A Behavior Centered Modeling Tool Based on ADOxx

Judith Michael, Fadi Al Machot, Heinrich C. Mayr

Application Engineering Research Group, Alpen-Adria-Universität Klagenfurt, Austria
{judith.michael, fadi.almachot, 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, which has been implemented using the meta-modeling platform ADOxx as a component of an ambient assistance information system.

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 on the basis of 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 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 of will be care givers and the supported persons themselves.

Section 2 briefly introduces the HCM-L using an example. Section 3 illustrates some features of the HCM-L Modeler. Section 4 outlines related work. Section 5 gives a resume and outlines future developments.

2   A Modeling Language for Ambient Assistance

We introduce the HCM-L only shortly here; more detailed information may be found in [2]. The HCM-L concepts were derived from analyzing the target AAL domain of (instrumental) activities of daily life [3] and their context [4]; the graphical notation considers the nine principles for designing cognitively effective visual notations [5].
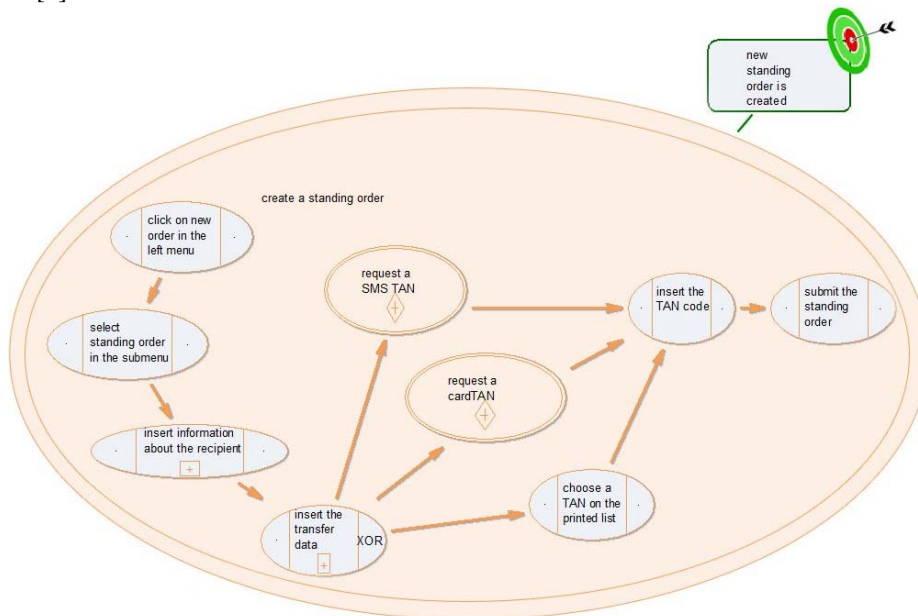

Fig. 1. Example BU for creating a standing order with several Operations and a Goal

Creating a HCM-L model starts from the most prominent elements in human behavior: activities. We call the resp. concept *Behavioral Unit (BU)*. Fig. 1 shows a BU 'create a standing order'.

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 Fig.1).

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.

Clearly, support information can not only be derived from dynamic structures. HCM-L, therefore, provides concepts for modeling structural contexts as well based one the areas described in [4]. For further information, please refer to [2].

A comprehensive control pattern-based analysis [6] revealed, that all relevant semantics can be expressed using HCM-L when modeling activities of human behavior, their hierarchies, and the relevant context information.

3   The HCM-L Modeler

The HCM-L Modeler was developed using the meta-modeling platform ADOxx[®2] [7] [8]. 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 [7]. 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 [2]) 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.

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. Basically, 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

---

[2] www.ADOxx.org

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.

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 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 Fig. 2 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

Fig. 2. 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.

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 lable of an operation or operation-makro. Further consistency check-queries are in preparation.

Figure 3 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").

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

Fig. 3. Result of the condition check

Reasoning Support

Both model and rule based reasoning approaches for behavior modeling requires the extraction of different features out of the given overall model. HCM-L Modeler, among others, offers the possibility to calculate the frequency of specific activities based on the user history: every operation is supported by a percentage value.

In addition to that, the HCM-L Modeler calculates for each operation the "importance value" based on the user history, and (if wanted by a user) the "cost value" based on the similarity between the current user profile and other users.

Furthermore, it delivers for every operation the smallest number of the remaining operations (i.e., operations to be executed) until reaching the current BU's goal, together with all possible paths leading to that goal and under consideration of all sub-units (e.g., see Fig. 4). I.e., the possible next steps are shown (for loops only one loop-iteration). As the selection of the next step is always based on the user's decision, loops are no problem.

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.
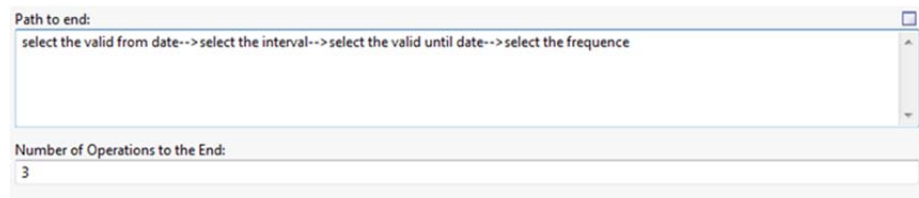
Path to end:
select the valid from date-->select the interval-->select the valid until date-->select the frequence

Number of Operations to the End:
3

Fig. 4. Path to the end (see fig.1) from the current operation "select valid from date"

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 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 simply check active operations and their states.

Media Files

The HCM-L Modeler offers the possibility to upload media files (video, audio and images files in different formats) into the tool. 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.

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.

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).

4   Related work

Several projects concern about activity recognition in the AAL domain. For example, [9] use smart meters to detect activities of daily living; [10] show how behavior tracking can help to address different cognitive deficits based on plan recognition; [11] 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., [12]).

As AAL is a rapidly growing domain, many projects aim at providing support for people, e.g. [13] for remembering the past, remembering to perform an intended action (e.g. take a medication), or to do cognitive training; [14] uses Case Based Reasoning techniques for solving support cases in a similar way to recently performed ones; [15] 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.

5   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 step-by-step description which can be displayed on an appropriate device. A first prototype was implemented and tested with 40 people in 2012 [16]. A beta version of the support component is currently under development and evaluation.

The support component uses the Operations' label, description and media files (like video, audio and images) to display a single behavioral step. 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 [17].

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. [15] provides here some interesting ideas. Other improvements will concentrate on the definition of the support texts following previous work in computational linguistics [18]. The idea of automatic support text generation from a model (see [19]) seems to be an interesting approach.

References

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

2. 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, pp 403-413, Springer (2013)

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

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

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

6. 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 (ICTer2012), pp 169-175, IEEE (2012)

7. 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)

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

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

10. 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)

11. 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)

12. 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)

13. 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)

14. Zhou, F., Jiao, J.R., Chen, S., Zhang, D.: A Case-Driven Ambient Intelligence System for Elderly in-Home Assistance Applications. IEEE Trans. Syst., Man, Cybern. C 41, 179-189 (2011)

15. 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)

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

17. 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)

18. 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)

19. 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)