Context Modeling for Active Assistance

Judith Michael (0000-0002-4999-2544), Claudia Steinberger (0000-0002-5111-2286)

Alpen-Adria-Universität Klagenfurt, Klagenfurt, Austria
{judith.michael, claudia.steinberger}@aau.at

Abstract. Context awareness is the key to any active assistance system. The Human Behavior Monitoring and Support project (HBMS) applies a multilevel context modeling approach, aiming to achieve context readability, reuse, adaptability and interoperability. The HBMS-System is the resulting active assistance system, which is multiply deployable in different domains to support the behavior of users in situations referring to the user's own episodic knowledge. The HBMS-System represents and preserves behavior and context knowledge in form of a Human Cognitive Model (HCM) expressed in a domain specific modeling language, called HCM-L. The first version of the HCM-L particularly focused on user behavior modeling. However, evaluations of first use case scenarios made clear that structural context elements like environment, spatiality and personal and social context have to be dealt in more detail. This paper summarizes the requirements for an extended HBMS context model and presents the advanced HCM-L at meta-level M2 also by giving examples on level M1.

Keywords: Domain Specific Conceptual Modeling Languages, Context Modeling, Active Assistance, Human Behavior Modelling and Support

1 Motivation

The processing of context information gives humans the ability to adapt their behavior to the world around [1]. Thus, the capability to acquire context information and to adapt to a physiological and cognitive user context is very important for systems aiming to actively assist users in situations of exhaustion, demands and excessive complexity. For [2], "*a system is context aware if it uses context to provide relevant information and/or services to the user, where relevancy depends on the user's task*".

The nature, scale and complexity of context pose challenges, which [3] assigned to four phases of a context life cycle for context aware systems: (1) **Context Acquisition**, sensing and capturing heterogeneous context information provided by physical sensors/devices and virtual sources. (2) **Context Modeling**, extracting and maintaining context of interest as models and classifying context entities and relationships between these entities. (3) **Context Reasoning**, deducing new knowledge based on available context. (4) **Context Dissemination**, distributing context information to the consuming context aware services and triggering actions based on the context.

These four phases have been a field of research for years in several areas like *Smart Homes*, *Semantic Web*, *Internet of Things*, *Pervasive Systems*, *Ambient Intelligence*,

adfa, p. 1, 2011.

In: C. Cabanillas, S. España, S. Farshidi (eds.):
Proceedings of the ER Forum 2017 and the ER 2017 Demo track,
Valencia, Spain, November 6th-9th, 2017,
published at http://ceur-ws.org

Ubiquitous Systems or *Activity Recognition* (e.g., [3], [4], and [5]). They all aim to acquire and utilize information pertaining to the physical world, to provide services accordingly and to adapt to changing context information. Thus, context awareness is the key to any active assistive system.

The HBMS project[1] applies a multilevel context modeling approach, aiming to achieve context readability, reuse, adaptability and interoperability. In our first approach [6] the HBMS context model (CM) focused mainly on user behavior and dynamics aspects. Structural context elements like the users environment, the spatial environment as well as the users personal and social situation where only dealt basically. However, use case evaluations[2] showed, that a more advanced consideration of structural context elements is necessary to be able to create models useful for the intended active support. This paper summarizes the requirements for the HBMS CM and presents our *advanced HBMS context modeling approach* at meta-level M2 giving also examples at level M1.

The paper is further structured as follows: section 2 presents the related HBMS project, its meta-modeling approach and given benefits for stakeholders in general. Section 3 discusses the state of the art of context aware systems and CMs as well as classifications of context. Section 4 defines the requirements for the HBMS CM. The advanced HBMS CM is introduced in section 5 showing the advanced meta-model and giving examples. The last section summarizes the results and gives an outlook on our future research.

2 The HBMS Project

The HBMS Project started in 2011 with the aim to actively assist individuals in activities of daily living and other situations using their own episodic knowledge. This user knowledge is represented and preserved in HBMS in the HCM, the **H**uman **C**ognitive **M**odel expressed in the domain specific modeling language ***HCM-L*** [6]. HCM-L consists of a few concepts to make it *intuitively comprehensible to relevant stakeholders of the active assisted living domain* [8]. HCM-L models are used in HBMS twofold: as conceptual models for communication and validation purposes between users and system engineers as well as machine readable context representations allowing context retrieval, reasoning, interoperability and reuse.

The 4-level model hierarchy Meta Object Facility (MOF) specification [7] is widely used in academia and practice for explaining the intension/extension relationships between meta-models and models. Each model is based on a meta-model which is based on a meta-meta-model. This stack is separated in four different *abstraction levels*: the meta-meta-model (M3) (the most abstract one), the meta-model (M2), the model (M1) and the application execution (M0). The HBMS multilevel context modeling approach uses the advantages of MOF. HCM-L concepts are modeled in HBMS at a meta-level (M2) building our HCM-L meta-model. HCM-L focuses on human behavior and its surrounding context ('things' related to behavioral steps) and provides models which

---

1 funded by the Klaus Tschira Stiftung gGmbH, Heidelberg

2 see https://youtu.be/F_wPVzq8AqM

can be used as a knowledge base in the HBMS support system. The associated HCM-L modeling tool has been developed using the meta-modeling platform ADOxx [9].[3] Based on the HCM, behavioral active assistance is provided to a user by the HBMS-System.

The HBMS approach tries to overcome the weaknesses of existing context modelling approaches: They store important user context only on application level (M0) which leads to the problem that the models are not explicitly available for communication purposes. Moreover, the HBMS approach provides:

- ***Benefits for developers:***
  - *Organization of context onto multiple abstraction levels:* the HBMS approach separates between meta-model, model and real world perspectives on user context and boosts flexibility and readability of CMs.
  - *Incorporation of user context models (M1) within the software logic* of the HBMS-System (Model Centered Architecture) as they represent an explicit formal context construction and are understandable by a computer.
  - *Interoperability between HCM-L models and external systems:* Distribution of context knowledge is possible via services for external systems (e.g., active assistance environment components in a distributed system).
  - *Domain independence:* The proposed context meta-model is independent from the domain; thus, model creation at (M1) can be easily done for other assistance domains than the proposed one.
- ***Benefits for end users:***
  - *Easy context model creation and customizing:* HCM-L enables defining CMs by using an end user friendly notation, which allows users and stakeholders to easily understand [8] and refine their CMs (M1).
  - *High readability:* CM clusters allow the handling of complexity of large amounts of context information; aggregated views on the context supports the integration of information from different diagrams (realization of the principles 'Complexity Management' and 'Cognitive Integration' by Moody [10]).
- ***Benefits for relevant user groups & systems:***
  - *Mediation of communication* between stakeholders, system administrators, administrators and the HBMS-System and also between different systems because HCM-L models can be read and understood by both, humans and computers.

3 State of the Art

Context models are used in heterogeneous context-aware application domains (e.g., in Pervasive Environments [4], for Business Processes [11], for Geographic Information Systems [12] or in the Active and Assisted Living domain (AAL) [6]). Having analyzed different context definitions and classifications, Rey and Coutaz [13] determine, that there is no context without context. Consequently, the context and its notation should be defined in terms of a ***purpose***. *The definition of context will be different, if there are*

[3] Free download at URL: http://www.openmodels.at/web/hcm-l

different purposes. Depending on the domain, different CMs fit best for a given purpose (e.g., to satisfy information requirements, or to navigate somebody). Currently, (1) there exists no standard for the specification of types of content that should be included in CMs and in conclusion, (2) today's context aware applications are still heterogeneous in their CM approaches.

3.1 Context Aware Systems and Context Models

According to Hu et al. [14] context-aware systems have traditionally been developed using one of the following three context modeling approaches:

- *Non application-level CMs*, where all actions such as context acquisition, processing, storing and reasoning are performed within system boundaries and the system directly communicates with the underlying sensing system.
- *Implicit CMs*, where systems have their own CMs, tools/libraries are used for processing context data which assist with gathering and pre-processing data but the context is still bound to the system. Clearly, from the perspective of the particular system, these CMs are explicit.
- *Explicit CMs*, where systems have their own CMs and use a shared context management infrastructure to populate their models at runtime. Context acquisition and context reasoning lie outside the system boundaries.

In the *first two approaches,* it is the responsibility of the context-aware system to acquire context data to handle faults and manage the context information and to perform context reasoning and retrieval. These facts increase the size and complexity of the application, the difficulty of implementation and reduce the possibility for context sharing with other context-aware applications. The *third approach* is based on explicit CMs allowing multiple context-aware systems to share a set of common context sources and information limiting the burden on resource constrained context sources. However, as different CMs fit best for a given purpose and no context standard is available, such common context management is hardly achievable.

In our opinion it is important for context aware systems to reflect at a conceptual level, what type of information should be considered in their CM and to ensure that this CM can easily be expressed, processed and shared. Thus, this paper focuses on an *implicit CM for the HBMS-System* enabling interoperability between different systems via a multilevel context modeling approach.

3.2 Classification of Context

A CM is needed in an active assisted system (a) to define and store context data capable of being processed for machines and (b) for communication, customization and validation purposes between users and system engineers. Because of the heterogeneity of context information, context classification is an important step in context modeling to discover context, to simplify context communication and customization, to infer possible actions and information needs and to make CMs interoperable towards different assistance systems.

The term ‘context’ has been defined and worked on by many researchers. Barwise uses the term ‘situation’ [15] to describe context. Dey and Abowd [2] see context as *“any information that can be used to characterize the situation of an entity. An entity is a person, place or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves”*. They identify location, identity, time and activity as primary context types, and secondary context as context that can be found using primary context types. Their definitions have been adapted by the research community. Gwizdka [16] extended the definition of context from Schilit and Theimer [17], and made a basic distinction between context that is internal or external to the user. Internal context describes user states, which can be made up of work context, personal events, communication context and emotional state of the user. External context describes the state of the environment, which can be made up of location, proximity to other objects, and temporal context. Petrelli [18] classified context in material and social circumstances. Material context refers to aspects such as the place of use, the device or the available infrastructure, while social context is equally important, related to aspects (being alone or not, who the others are) and personal traits (attitudes, preferences or interests).

[19] propose to use *Activity Theory* to classify context. Activity Theory [20] involves observing the nature of human activities on three levels: The level of activity (the overall process), the level of action (subtasks) and the level of operations that realize actions. While activities are informed by need, individual actions each pursue a specific goal. As the actions meet with success, the need of the overall activity is extinguished. In order to put these actions into effect, individual operations are performed. Based on that levels Kofod-Petersen and Cassens [19] introduced a CM that focuses on mental and physical information about the person (Personal Context), about what tasks a person is doing and which goal she or he has (Task Context), social aspects like relations to friends or relatives (Social Context), spatio-temporal information of situations (Spatio-Temporal Context) and a persons’ surroundings, such as things, services and information accessed by the person (Environmental Context). All together they form the User Context [19]. This CM presents a *subjective view on situations*, as the experience with a certain situation is personal. [21] classify contexts into physical and virtual contexts based on context sources. *Physical contexts* refer to contexts that can be aggregated by sensing devices. *Virtual contexts* are contexts that are specified by users or captured from user interactions, including user preferences, business processes, goals and tasks. They enable context aware applications to be more adaptive.

In addition to this context categorization schemes several more have been introduced focusing on different perspectives. [3] compares various categorization schemes and their scopes. They share common characteristics but need to be combined together in order to complement their strengths and mitigate their weaknesses.

The above-mentioned approaches for contextual classification make an effort towards creating a CM. Nevertheless, they are overlapping and mainly only *textually* described.

Since 2006 projects were carried out using an ontology-based model approach, to represent user context for their assistance system or AAL middleware. A broad variety of *upper and domain ontologies*, to classify context in assistive systems, have been developed this way (e.g., SOUPA, CoBrA-Ont, CoDAMs, the Delivery Context Ontology, mIO!, CONON, PiVON or the Situation Ontology). Beyond this user context ontologies, specifications and ontologies have been developed to describe context and environment where human activities occur (e.g., location ontologies like PlaceTime, Time ontology, different user agent profile specifications like FOAF ontology, online community specifications like the SIOC ontology and more [5]). UniversAAL was granted in 2010 with the mission of studying the results of promising previous projects integrating them into a single, consolidated one. UniversAAL assumes ontologies to be used for sharing context information between multiple applications using the universAAL middleware. By referring to the same ontology, two or more applications using the middleware can ensure, that context is being interpreted in the exact same way. But universAAL does not define any context classification or common context ontology itself but lets developers plug-in their own ontologies. As a support for ontology development, universAAL offers only a collection of proposed ontologies[4] and references existing ontology design patterns[5] to be reused or extended [22].

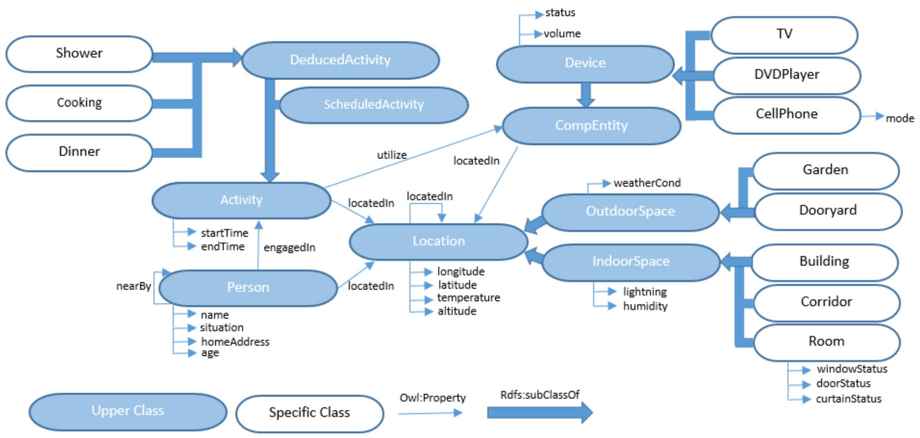

Fig. 1. COCON context ontology [23] partial.

Almost every active assistance project or context middleware has defined its own context classification, is reusing/extending existing ones or leaves that task open to application developers at all. Therefore, *context sharing and interoperability* between different systems, using the same context management infrastructure, is limited to assuming a common context ontology. Furthermore, currently used approaches are not well suited for communication and validation purposes between users and system engineers. Although existent context ontologies like COCON can be mapped to M2 (upper context ontologies, in fig. 1 marked in dark) and M1 (domain context ontologies, in fig. 1 marked in bright) [24], their representation is *too complex to use it for user communication*. Knowledge representing user behavior, environment, spatiality or profiles is not

4 e.g., https://github.com/universAAL/ontology/wiki/Physical-World
5 http://ontologydesignpatterns.org/wiki/Ontology:Main

covered by these ontologies at all. Important user context (e.g. about the structure and equipment of a flat) is mostly stored *only on application level M0* and is *not available as an explicit model* that can be used for *communication* purposes with the user.

In this paper we want to take care of these weaknesses and also cover user context as M1 models in our HBMS context model approach based on a meta-model (M2).

4 Requirements for HBMS Context Model

As presented in section 2, there is no context without context and the requirements regarding the CM of an active assistance system should be classified and defined in terms of its *purpose*.

The overall ***purpose*** of HBMS is to provide an *active assistance* system (HBMS-System) deployable in different domains supporting the behavior of according users in situations referring to the user's own episodic knowledge. Hence, the HBMS-System can be useful in different domains (e.g. to assist elderly people (AAL), to assist people working in production sectors or administration). Furthermore, we use models of behavior and other context details (e.g., special context types, behavior aspects or resources) for the purpose of communication among stakeholders, stakeholders and system administrators, administrators and the HBMS-System and also between different systems. Thereby a flexible context model has the purpose to customize the HBMS-System to domain specific application areas and their target groups.

Considering this purpose, following ***requirements*** for HBMS CM can be defined: **(1)** HBMS requires a modeling language, which enables the *definition of computer readable CMs* that are usable as a knowledge base within a model based HBMS-System architecture. **(2)** Produced CMs are expected to be *understandable also by human readers* and so by relevant stakeholders of the AA domain [8]. **(3)** HCM-L is expected to enable modeling context flexibly and to focus on human behavior and related elements (context).

Moreover Bettini et al. [4] propose to specify requirements for each domain specific purpose in regard to *heterogeneity, mobility, relationships and dependencies, timeliness, imperfection, reasoning, usability of modelling formalisms* and *efficient context provisioning*. Nevertheless, some of these requirements have limited influence on the HBMS CM itself and more on the HBMS-System working with the CM. Regarding *heterogeneity* the HBMS-System has to handle a large variety of context information sources differing in their update rate, the semantic level and flexibility: raw data from sensors, that has to be interpreted before it is usable by the HBMS-System; information about the user in a user profile, that changes rarely in correlation to the physical and mental health of a person; mostly static spatial user information (floor plans with rooms, doors, windows) and objects with often status changes like moveable objects (key, phone, purse). The HBMS CM has to handle this heterogeneous data and the HBMS-System ensures to keep the models up-to-date.

A CM must deal with *imperfection* like incorrect data, incomplete or even conflicting information. The quality of context information may differ strongly. The HBMS-

System needs to check and pre-process imperfect data before the context information is processed in the HCM, in order to keep the models correct.

There exist *relationships and dependencies* between context elements. HBMS has to handle these relationships, which can include spatial information (like an object is next to, or on another object) but also express dependencies (if objects are part of another object, they only exist as a part and not standalone). Spatial changes of objects are transitive to dependent objects. Furthermore, there exist relationships which are dependent on the perspective: from one side of the room a table is in front of a chair, from the other side the table is behind the chair.

An active assistance system is always related with *mobility*. The assistance has to be given in HBMS also via mobile devices based on information from the CM. Dependent on current locations, support will adapt itself to the environment by selecting relevant parts of information out of the CM with *efficient context provisioning* techniques. Different context views ensure *usability* and easy access to relevant context clusters. Context histories (sequences of behavior and different structural context states on M0) are preserved in the HBMS-System. *Timeliness* for support is made available by the HBMS-System by using efficient *reasoning* algorithms to provide the needed support information in time [26]. Powerful CMs provide *reasoning* mechanisms to support the user and provide consistency verification (see [26] for a comprehensive reasoning approach, based on Answer Set Programming, and model to OWL transformation).

The first HBMS CM approach focused with its version of HCM-L mainly on user behavior and dynamics aspects [6]. Structural context elements like the users environment, the spatial environment as well as the users personal and social situation where only dealt basically neglecting some of the requirements mentioned above. Thus, the first prototype of the HBMS-System was able to handle basic CM elements and relevant stakeholders had the possibility to communicate using the same modeling language. The corresponding modelling procedure has been described in [25]. However, use case evaluations showed, that not all necessary context aspects could be modeled with the available version of HCM-L at level M1 and a more advanced consideration of structural context elements would be necessary to be able to create models useful for the intended active support. In the following we introduce our advanced HBMS CM reducing these weaknesses and present a new, advanced HCM-L version.

5 Advanced HBMS Context Model

Based on context classification approaches and ontologies investigated in section 3.2, fig. 2 maps the gained context knowledge and structures of the advanced HBMS CM according to the MOF four-layer meta-modeling architecture [27]. The advanced HCM-L version is used in the following as the ‘notation concept’ to describe behavior, personal and social, spatial and environmental aspects of user context at level M1.

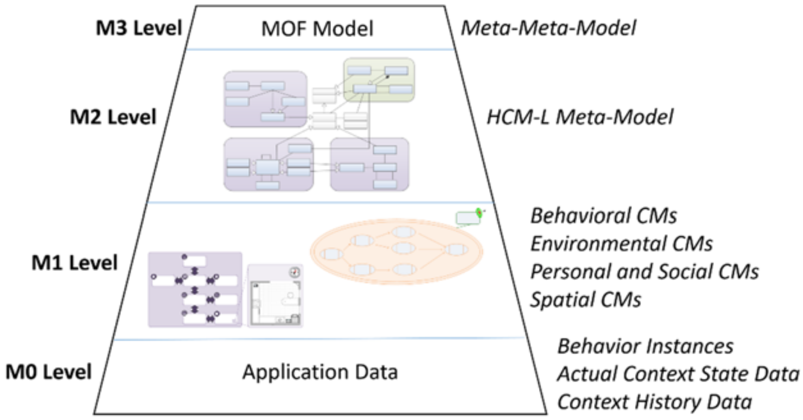

Fig. 2. HBMS context and MOF; the models will be explained in detail below

The *power of HCM-L* in describing level M1 models is determined by its meta-model, which is positioned at level M2. Thus, the HCM-L meta-model is a model of M1 models and specifies possible context abstractions in HBMS using HCM-L. Finally, M0 includes instances from the real world active assistance, like behavior instances, actual context states or context history data. Fig. 3 shows a more detailed view on the advanced HCM-L meta-model (M2) and the interconnections between the elements of its four context clusters (1) *Profile and social surrounding*, (2) *Behavior*, (3) *Environment* and (4) *Spatiality* of an assisted person.

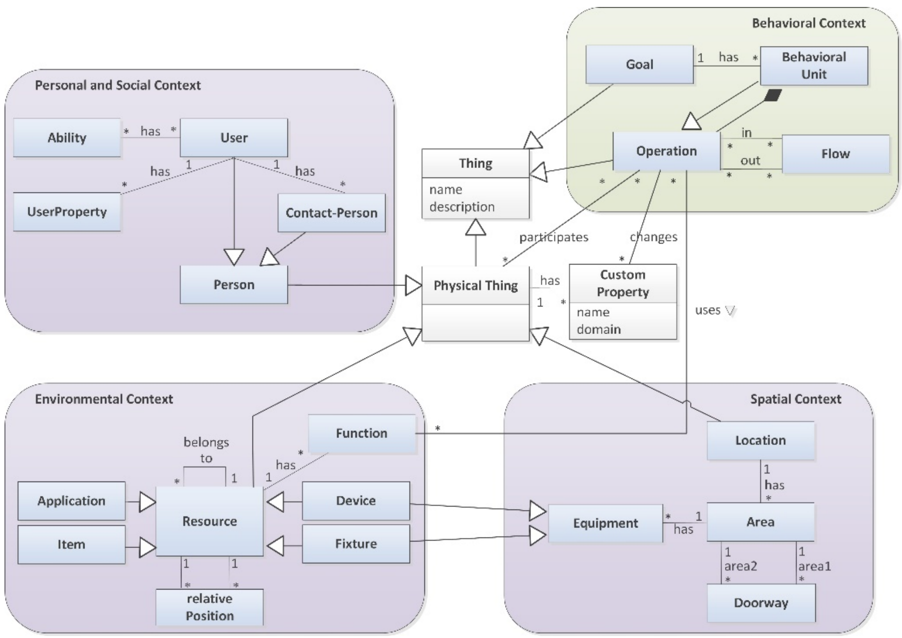

Fig. 3. Advanced HCM-L Meta-Model (M2) overview

The following subsections will detail this HCM-L meta-model clusters of user context and show some examples of corresponding level M1 models. Additionally, the interrelationships between the discussed user context clusters will be focused briefly.

5.1 Environmental Context

The *Environmental Context* of a user covers the resources that (1) have a function in operations of the assisted user or (2) are placed as equipment in the spatial context of the user and participate in operations. *Resources* (see fig. 4) can be portable or not, have looks, shapes and special types. Types are dependent on the M1 modeling domain and are defined there according to domain suggestions (e.g., in the AAL domain: cooking- cleaning- or communication-devices as types for 'device'; sleeping furniture, tables or lights as types for 'fixture'; food, drink or sanitary product as types for 'item'). On meta-model level we specialize a *Resource* to *Device* (on M1 e.g., dish washer, laptop, vacuum cleaner, TV, remote control), *Fixture* (e.g., table, tub, bed, chair, wardrobe), *Application* (e.g., online banking or hotel booking app) and *Item* (e.g., coat, umbrella, keys, sugar, bag) to stay as domain independent as possible on M2. Resources can belong together (e.g., TV and remote control) and can have a relative position to each other (e.g., the remote is on the TV; the TV is in the living room; the key is under the wardrobe). Relative positions can be changed on M0 during operation executions.

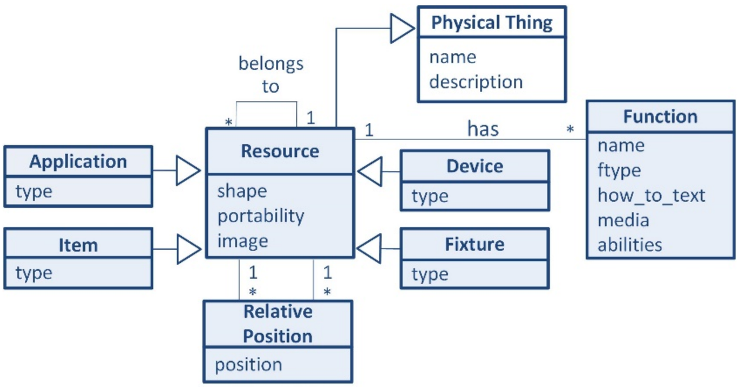

Fig. 4. Environmental Context (M2)

Resources offer functions (e.g., the device 'vacuum cleaner' offers the functions 'switch on', 'clean', 'empty dust bag'). HCM-L enables modelling this "resource user interface" as a set of *Functions* together with 'abilities' the user needs to handle and a function user guide in form of text and media (textual description how to empty the dust bag, video to demonstrate it, image to sketch necessary steps) used in *Operations* (see section 5.5). In this form operating instructions of *Devices* can be integrated into environmental CMs and can be used for active user assistance (by showing one *Operation* after another).

5.2 Personal and Social Context

The *Personal and Social Context* of a *User* (see fig. 5) covers (1) the *Abilities* that a *User* holds together with the 'level' of ability fulfilment. This enables the description of *User Abilities* concerning mobility, cognition and communication, conduct and self-sufficiency in domestic and medicinal aspects (types as 'atype'). Depending on the domain on M1, a user can have additional *User Properties* (e.g., has a pet or a certain medication in AAL). Besides this "user profile" the Personal and Social Context also covers (2) the social surrounding of a user in form of *Contact Persons* (in the AAL domain e.g., family members, friends, care persons or doctors). More properties for a person can be flexibly added as *Custom Property* of *Physical Thing*.

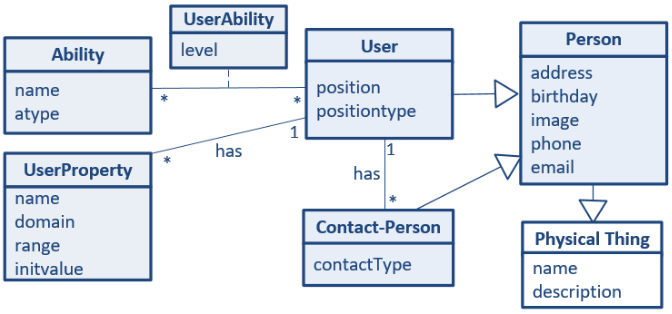

Fig. 5. Personal and Social Context (M2)

5.3 Spatial Context

The *Spatial Context* of a user covers the *Location* in which the user should be actively assisted (see fig. 6). This could be a flat or a house in the AAL domain. A *Location* can consist of several *Areas* ('atype', e.g., wet room, outdoor, living area, and pathway) with a size and shape. The *Areas* are connected via *Doorways* ('dtype', e.g., lockable door, opening, window) so that they form a kind of floorplan.

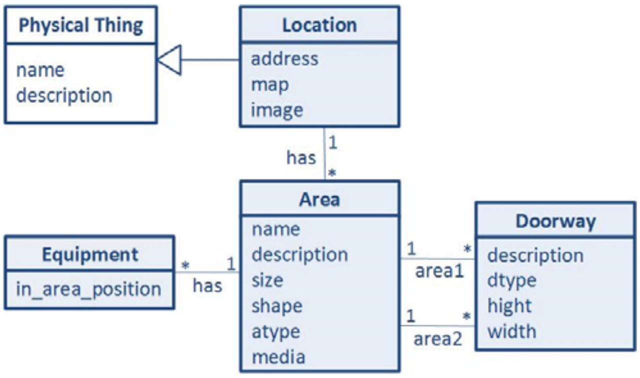

Fig. 6. Spatial Context (M2)

Each *Area* can be furnished with *Equipment* consisting of *Devices* and *Fixture* we already know from the Environmental Context. So we can position the fridge, dish washer, coffee machine, table, chair and oven as environment into the kitchen area of a user's flat location. Fig. 7 shows an example of such a Spatial CM in the AAL domain.

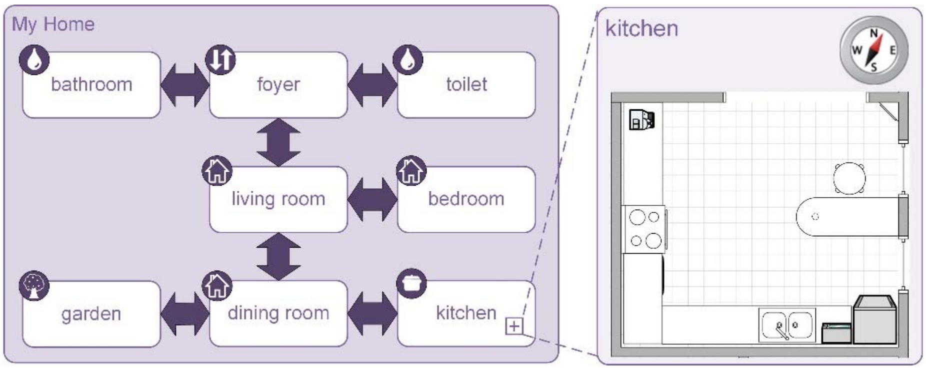

Fig. 7. Spatial Context Model (M1)

5.4 Behavioral Context

The *Behavioral Context* covers meta-modelling elements (M2) to describe abstracted behavior of the assisted user, possible sequences of actions (*Operations* connected by *Flows*) a person is doing under which conditions, which *Goal* is pursued and how other context clusters are included. Dynamic conceptual models called *Behavioral Unit* Models (BUMs) can be found as instances on M1. Behavioral context was focused in the first version of HCM-L [6][26][28] and imbedded in the advanced HCM-L approach.

5.5 Relations and dependencies between context elements

Returning to the HCM-L meta-model overview in fig. 2, there are interconnections and dependencies between all user context clusters. As *Persons*, *Resources* and *Locations* are generalized to *Physical Thing*, their characteristics can be customized on demand in dependence of the M1 level domain requirements. *Resource Functions* described in environmental models can be used for user assistance in *Operations* described in behavioural models. *Devices* and *Fixture* can be referenced and positioned as *Equipment* in spatial CMs. All *Physical Things* can participate in *Operations*.

Based on these interrelated M1-level CMs additional context knowledge, relevant for situated active assistance, can be derived applying model-based or rule-based reasoning approaches [26] (e.g., model based reasoning is used if the HBMS-System assists a person on how to watch TV in the living room). The system is aware that the required remote control is in the kitchen. Thus, it guides the person to pick up the remote control in the kitchen first, before helping with the main activity. Rule based reasoning comes into play if a fitting *Behavioral Unit* has to be deduced using information about operation frequency, a calculated cost value (similarity of the current user profile and other users) and information (e.g., the typical time an operation is performed).

6 Conclusions and further research

Natural language context classification models do not differentiate between abstraction levels at all and are not computer readable. Conventional context ontologies for assistance systems mix different context abstraction level data into one representation, which is computer- but not end user-friendly. Their main purpose is to parameterize assistance systems to stay interoperable. Specific user CMs, which are necessary to set up an assistance system, are hidden from the end user on application data level M0.

The major contribution of this paper was to introduce the advanced HBMS multi-level context modelling approach, overcoming weaknesses of the first HBMS CM approach. The basic context aspects have already been implemented in the HCM-L Modeler, which is provided as our modeling tool. A new release fully implementing the presented advanced HBMS CM is under development. Accordingly, the graphical notation of new modeling concepts needs to be adapted and complemented as well as a refinement of the procedure how to create models with the HCM-L. Evaluations with end users on the readability of the graphical notation as well as on the necessary completeness of the HBMS CM will complement this research.

Future research will deal with a refinement of modelling human goals, the investigation of relations between foundational ontologies and our meta-model, and the process to come from an implicit HBMS CM to a more explicit one, which is better reusable for other applications.

References

1. Hoareau, C., Satoh, I.: Modeling and Processing Information for Context-Aware Computing. A Survey. *New Gener. Comput. 27(3)*, pp. 177-196 (2009)
2. Dey, A.K., Abowd, G.D.: Towards a Better Understanding of Context and Context-Awareness. GVU Center Technical Reports (1999)
3. Perera, C., Zaslavsky, A., Christen, P., Georgakopoulos, D.: Context Aware Computing for The Internet of Things: A Survey. *IEEE Commun. Surv. Tutorials 16*, pp. 414-454 (2014)
4. Bettini, C., Brdiczka, O., Henricksen, K., Indulska, J., Nicklas, D., Ranganathan, A., Riboni, D.: A survey of ling and reasoning techniques. *Pervasive and Mobile Computing 6*, pp. 161-180. (2010)
5. Rodríguez, N. D., Cuéllar, M. P., Lilius, J., & Calvo-Flores, M. D.: A survey on ontologies for human behavior recognition. *ACM Computing Surveys 46 (4)*, pp. 1–33 (2014)
6. Michael, J., Mayr, H.C.: Conceptual Modeling for Ambient Assistance. In: Ng, W., Storey, V.C., Trujillo, J. (eds.): *Conceptual Modeling - ER 2013*, pp. 403-413. Springer (2013)
7. Object Management Group: Meta Object Facility (MOF) Specification. URL: www.omg.org/cgi-bin/doc/?formal/02-04-03.pdf (2002)
8. Michael, J., Mayr, H.C.: Intuitive understanding of a modeling language. In: *Proc. of the Australasian Computer Science Week Multi-conference (ACSW'17), Asia Pacific Conference on Conceptual Modeling (APCCM)*, pp. 1-10. ACM (2017)

9. Karagiannis, D., Kühn, H.: Metamodelling Platforms. In: Bauknecht, K., Tjoa, A.M., Quirchmayr, G. (eds.): *E-Commerce and Web Technologies 2455*, p. 182. Springer (2002)
10. Moody, D.: The "Physics" of Notations: Toward a Scientific Basis for Constructing Visual Notations in Software Engineering. *IEEE Trans. Software Eng. 35(6)*, pp. 756-779 (2009)
11. Saidani, O., Rolland, C., Nurcan, S.: Towards a Generic Context Model for BPM. In: *Hawaii International Conference on System Sciences* (2015)
12. Frank, A.U.: Tiers of ontology and consistency constraints in geographical information systems. *Geographical Information Science 15*, 667-678 (2001)
13. Rey, G., Coutaz, J.: Le Contexteur: une abstraction logicielle pour la réalisation de systèmes interactifs sensibles au contexte. In: *Proc. of Interaction Homme Machine*, pp. 105-112 (2003)
14. Hu, P. I.: An autonomic context management system for pervasive computing. *IEEE Int. Conf. on Pervasive Computing and Communications (PerCom)*, pp. 213-223. IEEE (2008)
15. Barwise, J., Perry, J.: Situations and Attitudes. *The Journal of Philosophy 78(11)*, pp. 668-691 (1981)
16. Gwizdka, J.: What's in the context. In: *Computer Human Interaction*. (2000)
17. Schilit, B.N., Theimer, M.M.: Disseminating active map information to mobile hosts. *IEEE Network 8(5),* pp. 22-32 (1994)
18. Petrelli, D., Not, E., Strapparava, C., Stock, O., Zancanaro, M.: Modeling Context Is Like Taking Pictures. In: *Proc. of the Context Awareness Workshop* CHI'00, ACM. (2000)
19. Kofod-Petersen, A., Cassens, J.: Using Activity Theory to Model Context Awareness. In: Roth-Berghofer, T., Schulz, S., Leake, D. (eds.) *Modeling and Retrieval of Context 3946*, pp. 1-17. Springer (2006)
20. Leont'ev, A.N.: Activity, Consciousness, and Personality. Prentice-Hall, Englewood Cliffs, NJ (1978)
21. Zhang, D., Huang, H., Lai, C.-F., Liang, X., Zou, Q., Guo, M.: Survey on context-awareness in ubiquitous media. *Multimed Tools Appl. 67(1)*, pp. 179-211 (2013)
22. UniversAAL: developers handbook - understanding ontologies. URL: https://github.com/universAAL/platform/wiki/RD-Understanding-Ontologies (2015)
23. Wang, X.H., Zhang, D.Q., Gu, T., Pung, H.K.: Ontology based context modeling and reasoning using OWL. In: *IEEE Annual Conf. on Pervasive Computing and Communications Workshops*, 2004. Second, pp. 18-22 (2004)
24. Henderson-Sellers, B.: Bridging metamodels and ontologies in software engineering. *Systems and Software* 84 (2), pp. 301–313. (2011)
25. Michael, J., Mayr, H.C.: Creating a Domain Specific Modelling Method for Ambient Assistance. In: *Int. Conf. on Advances in ICT for Emerging Regions (ICTer2015)*, pp. 119–124. IEEE (2015)
26. Al Machot, F., Mayr, H.C., Michael, J.: Behavior Modeling and Reasoning for Ambient Support: HCM-L Modeler. In: Proceedings of the International Conference on Industrial, Engineering & Other Applications of Applied Intelligent Systems (IEA-AIE 2014)
27. Kleppe, A.G., Warmer, J.B., Bast, W.: MDA Explained: The Model Driven Architecture: Practice and Promise. Addison-Wesley Longman Publishing Co., Inc. (2003)
28. Mayr, H.C., Michael, J.: Control pattern based analysis of HCM-L, a language for cognitive modeling. In: *Int. Conf. on Advances in ICT for Emerging Regions (ICTer2012)*, pp. 169–175. IEEE (2012)