).

The usage of the ADOxx simulation and evaluation functionalities are not well documented. The editor of the ADOxx script does not highlight the keywords and does not show syntax errors.

4. Evaluation of Understandability

As has been pointed out in section 2, intuitive understandability for the affected user groups is a key requirement for domain specific modeling languages. To evaluate HCM-L according to that requirement a preliminary study was performed that assessed the readability and self-explanation [10] as well as the simplicity and understandability [11] of HCM-L models. In particular, we tried to find out if test persons are able to understand the following aspects in a model without previous explanations. (1) The model is about sequences of activities, (2) the sequences have a start and end point and a goal, (3) there exist merges and joints (logical XOR and AND semantics), and (4) there are nested elements (hierarchies). In a pre-test the practicability of the study was evaluated and some questions slightly changed.

The participants of the study were first year students: 10 of the lecture ‚Informatik für BetriebswirtInnen'(group 1) and 14 of the lecture 'Einführung in die strukturierte und objektbasierte Programmierung' (group 2). They were chosen because they have a basic interest in IT but nearly no knowledge about conceptual modeling.

First, we showed them a graphic (an instance of a HCM-L model) and requested them to describe as detailed as possible what this graphic is about. The analysis of the given answers was based on the occurrence of the above aspects and semantically similar words. Most of the results were not very surprising: the majority understands that the model shows some activities within a certain sequence, that there is a start and endpoint and that a hierarchy exists (see Table 1). In contrast to that, logical operators were not that well understood: the XOR semantics was understandable for 87.5%, but the AND semantics only for 58.3%.

Table 1. Results of part B of the evaluation

	Group 1	Group 2	Sum
Description of the steps	100%	92,9%	95,8%
Mentioned in the text:			
Activities	90%	85,7%	87,5%
Sequences	90%	92,9%	91,7%
Start	80%	64,3%	70,8%
End	100%	85,7%	91,7%
XOR Semantics	90%	85,7%	87,5%
AND Semantics	60%	57,1%	58,3%
Hierarchies	70%	78,6%	75,0%

Secondly, the test persons were asked a list of open questions: (1) What is the first step of the shown activities? (2) What can a person do after entering the living room? (3) Which goal has 'evening activity'? (4) Is it possible to perform the activities 'watch a DVD' and 'watch TV' in parallel? (5) Circle the element in the graphic, which is reached, if the goal is fulfilled. (6) In how many steps is the goal of the 'evening activity' reached (a) without sub steps (b) with sub steps)? (7) What means the + at the element 'switch on the TV'? What is the difference to the + at the activity 'watch a DVD'? Table 2 shows that except from finding out the sub steps any other aspect was in sum understood by 58% of the participants and more.

Table 2. Results of part C of the evaluation

Question	Group 1	Group 2	Sum
1	100%	85,7%	91,7%
2	100%	92,9%	95,8%
3	60%	64,3%	62,5%
4	90%	92,9%	91,7%
5	70%	50,0%	58,3%
6a	70%	50,0%	58,3%
6b	0%	14,3%	8,3%
7	50%	78,6%	66,7%

Question 6b relates mainly to the problem of capturing the meaning of the logical operators. For a detailed explanation of the results see [12]. A more comprehensive study with test persons from the HBMS target groups has recently be finished and will be published soon.

5. RELATED WORK

Several projects concern about activity recognition in the AAL domain. For example, [17] use smart meters to detect activities of daily living; [18] show how behavior tracking can help to address different cognitive deficits based on plan recognition; [19] introduces an ambient intelligent living assistance system for mapping of real time sensor data to activities of a person.

Regarding modeling approaches, most related research and development endeavors also are in favor of using DSMLs (see, e.g., [20]).

As AAL is a rapidly growing domain, many projects aim at providing support for people. E.g. [21] for remembering the past, remembering to perform an intended action (e.g. take a medication), or to do cognitive training; [22] uses Case Based Reasoning techniques for solving support cases in a similar way to recently performed ones; [23] focus on modeling of personal goals and user characteristics to identify a possible impact on the system goals in general. However, none of these approaches supports a comprehensive recognition and exploitation of a person's basic and instrumental daily behavior.

Regarding reasoning techniques for AAL systems, the major approaches can be categorized in three categories:

Knowledge-Based: Knowledge-Based reasoning systems can be divided in two categories, logic based and ontology based. Usually, first order logic and description logic are used to model complex contexts, e.g., Answer Set Programming [15] [24], Fuzzy Answer Set Programming (FASP) [25] and reasoning in fuzzy Ontology Web Language (OWL 2) [26] [27] [28].

Graphical Models: Graphical Models are used to model complex scenes or multimodal sensor data because of the characteristics of the inherent structure and semantics of complex activities that require higher level representation and reasoning methods. E.g., Bayesian Propagation Networks [29] [30], Dynamic Bayesian Networks [31] and Hidden Markov Models [32] [33] and Dempster-Shafer [34] [35], Conditional Random Fields (CRFs) [36] [37], Mixture Models [38] and Gaussian Mixture Models (GMM) [39].

Syntactic: Syntactic approaches are used to express the structure of a process using a set of production rules to describe the real world events, e.g., context free grammar and attribute grammars [26], stochastic free grammars [27] and fuzzy logic [28].

6. OUTLOOK

As already been mentioned, the HCM-L Modeler is just one of the HBMS information system components. The main access point for the end users will be the support component. For this, the models are transformed into a systematic description, which can be displayed on an appropriate device. A first prototype was implemented and tested with 40 people in 2012 [40]. A beta version of the support component is currently under development and evaluation.

As we have shown, the HCM-L Modeler is a powerful and comprehensive tool for developing, managing and exchanging models written in HCM-L. The next development steps for the HCM-L Modeler will focus on the design of advanced reasoning approaches, model optimization, model checker, complex event detection and sensor data fusion with respect to sensors' uncertainty.

Furthermore, we will work on the model visualization layers to show the overall model architecture (all models and sub-models with respect to the structural context) in 3D to give the modeler and the software developers the possibility to understand the models and the interaction between them more easy.

The support component will be further tested, and we will pay more intention on individual users' preferences. [23] provides some interesting ideas. Other improvements will concentrate on the definition of the support texts following previous work in computational linguistics [41]. The idea of automatic support text generation from a model (see [42]) seems to be an interesting approach.

7. REFERENCES

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

[2] Karagiannis, D., Grossmann, W. and Höfferer, P.: Open Model Initiative: A Feasibility Study, URL: www.openmodels.at (2008)

[3] Fill, H.G., Karagiannis, D.: On the Conceptualization of Modelling Methods Using the ADOxx Meta Modelling Platform. Enterprise Modelling and Information Systems Architectures 8(1), pp 4-25 (2013)

[4] Leont'ev, A.N.: Activity, Consciousness, and Personality Prentice-Hall, Englewood Cliffs, NJ (1978)

[5] Michael, J., Mayr, H.C.: Conceptual Modeling for Ambient Assistance. In: Ng, W., Storey, V.C., Trujillo, J. (eds.) Conceptual Modeling - ER 2013, LNCS 8217, Springer. (2013)

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

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

[8] Moody, D.: The "Physics" of Notations: Toward a Scientific Basis for Constructing Visual Notations in Software Engineering. IEEE Trans. Software Eng. 35, pp. 756–779. (2009)

[9] Mayr, H.C., Michael, J.: Control pattern based analysis of HCM-L, a language for cognitive modeling. In: International Conference on Advances in ICT for Emerging Regions (ICTer 2012), pp 169-175, IEEE (2012)

[10] Batini, C., Ceri, S., Navathe, S.B.: Conceptual database design. An entity relationship approach Benjamin/Cummings, Redwood City, Calif. (1991)

[11] Moody, D.L., Shanks, G.G.: What makes a good data model? Evaluating the quality of entity relationship models. In: Goos, G., Hartmanis, J., Leeuwen, J., Loucopoulos, P. (eds.) Entity-Relationship Approach — ER '94 Business Modelling and Re-Engineering, 881, pp. 94–111. Springer, Berlin, Heidelberg (1994)

[12] Michael, J.: Kognitive Modellierung für Assistenzsysteme. Dissertation, Universität Klagenfurt (2014)

[13] Karagiannis, D., Kühn, H.: Metamodelling Platforms. In: Bauknecht, K., Tjoa, A.M., Quirchmayr, G. (eds.) E-Commerce and Web Technologies, 2455, p. 182. Springer, Berlin, Heidelberg (2002)

[14] ADOxx® Software Engineer Tutorial Package. URL: http://www.adoxx.org/live/tutorial; [last visited on September 29, 2013]

[15] Al Machot, F., Mayr, H.C., Michael, J.: Behavior Modeling and Reasoning for Ambient Support: HCM-L Modeler. Modern Advances in Applied Intelligence. Springer International Publishing, pp. 388-339 (2014)

[16] Strobl, T., Katzian, A.: Darstellungsvarianten für handlungsunterstützende Apps: Ergebnisse einer Evaluationsstudie im Rahmen des Projekts HBMS, Wohnen-Pflege-Teilhabe - 7. Deutscher AAL-Kongress, Berlin (2014)

[17] Clement, J., Plönnigs, J., Kabitzsch, K.: Intelligente Zähler: ADLs erkennen und individualisieren. Smart Meter: Detect and Individualize ADLs. In: Technik für ein selbstbestimmtes Leben. VDE-Verlag, (2012)

[18] Giroux, S., Bauchet, J., Pigot, H., Lussier-Desrochers, D., Lachappelle, Y.: Pervasive behavior tracking for cognitive assistance. In: Proc. 1st ACM Int.Conf. on PErvasive Technologies Related to Assistive Environments - PETRA '08. ACM Press (2008)

[19] Nehmer, J., Becker, M., Karshmer, A., Lamm, R.: Living assistance systems. An ambient intelligence approach. In: Proc. 28th Int. Conf. on Software Engineering, pp. 43-50. ACM, 1134293 (2006)

[20] Izquierdo, J.L.C., Cabot, J.: Enabling the Collaborative Definition of DSMLs. In: Achour-Salinesi, C.B., Norrie, M.C., Pastor, O. (eds.) Advanced information systems engineering. CAiSE 2013 proceedings, LNCS 7908, pp. 272-287, Springer (2013)

[21] González, M.F., Facal, D., Navarro, A.B., Geven, A., Prost, S., Tscheligi, M.: External memory support for cognitive care, results from the first Hermes user trial. In: Ipiña González de Artaza, D.L., Bravo, J. (eds.) Proc. Workshop of Ambient Assisted Living IWAAL, pp. 17-24. Garceta, Madrid (2010)

[22] Zhou, F., Jiao, J.R., Chen, S., Zhang, D.: A Case-Driven Ambient Intelligence System for Elderly in-Home Assistance Applications. IEEE Trans. Syst., pp 179-189 (2011)

[23] Sutcliffe, A., Sawyer, P.: Modeling Personalized Adaptive Systems. In: Achour-Salinesi, C.B., Norrie, M.C., Pastor, O. (eds.) Advanced information systems engineering. Proc. CAiSE 2013, LNCS 7908, pp. 178-192. Springer, (2013)

[24] Al Machot, F., Kyamakya, K., Dieber, B., Rinner, B.: Real time complex event detection for resource-limited multimedia sensor networks. In 8th International Conference on Advanced Video and Signal-Based Surveillance (AVSS), IEEE, pp. 468–473. (2011)

[25] Janssen, J., Schockaert, S., Vermeir, D., de Cock, M.: Fuzzy answer set programming. In Answer Set Programming for Continuous Domains: A Fuzzy Logic Approach. Springer, pp. 35–48. (2012)

[26] Bobillo, F., Straccia, U.: Fuzzy ontology representation using owl 2. International Journal of Approximate Reasoning, vol. 52, no. 7, pp. 1073–1094. (2011)

[27] Bobillo, F., Delgado, M., Gomez-Romero, J.: Reasoning in fuzzy owl 2 with delorean. In Uncertainty Reasoning for the Semantic Web II. Springer, pp. 119–138. (2013)

[28] Wlodarczyk, T. W. et.al.: Swrl-f: a fuzzy logic extension of the semantic web rule language. In Proceedings of the International Conference on Web Intelligence, Mining and Semantics. ACM, p. 39. (2011)

[29] Ribaric, S., Hrkac, T.: A knowledge representation and reasoning based on petri nets with spatio-temporal tokens. In EUROCON, 2007. The International Conference on" Computer as a Tool". IEEE, pp. 793–800. (2007)

[30] Turaga, P., Chellappa, R., Subrahmanian, V., Udrea, O.: Machine recognition of human activities: A survey. Circuits and Systems for Video Technology, IEEE Transactions, vol. 18, no. 11, pp. 1473–1488. (2008)

[31] Murphy, K. P.: Dynamic bayesian networks: representation, inference and learning. Dissertation, University of California. (2002)

[32] Yamato, J., Ohya, J., Ishii, K.: Recognizing human action in timesequential images using hidden markov model. In Computer Vision and Pattern Recognition (CVPR'92), IEEE, pp. 379–385. (1992)

[33] Starner, T., Weaver, J., Pentland, A.: Real-time american sign language recognition using desk and wearable computer based video. Pattern Analysis and Machine Intelligence, IEEE Transactions, vol. 20, no. 12, pp. 1371–1375. (1998)

[34] Sentz, K., Ferson, S.: Combination of evidence in Dempster-Shafer theory. Citeseer, Vol. 4015. (2002)

[35] Wu, H., Siegel, M., Stiefelhagen, R., Yang, L.: Sensor fusion using dempster-shafer theory [for context-aware hci]. In Instrumentation and Measurement Technology Conference, 2002. IMTC/2002. Proceedings of the 19th IEEE, vol. 1, pp. 7–12. (2002)

[36] Liao, L.: Location-based activity recognition. Dissertation, University of Washington. (2006)

[37] Zhao, L., Wang, X., Sukthankar, G., Sukthankar, R.: Motif discovery and feature selection for crf-based activity recognition. In Pattern Recognition (ICPR 2010), IEEE, pp. 3826–3829. (2010)

[38] Barger, T. S., Brown, D. E., Alwan, M.: Health-status monitoring through analysis of behavioral patterns. Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions, vol. 35, no. 1, pp. 22–27. (2005)

[39] Zhuang, X., Huang, J., Potamianos, G., Hasegawa-Johnson, M.: Acoustic fall detection using gaussian mixture models and gmm supervectors," in Acoustics, Speech and Signal Processing - ICASSP 2009. IEEE, pp. 69-72. (2009)

[40] Gratzer, W.: Ontologiegestützte Assistenz durch mobile Endgeräte im Kontext von AAL, Master Thesis. Universitätsbibliothek Klagenfurt, (2013)

[41] Fliedl G., Gratzer W., Strobl T., Winkler Ch.: Mobile Instruction Apps Based on Linguistic Text Reduction. In: Rault, J.-C. (ed.) Proc. ICSSEA 2013, Paris (2013)

[42] Leopold, H., Mendling, J., Polyvyanyy, A.: Generating Natural Language Texts from Business Process Models. In: Ralyté, J. et al. (eds.) Advanced Information Systems Engineering. Proc. CAiSE 2012, LNCS 7328, pp. 64-79. Springer, New York (2012)