Internet of Things 3–4 (2018) 52–68

Contents lists available at ScienceDirect

Internet of Things

journal homepage: www.elsevier.com/locate/iot

IoT-based human action prediction and support

Gabriel Machado Lunardi[a], Fadi Al Machot[b], Vladimir A. Shekhovtsov[c], Vinícius Maran[d], Guilherme Medeiros Machado[a], Alencar Machado[e], Heinrich C. Mayr[c,*], José Palazzo M. de Oliveira[a]

[a] Programa de Pós-Graduação em Computação (PPGC), Universidade Federal do Rio Grande do Sul, Porto Alegre, Brazil
[b] Research Center Borstel–Leibniz Center for Medicine and Biosciences, Borstel, Germany
[c] Institute of Applied Informatics. Alpen-Adria Universität, Klagenfurt, Austria
[d] Laboratory of Ubiquitous, Mobile and Applied Computing (LUMAC). Universidade Federal de Santa Maria, Cachoeira do sul, Brazil
[e] Colégio Politécnico, Universidade Federal de Santa Maria, Santa Maria, Brazil

ARTICLE INFO

Article history:
Received 5 September 2018
Accepted 10 September 2018
Available online 13 September 2018

Keywords:
Probabilistic ontologies
Uncertainty reasoning
Ambient assistance
Smart home
Context-awareness

ABSTRACT

It is an important topic in Active and Assisted Living (AAL) research and development to support elderly people suffering from memory impairment in their daily activities. A promising approach to such support is providing memory aids based on knowledge of how the person to be supported usually (i.e., in an unimpaired condition) copes with her/his daily activities. Such knowledge may be captured by IoT solutions,appropriately structured and stored in a knowledge base, and exploited when the need of support is detected. Determining the best help for a given situation implies decision-making, since the actions–flow (behavior) of an activity usually involves probabilistic branches: An automated system needs to decide which of the possible next actions is best suited for the user in a given situation. Problems of this nature involve uncertainty levels that have to be dealt with. Many approaches to this problem exploit statistical data only, thus ignoring important semantic data as, for instance, are provided by Ontologies. However, ontologies do not support reasoning over uncertainty natively. In this paper, we present a probabilistic semantic model that represent information from IoT sources and enables reasoning over uncertainty without losing semantic information. This model is implemented as an extension of the Human Behavior Monitoring and Support (HBMS) approach that provides a conceptual "human cognitive model" for representing the user–s behavior and its context in her/his living environment. The performance of this approach was evaluated using real data collected from a smart home prototype equipped with installable sensors and IoT devices. The experiments provided promising results which we will discuss regarding limits and challenges to overcome.

* Corresponding Author.
E-mail addresses: gmlunardi@inf.ufrgs.br (G.M. Lunardi), almachot@fz-Borstel.de (F.A. Machot), volodymyr.shekhovtsov@aau.at (V.A. Shekhovtsov), vinicius.maran@ufsm.br (V. Maran), guimachado@inf.ufrgs.br (G.M. Machado), alencar.machado@ufsm.br (A. Machado), heinrich.mayr@aau.at (H.C. Mayr), palazzo@inf.ufrgs.br (J.P.M. de Oliveira).

https://doi.org/10.1016/j.iot.2018.09.007
2542-6605/

1. Introduction

The likelihood of disease increases with age. Some of these diseases, e.g. memory impairments, compromise the ability to cope with the Activities of Daily Life (ADLs) [1] like dressing, cooking, using electronic devices, taking a medication and so on. This in turn affects the ability of the person concerned to live independently and self-determined in his or her accustomed home environment. Active and Assisted Living (AAL) [2–4] is the discipline that seeks for maintaining individual independence. It aims at providing appropriate assistance tailored to the particular situation by various kind of services like mobility assistance, social inclusion, communication and ADL facilitation. The challenge here is to develop robust, unobtrusive, automated, easy to use and safe systems for human assistance.

The HBMS (Human Behavior Monitoring and Support) project [5–8] addresses that challenge. It aims at deriving support services from integrated models of abilities, current context and episodic knowledge that an individual had or has, but has temporarily forgotten. The core of the HBMS system is the Human Cognitive Model (HCM) [6]. It preserves the episodic memory (roughly speaking: the flow and execution of actions in ADLs) of a person in the form of conceptual models of behavior linked to context information related to these activities. Accordingly, the HBMS system follows the Model Centered Architecture (MCA) paradigm [7].

Like in other approaches, user support is a decision-making problem in HBMS, since there is typically not only one sequence of actions leading to a (ADL) goal: often there are alternatives where the system needs to decide which of the possible next actions is the most appropriate one for the user. This can be seen as a prediction problem, since the next actions correspond to the immediate future. Problems of this nature generally involve uncertainty levels that must be represented and treated [9].

An initial solution to predicting user behavior was to reason over an OWL-DL ontology [10] representing the HCM semantic knowledge within the MCA language framework. The main limitation of this solution is that description logic (DL) does not give native support for reasoning about uncertainty; it is a subset of First Order Logic (FOL) which deals with sets of axioms, “the logical consequences of which are sentences that are true in all interpretations” [11]. FOL thus is limitedly suitable when there is uncertainty in the relationships among concepts as in our case.

Some research efforts try to overcome these limitations using fuzzy logic, neural networks or Bayesian networks. However, these strategies produce purely statistical results as the prediction is made without exploiting semantic information as could be provided by an ontology representing uncertainty.

Therefore, this paper proposes probabilistic extensions to both the HCM semantic model and its OWL representation, i.e. addresses them on semantic and language representation levels of the MCA language paradigm. This will allow reasoning over uncertainty aimed at providing support to the user.

The viability of the approach was demonstrated using a case study scenario where the user support is simulated for particular ADLs. It proved that the probabilistic semantic model supports the automatic generation of Bayesian Networks in a specific situation depending on the available evidence within that model. After that, we evaluated the performance of the approach using the HBMS smart home prototype where data was collected through sensors. This experiment provided promising results and revealed limits and challenges to overcome.

The paper is organized as follows: Section 2 presents the necessary conceptual background on the areas covered by this study. Section 3 shortly sketches HBMS, the system we used for implementing our approach which is presented in Section 4. Section 5 outlines the case study. Evaluation and discussion is exposed in Section 6. Comparison to related works is given in Section 7 followed by an outlook to future perspectives in Section 8.

2. Conceptual background

2.1. Active and assisted living based on IoT

AAL is a research field focusing on innovative solutions allowing people to age independently, safely and comfortably in their living environments [12]. It expands on the concept of Smart Home (SH), i.e. a home environment assembled with a series of sensors, actuators and devices to monitor a person’s ADLs and to take care of basic tasks like switching off electrical devices, closing windows and so on [13].

Being able to live independently roughly means being able to perform the basic and instrumental ADLs [1,14,15]. Consequently an AAL system has to be aware of these in order to provide assistance at the right time. Many AAL solutions rely, for that purpose, on the information provided by IoT smart devices [16,17]. Such devices are able to answer requests related to their set of properties, current state values, allowed operating instructions, and other information. Among the solutions supporting such devices we can distinguish the Open Semantic Framework by Siemens and the related projects [18,19], and the Weave project [20].

Other AAL approaches try to exploit the output of independent Activity Recognition Systems (ARS) [21]. An example is our own HBMS system prototype: it even may be coupled to several ARS at a time. For, during our investigations and experiments, it turned out that currently available ARS have only restricted recognition realms. As a consequence, they do not reveal all aspects to be recognized when aiming at supporting ADLs. In contrast to that, integrating and reasoning on the outputs of a combination of ARS delivers acceptable results [22].

2.2. Context-awareness and modeling

Activities take place in an environment: the context. Contextual information, therefore, is crucial for activity recognition as well as for advice deduction and providing assistance. Consequently, AAL systems have to be context-aware [23]. There are several types of contextual information such as location, people, objects, time, things, etc. [24]. Context-aware applications exploit such information for answering questions such as "who", "where", "when","what" and "with what" in order to determine "why" a particular situation is occurring [25].

This is where modeling comes in: Behavior and context models are the basis for data-driven approaches (support deduction by analytics), knowledge-driven (deduction by reasoning) or a combination of them. For example [24], when a person hesitates in an ADL and does not remember the steps (actions [26]) to complete, an AAL system could give advices based on the respective models. Among the existing techniques for context modeling are: Conceptual Modeling, Key-Value Pairs, Markup Schemas, Graphics, Object-oriented, Logic-based and Ontology Based [27].

Moore et al. [27] analyzed and compared such techniques on the basis of the following six features commonly required by intelligent context-aware systems:

- Distributed Composition: refers to the fact that context-aware applications are generally implemented in distributed and dynamic systems;
- Partial validation: ability to partially validate contextual knowledge at the structural level, as well as at the instance level, given the potential existence of errors in the definition of contextual relations between entities;
- Quality of Information: the quality of information provided by sensors (e.g. within an IoT device) over time, as well as the wealth of information characterizing entities;
- Incompleteness and Ambiguity: ability to deal with incomplete and ambiguous information;
- Level of Formality: description of facts (contextual) and interrelations in an accurate and traceable way; and
- Applicability / Adaptability: allows use in existing domains, systems, and infrastructure.

The comparison revealed that modeling techniques based on ontologies achieve good results regarding these criteria [27,28]. The most commonly used language for implementing ontologies is OWL [29].

2.3. Handling uncertainty

Uncertainty is omnipresent in what happens around us, consequently assertions are mostly uncertain, like a weather forecast; a diagnosis of a disease or the perception of the world [30]. Uncertainty may be understood as the lack of adequate information to make a decision. For example, contextual information derived from data collected by IoT device sensors is not always accurate, due to hardware failure, lack of power, communication problems, etc. Thus, a context-aware system must be sensitive to such uncertainty for making the best decisions. The representation and treatment of uncertainty appears as a crucial necessity for a satisfactory and logical reasoning [2].

Among the main approaches to reasoning about uncertainty we highlight: Fuzzy logic, probabilistic logic, Bayesian networks, Hidden Markov model and Dempster–Shafer's theory of evidence. Each of these has a different suitability depending on the intended domain. In general, probabilistic reasoning allows the processing on uncertainty, because it treats hypotheses to identify the probability of occurrence even before the set of information that evidences them is known. With this, it is possible to make predictions through uncertain reasoning [29].

Bayesian Networks (BN) are commonly used to cover the limitation of DL regarding uncertainty. A BN models a set of random variables $X = X_1, \ldots, Xn,$ each variable being a node of a Directed Acyclic Graph (DAG). The nodes are connected through directed arcs representing dependence between them. For example, let $Xi \rightarrow Xj$ represent that Xi is the parent of Xj. Each node of the graph contains a Conditional Probability Table (CPT) $P(Xi|parents(Xi))$ which represents the probability of Xi conditioned to its parents. This suggests to combine for uncertainty treatment the expressive power of FOL as is intrinsic to OWL-DL encoded ontologies, and probabilistic reasoning through Bayesian Networks. This idea is addressed by the area of probabilistic ontologies.

2.4. Probabilistic ontologies

Probabilistic ontologies can be defined as being explicit, formal, structured and shareable representations of knowledge about an application domain by means of: entity types; properties of these entities; relationships; processes and events that occur with these entities; statistical regularities that characterize the domain; inconclusive, ambiguous, incomplete, unreliable knowledge related to domain entities; and uncertainty over all previous forms. As such, probabilistic ontologies expand the expressive power of standard ontologies by allowing for an adequate representation of the statistical regularities of an application domain [31].

One way to generate such ontologies is to use a First-Order Probabilistic Language (FOPL) which combines aspects of probabilistic representation with FOL. The Multi-Entity Bayesian Network (MEBN) can be regarded as the state-of-the-art of such languages [32]. To combine MEBN with OWL-DL, an upper ontology called Probabilistic Ontology Web Language (PR-OWL) was created by P. Costa [31] the expressiveness of which is powerful enough to represent even highly complex domains. Currently it is in second version, and an open-source visual tool called UNBBayes (available at: http://unbbayes.sourceforge.net/) was developed to support the construction of PR-OWL2 ontologies [33].

PR-OWL is a first order Bayesian logic based on MEBN and as such integrates FOL with probability theory. The Bayes' Theorem provides a basis for inference [34].

2.5. Multi-entity Bayesian networks

In practical terms, MEBN represents an Universe of Discourse as a set of interrelated entities and their respective attributes. Knowledge about the attributes of entities and their relationships is represented as a collection of MEBN fragments (MFrags) organized in MEBN theories (MTheories). A MTheory is defined as a set of MFrags that together satisfy constraints of consistency ensuring the existence of a single joint probability distribution. An isolated MFrag can be compared to a standard BN in which its random variables are called nodes representing the attributes and properties of a set of entities and its arcs are relations of direct dependence between nodes [11]. It is also worth to say that MEBN by itself does not specify a standard for CPTs, but the UnBBayes tool provides a flexible way of declaring them to be called Local Probability Distribution (LPD). This is accomplished through a special scripting language. Further details can be found in [35].

An MFrag consists of three types of nodes [35].

- resident nodes: random variables forming the center of a MFrag. The LPDs are defined here exclusively and explicitly. If they can not be explicitly defined, a uniform distribution is assumed. The possible values of a resident node can be an existing entity or a list of mutually exclusive values;
- input nodes: influence the distribution of the resident nodes; their distributions are set in their own MFrags;
- context nodes: represent conditions that must be satisfied for the influences and local distributions of an MFrag;

The MEBN reasoning and inference process is carried out, first, by the interposition of a question which implies the generation of a Specific Situation Bayesian Network (SSBN). This is a common BN that aims to determine the probabilities of a situation in the domain. Instances and evidences (known facts of the domain) have a fundamental role in this stage as they are the ones that compose the questions to be submitted to the reasoning algorithm [36].

2.6. Model centered architecture

Our contribution is inspired by the Model Centered Architecture (MCA) paradigm [7,37]. According to this paradigm, an information system is treated as a compound of various models and ''model handlers" that produce and/or consume/exploit models. The models, in particular, describe not only the application domain and it's functionality but also the system's interfaces (Fig. 1). Each of these models is formed with the means of a Domain Specific Modeling Language (DSML) providing the backbone for semantics and structure, and for computer-aided processing, represented by means of a related Domain Specific Representation Language (DSRL).

MCA comprises a *metamodel hierarchy*, a *representation language framework*, and a set of *architectural patterns*.

Metamodel hierarchy

The metamodel hierarchy forms the semantic core of a MCA system: MOF [38] level M2 defines the metamodels of the particular DSMLs, M1 specifies all system and data components in the form of extensions (instances) of the respective M2 metamodels; M0 represents extensions (instances) of the M1 models, i.e. models of concrete objects, functions and processes. Fig. 1 shows the different areas of conceptualization of the MCA paradigm in more detail:

1. the application domain(s) together with application data exchange and user interfaces; for every application (sub)domain as well as for every interface, an appropriate M2-level DSML has to be defined;
2. the model exchange and model management interfaces. The management interfaces allow for handling metamodels and models as artifacts; the exchange interfaces allow for the export and import of (meta)models on all levels and thus support the integration of artifacts from external sources. A concrete application system results from the instantiation of the M2 metamodels on the M1 level. The running application itself results from creating extensions of M1 model elements on the M0 level, for example by MDA [39] or models@runtime [40] mechanisms.

Language representation framework (Fig. 2)

This framework provides the syntactic means for representing the core semantics by use of different appropriate representation languages: from grammar definition means via the concrete grammars for the languages specifying the means of representing semantic concepts, and down to providing concrete language representations of core semantic concepts from different levels. As depicted on Fig. 2, we distinguish the following elements of the language definition framework:

1. Grammar definition elements. In our research, we use a specific version of EBNF, compatible with the ANTLR grammar definition language [41].
2. Language definition elements: define grammars for the various representation languages (RL) related to the defined DSMLs: meta-meta-model RLs (not shown on Fig. 2 for brevity), meta-model RLs, model RLs and instance/data RLs.
3. Representation elements: representations of the models of all levels.

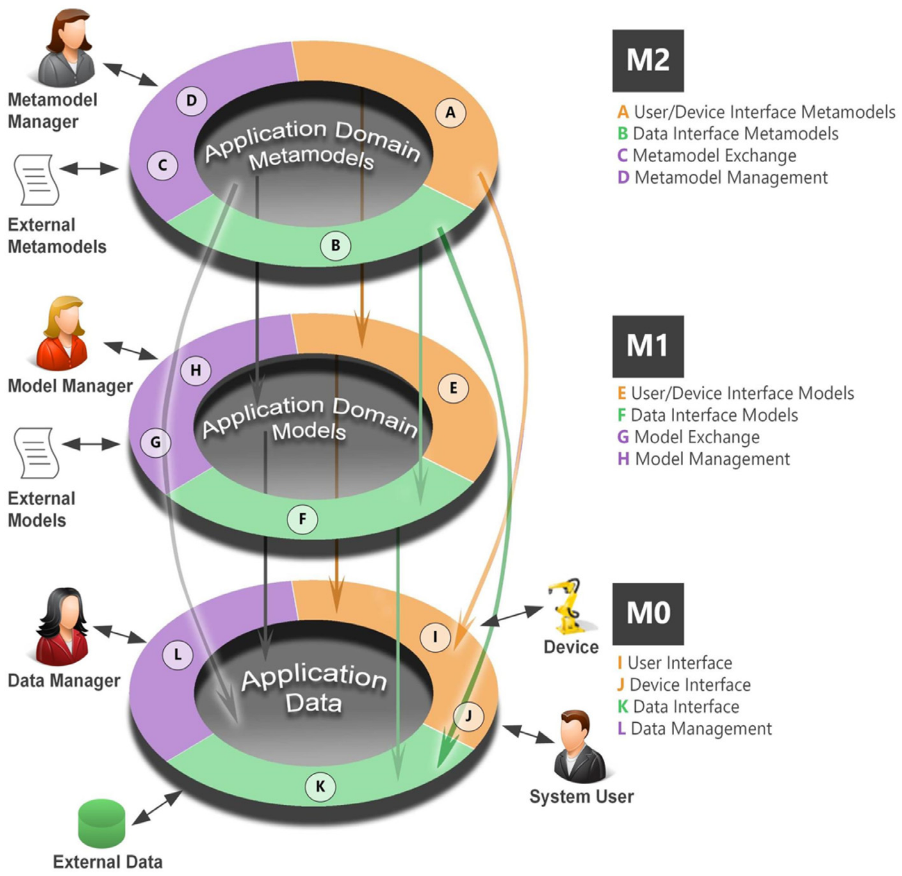

Fig. 1. MCA Architecture (adapted from [7,37]).

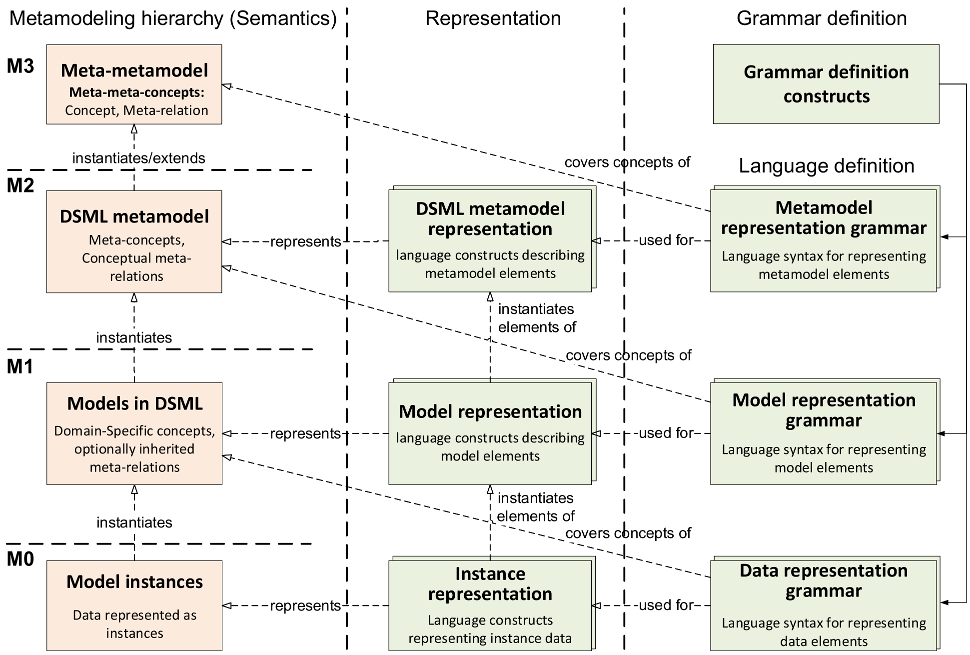

Fig. 2. MCA Language architecture (extended from [37]).

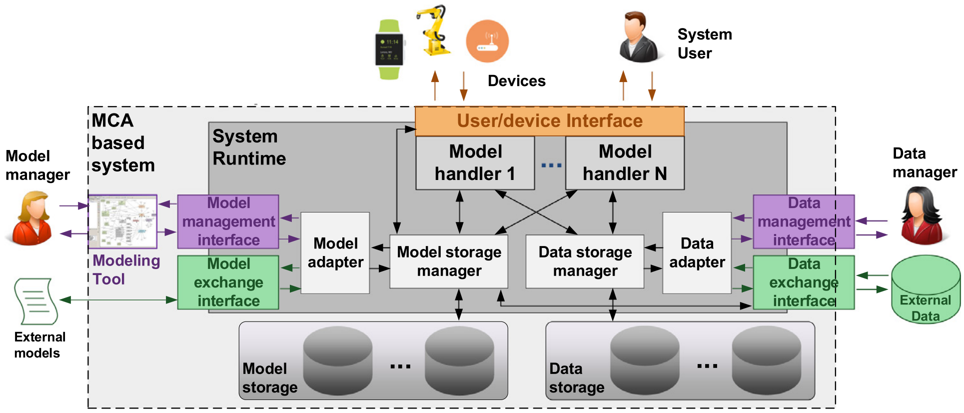

Fig. 3. MCA pattern architecture (from [37]).

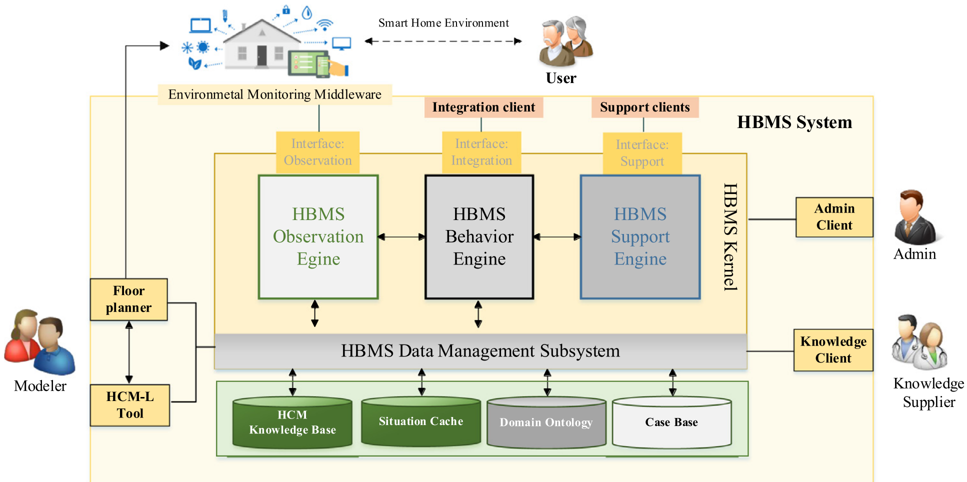

Fig. 4. Simplified HBMS-System's components and its stakeholders [6].

Architectural patterns

A set of architectural patterns (Fig. 3) includes the modeling tool pattern (components used for creating models), the model transfer interface and the model adapter patterns (for the components providing the models to the consumer system), the model management pattern, the storage pattern, and the model handler pattern which describes the components, that use, create and manipulate the models to provide the functionality of the system.

3. Human behavior monitoring and support

The HBMS (Human Behavior Monitoring and Support) system is the result of an AAL project of the same name. It's aim is to maintain, as long as possible, the personal autonomy of an individual with mild to moderate cognitive impairments in performing his/her daily activities. The HBMS system establishes two closely interwoven processes:

1. In the *learning mode* (i) the behavior of a target person is observed using available activity recognition infrastructure and systems, and (ii) these observations are transformed, integrated and preserved in a knowledge base, the conceptual Human Cognitive Model (HCM).
2. In *the support mode*, the person's behavior is observed as well, however primarily for recognizing situations of need of assistance; if such situations occur, the HCM and a domain ontology are exploited for determining the most appropriate action to help the person reaching her/his current goal [6].

The overall system's architecture implements MCA architectural patterns [7]. This architecture as well as the stakeholder's roles are depicted in Fig. 4: The target user lives in a smart home. The modeler uses the HCM-L tool to (1) specify the floor plan of the smart home, (2) adapt and extend, if necessary, the HCM established by observation, or (3) establish the HCM manually if no automatic recognition is available. The modeler supports the *Human Cognitive Modeling Language (HCM-L)*, a DSML for describing a person's episodic knowledge (autobiographical events and contextual information) and its context in the form of conceptual models [42]. The Administrator manages the system installation and configuration, and monitors the system's state. The knowledge supplier provides and integrates domain knowledge into the HBMS data; moreover, the knowledge client may be used by doctors and caregivers for diagnosis purposes. Clearly, any kind of access to the user's personal data is based on previous access authorization.

The HBMS Observation Interface (HBMS-OI) acquires observed behavioral data from heterogeneous activity recognition systems (environmental monitoring middleware in Fig. 4). The interface to such systems is again defined using a DSML, the *Activity Recognition Environment Modeling Language* (AREM-L) [22]. The remaining HBMS kernel components are the "model handlers" according to the MCA paradigm: The *observation engine* listens to the data arriving from HBMS-OI, analyzes and processes this data, and transfers recognized behavior situations to the HBMS *Behavior Engine*. The Behavior Engine transforms the received information into HCM-L model fragments (recognized context and behavior). In learning-mode, these fragments are collected to form sequences of actions which then are integrated into the (growing) HCM. As the integration is based on heuristic rules, interaction via the integration client might improve the HCM creation if the target user is able and willing to cooperate. In support mode, the Behavior Engine continuously matches the observed behavior against the HCM, retrieves appropriate knowledge from HCM while looking for discrepancies, determines the necessary knowledge to correct these discrepancies (e.g. in a form of sets of correcting actions) and transfers such knowledge to the HBMS *Support Engine*. The latter is responsible for the context sensitive multi-modal assistance of the target user by using an appropriate support client [6].

The HBMS data layer is composed of at least four data sources: (i) The *HCM Knowledge Base* stores the complete cognitive model of a target user. (ii) The *Situation Cache* contains currently relevant and referenced HCM fragments, and state data collected from observations; i.e., it is the system's operational knowledge base. (iii) The *Domain Ontology* stores domain-specific knowledge that is referenced from both HCM and Situation Cache. (iv) The *Case Base* stores all observed action sequences. Making use of this information for determining the most likely next step to be advised is the challenge addressed in this paper.

The development of HCM-L was guided by the conceptualization of daily activities of a person's private life (ADLs, see Section 2). Therefore, HCM-L is based on activity theory as proposed by Leontyev [43]. All HCM-L concepts and their semantic relationships are expressed using the meta-model published in [6]. Regarding the MCA language framework, HCM-L forms a part of the semantic core. The HBMS knowledge base, i.e. the HCM, is represented using OWL as a representation language. Where needed, OWL expressions are translated into constructs of another representation language.

The modeling tool, *HCM-L Modeler*, has been developed using the meta modeling platform ADOxx® [5]. The tool (available at: http://www.open-models.at/web/hcm-l/download) provides the means for visualizing, analyzing, verifying and validating HCM-L models, in particular the HCM.

4. Approach for predicting human actions

In its first version, the HBMS Behavioral Engine utilized non-probabilistic reasoning for solving its tasks. For the purpose of handling uncertainty, we had to

1. inject probability extensions into the semantic core for HCM-L (the metamodeling hierarchy);
2. convert the resulting extended core to get a PR-OWL representation that can be used by ontology reasoners supporting OWL.

The measures for extending the semantic concepts and representing them by means of OWL2 to be used for reasoning, are shown on Fig. 5: We started from the two separate languages, HCM-L and PR-OWL (1). Then we extracted the probabilistic concepts of PR-OWL from the PR-OWL metamodel and injected them into the HCM-L metamodel to form the resulting extended PR-HCM-L metamodel (2). To provide a representation for PR-HCM-L, we developed a converter [44] to transform the UML-based HCM-L metamodel representation into OWL2 such that the PR-HCM-L metamodel representation is conformant to the (common) OWL2 grammar (3).

The detailed description of this extension process is presented in the next section.

4.1. Extension of the semantic core model by probabilistic information

According to the MCA Language Framework, we distinguish the core concepts (semantics) of the metamodel and the constructs of the representation language. In the case of HCM-L, the metamodel representation language may be UML-based, but may also have a different syntax. In this section, we show how the probabilistic extensions are injected into the core concepts.

Core concepts

Fig. 6 shows the four core concepts of HCM-L and their relationships: (1) *Behavioral Unit* encapsulates the alternative sequences of (2) *operations* (actions) that a person performs in order to achieve the (3) *Goal* of a daily activity. Operations are connected by (4) *Flows* thus forming sequences. Alternative action sequences for reaching a given goal result in branches and joints in the Behavioral Unit.

Context modeling is addressed by the concept *Thing* and its relationships. It describes concrete or abstract objects, features and people, which have a purpose. *Person* represents a person by various information about the psychological and

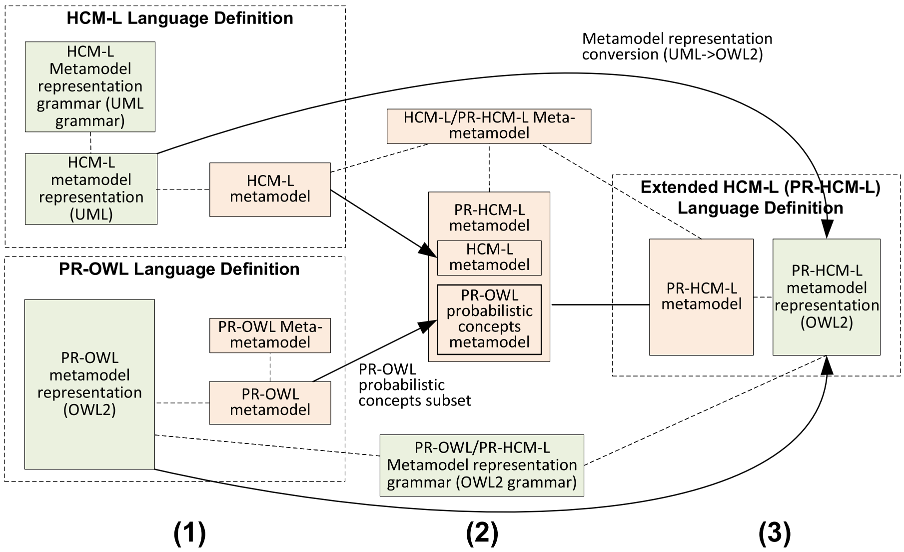

Fig. 5. Extending the Core Semantic Model by PR-OWL probabilistic concepts.

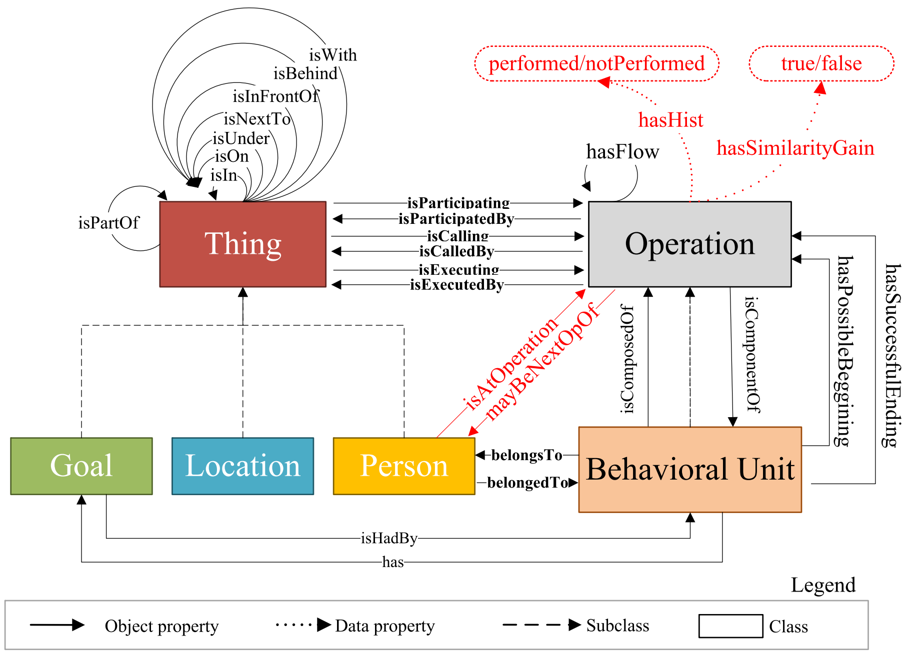

Fig. 6. HCM-L core meta-model and probabilistic extensions.

physical state, modeled as attributes and/or parts of it. *Location* is for the description of rooms and other spaces, facilitating the acquisition of data (temperature, humidity, sound, etc.). Each *Operation* relates to some Thing through three associations: (i) *Calling*, Thing that start operations; (ii) *Participating*, Things that contribute or are manipulated by operations; and (iii) *Executing*, Things that execute operations.

Probabilistic extensions

The probabilistic extensions to the semantic core model were determined by separating the conceptual elements of a probabilistic ontology (implemented in PR-OWL) from their OWL representation, and injecting these elements into the HCM-L core. The resulting set of extensions is depicted on Fig. 6 in red color. It consists of the two attributes, *hasHist* and *hasSimiliarityGain*; and the two relations, *isAtOperation* and *mayBeNextOperationOf*.

These extensions exploited the MTheory presented in Fig. 7 for answering the following question (query): *What is the probability of the user performing a given operation?* The answer is provided by the resident node *mayNextOperationOf(p, op)*

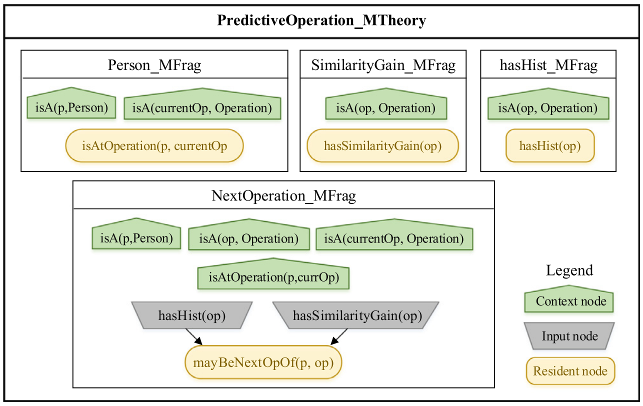

Fig. 7. MTheory for reasoning over uncertainty of operations.

which means: "may be the next operation of". This name was chosen to mention the uncertainty level associated with the relationship between Person and Operation.

An MTheory is a repeatable structure (template) from which SSBNs are generated according to the behavior of the user in a given ADL. Thus, at run time, when there is a need to generate an SSBN to predict a next operation, i.e., when a situation of cognitive decline is detected, the network structure is generated dynamically according to MTheory. This is composed by the four fragments described below whose random variables related to the context nodes are: "p" for Person, "currentOp" and "op" (future operation) for Operation. Resident nodes will be displayed as: $<$*resident node name*$>$ $=$ *states*.

- *Person_MFrag*: represents the probability of the user being stopped at a given "currentOp" within the activity. This fact is represented by the resident node *isAtOperation (p, currentOp)* = {*True, False*};
- *SimilarityGain_MFrag*: represents the likelihood of "op" being performed in a similar manner by different users. In other words, it is a probabilistic similarity measure based on the Gaussian distribution whose parameters are calculated using the number of times a specific operation is chosen by all the users who performed the same activity. The responsible resident node is as follows: *hasSimilarityGain (op)* = {*True, False*};
- *hasHist_MFrag*: represents the probability of "op" being executed based on the user's history. It can be understood as the historical importance, represented by the resident node *hasHist (op)* = {*performed, notPerformed*};
- *NextOperation_MFrag*: represents the probability of a next "op" being recommended to the user through the resident node *mayBeNextOperationOf (p, op)* = {*True, False*}. The probability distribution of this node is directly influenced by the two previous MFrags that work as priority measures.

In order to generate, according to the MTheory, the structure of the SSBNs dynamically, each resident node of an MFrag must also implement its own local probability distribution, whose values can be reported by a domain expert or can be generated through machine learning algorithms.

Injecting probabilistic extensions into the core semantics

For injecting the probabilistic extensions into the core semantics we followed the aspect-oriented paradigm [45], because of its similar motivation [46]: the concept sets to be combined belong to different concerns of interest for the problem at hand (*separation of concerns* [47]). In our case the concerns are human cognitive modeling and probability modeling. For this purpose we

1. mapped the meta-metamodel of the injected PR-OWL metamodel into the target: the HCM-L meta-metamodel. This mapping was straightforward as originally both meta-metamodels contained a common set of elements which are of interest for our problem, in particular, including metaclasses *Attribute* and *Relation*;
2. determined the set of *extension points* within the HCM-L metamodel, i.e., the metamodel elements where the meta-attributes and meta-relations can be injected. In our case, this set consists of *Operation* and *Person*;
3. specified the injections:
   (a) *hasHist* and *hasSimiliarityGain* attributes are injected to belong to *Operation*;
   (b) *isAtOperation* and *mayBeNextOperationOf* relations are injected to connect *Operation* and *Person*.

As a result, we obtained the PR-HCM-L metamodel, graphically represented in Fig. 6.

4.2. OWL2 representation

To obtain a representation of the PR-HCM-L metamodel suitable for inference, we developed a UML to OWL2 converter [44] and used it for converting the concepts originating from the HCM-L metamodel and originally represented by means of UML. In particular:

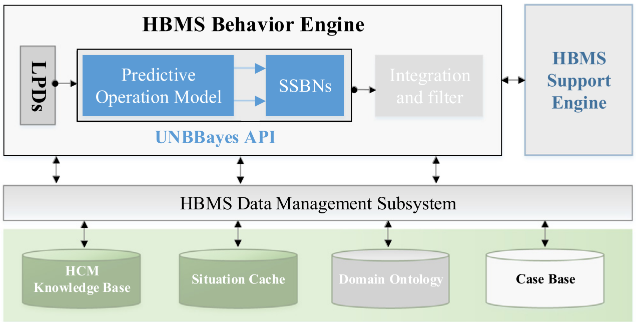

Fig. 8. Probabilistic HBMS behavior engine.

1. the conceptual classes were represented by OWL2 classes with the exception of the *Flow* class described below;
2. the relations between *Thing* (context elements) and *Operation* (action) were represented as three direct-inverse pairs of object properties in OWL2: (*IsParticipating, IsPacticipatedBy*); (*isCalling, isCalledBy*); and (*isExecuting, isExecutedBy*), the relation between *Behavioral Unit* and *Person* resulted in the pair (*belongsTo, belongedTo*), the relation between *Behavioral Unit* and *Goal* - in the pair (*has, isHadBy*).
3. the compositional relation between *Behavioral Unit* and *Operation* were represented by a pair of object relations (*isComposedOf, isComponentOf*), together with the axioms making it a composition. This relationship allows for modeling activities hierachically, i.e., when an operation is not a single step but a sub activity within another upper activity.
4. the *Behavioral Unit*'s attributes *hasPossibleBeginning* (indicates possible start operations) and *hasSuccessEnding* (indicates successful end operations) were represented by two eponymous object properties.
5. the *Flow* class was represented by an object property *hasFlow* with domain and range being *Operation*.
6. the *Connection* class and its *Part-Of* specialization were represented by the object properties *isIn, isOn, isUnder, isNextTo, isInFrontOf, isBehind, isWith* and the transitive propoerty *isPartOf*, respectively.

The concepts and properties originated from the PR-OWL metamodel (*hasHist, hasSimiliarityGain, isAtOperation* and *mayBeNextOperationOf*) did not require the above conversion as they were originally represented by means of OWL2. So they just were connected to the converted HCM-L elements in question.

As a result, we obtained an OWL2 representation of the complete PR-HCM-L metamodel, ready to serve for prediction and inference.

4.3. Support architecture

For exploiting the extended HCM metamodel, the HBMS Behavior Engine (see Section 3) had to be adapted as outlined in Fig. 8. Note that this Figure does not show the complete Behavior Engine functionality, but only the functions affected by our extensions.

The important part here is the Predictive Operation Model that, converted into a syntactic representation, supports tool-based reasoning and inference. Our contribution is to make this model an instance of the PR-HCM-L metamodel represented by means of OWL2.

We defined the Predictive Operations Model using the UNBBayes modeling tool. Within the MCA pattern framework, this tool implements the Modeling Tool pattern.

This model is fed by LPDs, whose probability values can be derived by machine learning algorithms, existing in the Behavior Engine. These algorithms are supported by the data from the Case Base, which contains all the historical sequences of actions performed by the user. Otherwise, LPDs may also be provided by a specialist if no automated learning has been performed.

Then, instances of the current behavior model are retrieved from the HCM knowledge base by the Behavior Engine and aligned with the domain ontology. These instances replace the markers of the probabilistic semantic model, thus generating the SSBNs, to which an usual Bayesian network inference algorithm is applied. SSBNs are generated for each next possible operation relative to the one the user stopped at. Therefore, an integration module merges the SSBNs, selecting the operation that is most likely to be recommended to the user.

Though we developed this approach as a part of the HBMS system, we claim that it is sufficiently generic to be applied to any system based on MCA architectural patterns. For, the key modules (predictive operations model + SSBNs) are based on UnBBayes API and can be accessed from the MCA model handler components.

5. Case study

In this section we present a case study demonstrating the application of the proposed approach.

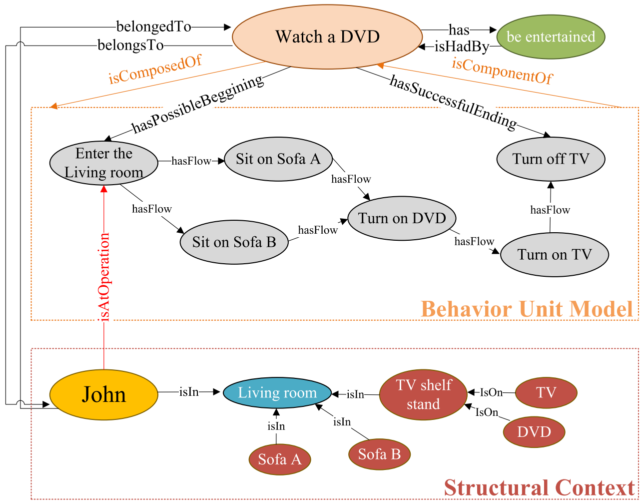

Fig. 9. John's behavior and context models.

5.1. Scenario

The case study starts from a very simple scenario, focusing on a elder person John who suffers from mild cognitive impairments, moderate hearing loss and visual impairment:

John lives in a smart home (equipped with the HBMS system) which consists of a kitchen, a living room, a dining room, a bedroom and a bathroom. In this environment, in the afternoon, John usually watches a movie using the DVD player. To do this, he enters the living room and decides on which sofa he will sit, on sofa A or sofa B. However, John does not always remember which sofa provides better distance and angle for sight and hearing respectively, sofa A, or sofa B. Upon detecting this situation, the HBMS system comes into action recommending John the best option based on his own knowledge (while having been cognitively well). After finding the most suitable sofa John activates the DVD and finally the TV. The activity ends when the TV is turned off. (For simplicity reasons turning off the DVD is not modeled.)

5.2. Applying the approach

Fig. 9 shows a graphical representation of John's former behavior when performing the activity "Watch a DVD" learned by HBMS via observation. As such, the "Behavioral Unit Model Watch a DVD" is a MOF M1 instance of the metamodel concept "Behavioral Unit" presented in Fig. 6. This Behavioral Unit Model is directly related to the "Structural Context (Model)" which again is an instance of the particular metamodel concepts (we used different colors for better readability). The relations between each Operation and the context elements as well as the deficiencies of John were suppressed from the diagram for readability purposes, but they exist.

The user support process is initiated when a deviation in John's behavior is detected by the HBMS Behavior Engine such that John might not reach his initially recognized goal to watch a DVD in a best way (i.e., when John hesitates or choses a suboptimal Sofa). For this purpose the HBMS Behavior Engine compares the time window spent when the operation is performed normally with the current time spent. In the current scenario, John stops at the "Enter the living Room" operation for some time, which suggests that he has a cognitive problem in deciding upon what to do next. Thus, the LPDs are informed to the probabilistic semantic model that generates the SSBNs for each possible next operation, "Sitting on Sofa A" and "Sitting on Sofa B", as shown in Fig. 10, to estimate the chance of recommending one or another to John. The values of each LPDs were obtained from the HBMS dataset (see the next section).

The values in Fig. 10 are explained using the *"Sit on Sofa B"* operation; the same logic applies to the *"Sit on Sofa A"*. Thus, the LPD for the resident node *hasHist(sitOnSofaB)* means that the operation was performed 67% of the time (*performed*) and not performed 33% of the time (*notPerformed*). On the other hand, the LPD for the resident node *hasSimilarityGain(sitOnSofaB)* means that the operation was performed in a similar way (*True*) in 80.8% of the cases and performed in a non-similar way (*False*) in 19.2% of the cases (assuming that John lives with a person who shares the execution of some activities). Finally, the LPD for the resident node *mayBeNextOpOf(John, sitOnSofaB)* describes how the *hasHist* and *hasSimilarityGain* values influence the inference of probabilities for the *"Sit on Sofa B"*. Thus, if

- the operation is performed well and in a similar way, it will have a 73.9% chance of being recommended and 26.1% of not being recommended (line 3). If it is performed well and in a non-similar way, then the chance of recommendation falls to 43% and that of non-recommendation rises to 56.9% (line 5);

Sit on Sofa A

hasHist(op)	hasSimilarityGain(op)
performed=.33, notPerformed=.67	true=.65, false=.35

mayBeNextOpOf(p, op)

```
01. if any op have (hasHist = performed)[
02.   if any op have (hasSimilarityGain = true)
03.     [true =.51, false =.49]
04.   else
05.     [true =.34, false =.66]
06. ] else [
07.    if any op have (hasSimilarityGain = true)
08.     [true =.34, false =.66]
09.    else
10.     [true =.49, false =.51]
11. ]
```

Sit on Sofa B

hasHist(op)	hasSimilarityGain(op)
performed=.67, notPerformed=.33	true=.808, false=.192

mayBeNextOpOf(p, op)

```
01. if any op have (hasHist = performed)[
02.   if any op have (hasSimilarityGain = true)
03.     [true =.739, false =.261]
04.   else
05.     [true =.431, false =.569]
06. ] else [
07.    if any op have (hasSimilarityGain = true)
08.     [true =.431, false =.569]
09.    else
10.     [true =.261, false =.739]
11. ]
```

Fig. 10. LPDs for each possible next operation.

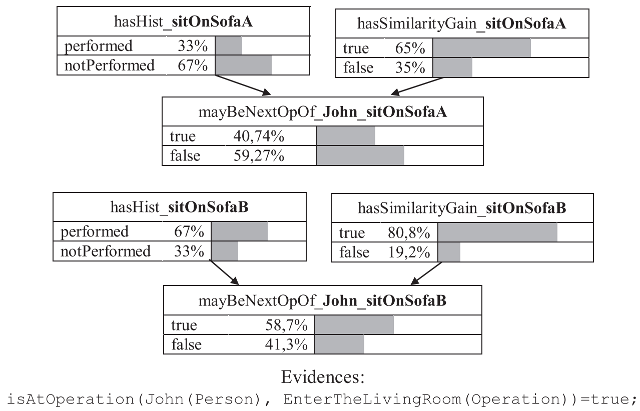

Fig. 11. The resulting SSBNs for each possible next operation.

- the operation is performed poorly and in a similar way, it will have a 43.1% chance of being recommended and 56.9% of not being recommended (line 8). If it is performed poorly and in a non-similar way, then the chance of recommendation drops to 26.1% and that of non-recommendation rises to 73.9% (line 10);

With this, and from the evidence *isAtOperation* = *true* that determines the operation of the activity in which John is stopped, two queries, one for each possible next operation, are performed: **(A)** *What is the probability that John will sit on the sofa A?* **(B)** *What is the probability that John will sit on the sofa B?* The result of these queries is expressed by the *mayBeNextOpOf* node, shown on Fig. 11 in the SSBNs for the prediction and subsequent recommendation of the most appropriate operation. The upper SSBN refers to the query (A), whose probability of recommendation is 40.74%. The lower SSBN refers to the query (B), whose likelihood of recommendation is 58.7%. Therefore, the recommended operation for John would be *"Sit on Sofa B"*, delivered to the most appropriate device e.g. as a warning sound when the wrong sofa is chosen.

6. Evaluation

In this section we describe how the evaluation was conducted and discuss the results. The test aims to measure the performance of the proposed approach, checking how well John would be supported. In other words, we verified if the recommended operation *"sit on Sofa B"* would be performed by him if he was not affected by cognitive decline. To do so, we firstly detail our testing environment and the performance measures adopted. Finally, we present and discuss our results.

6.1. Testing environment and dataset

We made our test in a smart home prototype of the HBMS project faithfully represented in Fig. 12. Within this environment, "John's activity" was performed by a volunteer 10 times representing John as when he was cognitively well. Thus, real data were collected through sensors installed on the following objects [object name (sensor type)]: (i) Sofa A (sensing); (ii) Sofa B (sensing); (iii) living room (sensing carpet); We also used the following IoT devices: (iv) TV (IoT switch); and (v) DVD (IoT switch). These sensors and devices basically trace the actions of the activity counting how many activations were done. This resulted in the HBMS dataset.

The sensors included in the setting were chosen by using different factors which have to be considered when implementing them in real environments. For example, small, cheap, non-intrusive, easy to be installed, compatible with IoT devices.

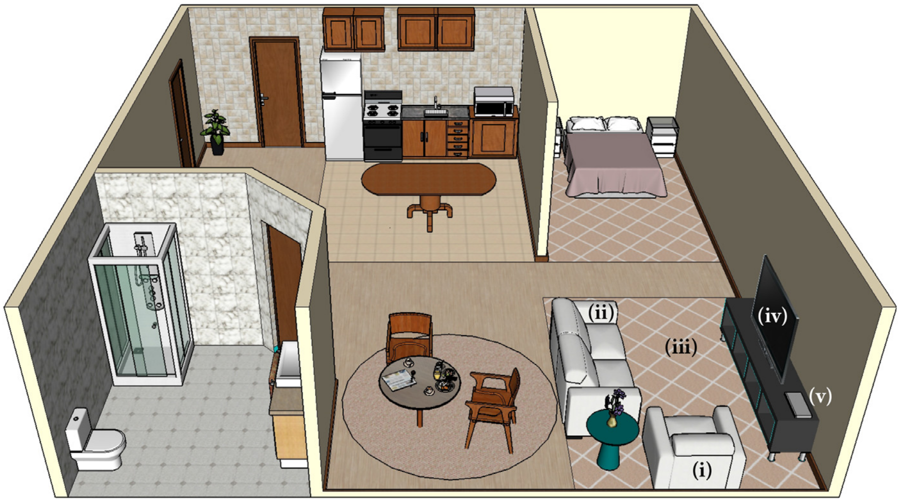

Fig. 12. HBMS Smart Home prototype.

Such requirements offer the possibility to use the proposed approach in real life scenarios. Additionally, due to privacy issues, we did not involve any visual or audio sensors; aside from that, privacy issues were out of scope for this research but will be addressed in future when integrating visual sensors.

During the study, the performed activities were annotated by using an annotation-app on a tablet. For each activity of interest, a separate data channel was created in the database. Every time a predefined activity was started, the corresponding channel was set to 1, and after finishing the activity, the channel was set back to 0. This guaranteed a precise annotation of the performed tasks with their start and end time, that was required for further data analysis and activity classification purposes.

The desired basis format was defined as a file where each row represents the sensor states and annotation values at a specific point of time. Using this base-file, dynamic windows for each activity have been created. This approach allows for a dynamic definition of the size of the needed dataset to enable its usage in various fields of application.

Although there are well known international datasets for AAL, such as CASAS (available at: http://ailab.eecs.wsu.edu/casas/datasets.html), they do not fit the system requirements proposed here due to: (i) lack of data that offer choices (between operations) to train the probabilistic semantic model; and (ii) too much concern in providing data to test activity recognition algorithms. Therefore, we had to create and use an appropriate (HBMS) dataset which now can be used by other approaches for comparison purposes.

6.2. Results and discussion

Initially, we split the 10 samples of the HBMS dataset into two partitions. The first one corresponds to the training subset and the second to the test subset. This partitioning process is known as holdout and consists of dividing a dataset into 70% for training and 30% for testing i.e., in our case, 7 instances for training and 3 instances for testing. We used each training partition to estimate the probability values that feed the probabilistic semantic model making it able to predict and recommend the next possible operation. Thus, we used the test partition to verify the number of times in which the user chose or not the correct recommended operation, “sofa B”.

However, we have identified a problem with the use of holdout: it is not possible to evaluate how much the performance of an approach varies with the different combinations of instances in its training. Thus, to make the results less dependent on the defined partition, 10 random partitions were established to obtain a mean retention performance, a method called random sampling. We defined the number of partitions based on tests, since, because it was random and based on few data, the sampling could create the same or very similar partitions. Based on these tests, we verified that this problem started to appear while creating more than 10 partitions, thus justifying the upper limit on the number of partitions. Incidentally, this would imply an erroneous increase or decrease in the means of performance obtained. Moreover, it is worth to say that the random partitions were obtained through the Weka software preprocessing module with the random sampling filter.

In other words, we can say that 10 “new” HBMS datasets have been created and, as a consequence, we repeated the process described in the first paragraph 10 times. From each test partition of each “new” dataset we got one confusion matrix used to calculate the adopted performance measures: precision, recall, and F-measure. We chose them given the fact that they are commonly used in tests which involve prediction. Below we present the notion of each one of them:

- *Precision:* A number of times that the user followed the recommended operation, divided by the total number of samples in the test set.

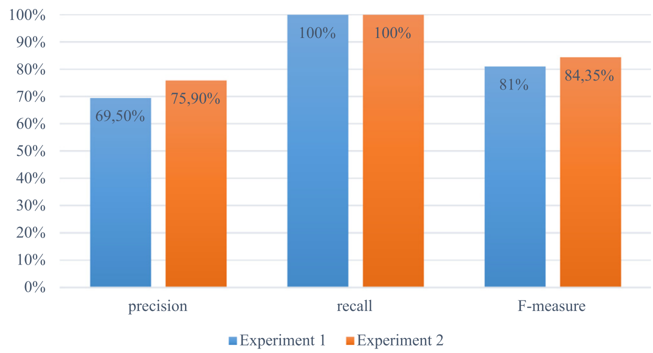

Fig. 13. Experiment results.

- *Recall*: A number of times the user followed the recommended operation, divided by the total number of samples where the user followed the recommended operation.
- *F-measure:* represents the weighted harmonic mean of precision and recall. If the weight equals to 1, then precision and recall have the same degree of importance; this is the setting we use in this work.

Fig. 13 shows the results obtained for metrics mentioned previously for each experiment. Standard deviation values are 18.106; 0; and 16.2 for precision, recall and f-measure respectively.

Therefore, it was possible to conclude that, despite the low amount of data available, the obtained results were promising, with an average precision of approximately 70%. We believe that better results could be obtained if more data were available. Therefore, a new data collection is planned involving more users in the HBMS smart home prototype that will result in a larger dataset. Then, new experiments will be possible. Moreover, we could check that the approach is able to provide support to a user, specially if no data is available to form the behavior model and to estimate probability values. In this case, following the MCA paradigm, such values could be formed by a specialist in the domain, using the user-friendly HCM-L Modeler tool, minimizing the cold-start problem faced by data-driven-only approaches.

7. Related work

Most of the related research that offers some kind of support to a user do it purely through data-driven techniques. As identified through a broad study done by Ranasinghe et al. [21] about activity recognition methods, these techniques (Bayesian Models, Markov Models, Decision Trees, Artificial Neural Networks, Support Vector Machines and others) mostly rely on low-level or raw data. In contrast to data-driven techniques, knowledge-driven techniques (ontology based approaches, logic, or evidence theory) use knowledge provided by an expert. The point is that, in the existing body of work, knowledge-driven techniques are mostly used to support data-driven ones. So, this is the main difference of our work as we provide support to a user totally through knowledge-based means. Below is an overview of the state-of-the art research which is closest in scope to our proposal.

Rashidi et al. [48,49] present CASAS, an integrated set of components that aim towards applying machine learning and data mining techniques to a smart home environment to detect activity patterns, generate automation policies for those patterns, and adapt to the changes in those patterns. Therefore, we cite these works as examples of purely data-driven approaches. In addition, they only address activity recognition tasks and define no clear way for providing the user support like we propose in this work.

Serral et al. [50,51] present a knowledge-driven approach that uses models at runtime to solve the cold-start problem of user behavior automation in a Smart Home, i.e., requiring a large representative dataset to support model training. In particular, they propose a context-adaptive task model and a context model that aids in specifying behavior patterns. They also design and implement a software infrastructure to support the automation of these behavior patterns in the deployed system. On the other hand, our work is different because we do not automate activities but rather assist the user in each action when necessary. Also, their reasoning is based on predefined rules and there is no user-friendly tool for creating the models.

Chen et al. [52] introduce an ontology-based hybrid approach by incorporating data-driven learning capabilities into an approach for modeling ADLs and minimizing the following problems of data-driven-only techniques: (i) cold-start problem; (ii) low model applicability and reusability; and (iii) incompleteness of activity models. However, the authors only consider the activity recondition topic without any prediction aspect related to actions. Additionally, although the ontology is used, there is no reasoning over it. Hence, this work is an example which, unlike our method, only uses knowledge-driven approach as supporting technique and not as its core.

Ceballos et al. [53] use BPMN to model the workflow of daily activities and Bayesian networks to predict actions and assist users. Although this approach looks similar to our work, the authors use a general purpose modeling language (BPMN) that implies its knowledge by the modeler, this may not be true for a caregiver, a doctor, or a relative. In our case, the HCM-

Table 1
A short summary of human support approaches in the frame of Active and Assisted Living (AAL)

Authors	Modeling	Reasoning	Application
[55]	Probabilistic models	Hierarchical Bayesian models	Alzheimer's patients
[56]	Ontology Web Language (OWL)	A Middleware to define reasoning approaches	Taking medications
[54]	Ontology Web Language (OWL)	Multi-Entity Bayesian Network	Pervasive Application for Medicines Management
[57]	Activity Theory (AT) framework	Causal Diagrams (CD)	Medical consultation
[58]	Answer Set Programming	Answer Set Programming and Dempster-Shafer	Cognitive support

L is a DSML specially designed to be used by such people. In addition, the prediction is based on the direct conversion of a BPMN model to a Bayesian network without the use of a probabilistic model, a situation that can bring inconsistencies in the inference process, as the model semantics is lost in such conversion. In our work we aim at preserving the model's semantic meaning in the reasoning phase.

Machado et al. [54] develop an AAL system that monitors a person's activities through sensors and compares the sensor data to a context model in order to figure out the situation, i.e., the current state of the environment. The system uses a probabilistic semantic model for predicting unwanted situations and acts pro-actively to prevent such situations from arising. These proactive actions are executed in form of interactions with the user (e.g. visual and audio warnings, notices to the caregiver) that are triggered when the possibility of an undesired situation is detected. Therefore, this work is directly related to ours, once its prediction is based on a probabilistic semantic model, too. However, we focus on covering every ADL of a user while Machado et al. focus on unwanted situations like incorrect or missed medication, improper handling of gas, fire, electric, water, among others. Table 1 shows a short summary of human support approaches in the frame of Active and Assisted Living (AAL).

8. Conclusion

In this work we presented a set of probabilistic extensions to the HBMS approach that aims, for a while, to go along a person's life in her/his living environment while the person is cognitively well, in order to learn and keep his/her ADLs knowledge. These extensions form a probabilistic semantic model which is subsequently represented in ontological format (by means of PR-OWL conversion) to provide for tool-supported reasoning and inference.

When a mild signal of memory loss starts to happen then the system starts retrieving pieces of learned knowledge of the person's behavior. The ontological representation of the probabilistic semantic model supports this process through reasoning over uncertainty that allow to predict the most likely next operation to support the users i.e. help them to remember what they were used to do. Our proposition was tested in the HBMS smart home prototype where real data was collected trough installable sensors and IoT devices from volunteers who performed common ADLs simulating a person's healthy cognitive state. After that, we used this data to train our probabilistic semantic model and then to test the prediction performance through the holdout process, random sampling and the performance measures. Our results showed that using the probabilistic semantic model to make predictions is promising, specially when no data is available. In this case, a domain expert could form probability values enabling the system to work even without data. This is worth when a user who was not monitored by the system along his life is submitted to the system's guidance. Our contributions are: (i) the probabilistic semantic model which contains additional attributes and relationships allowing to derive assistance to the user without losing semantic meaning; and (ii) the architectural framework allowing for converting the semantic model into a PR-OWL representation to provide for tool-supported probabilistic reasoning in HBMS system.

In future, we intend to refine the probabilistic semantic model with new priority measures to increase the adaptability of predictions, and perform new tests using real case scenarios.

Acknowledgments

The authors would like to thank CAPES, CNPq and FAPERGS Brazilian research agencies as well as the Klaus Tschira Stiftung, Heidelberg, for partially supporting this work.

References

[1] D. Foti, J.S. Koketsu, Activities of daily living, Pedretti's Occup. Therapy: Pract. Skills Phys. Dysfunct. 7 (2013) 157–232.
[2] H. Aloulou, M. Mokhtari, T. Tiberghien, R. Endelin, J. Biswas, Uncertainty handling in semantic reasoning for accurate context understanding, Knowl.-Based Syst. 77 (2015) 16–28.
[3] C. Siegel, T.E. Dorner, Information technologies for active and assisted living—influences to the quality of life of an ageing society, Int. J. Med. Inform. 100 (2017) 32–45.
[4] A.A. Chaaraoui, F. Florez-Revuelta, M. Harbach, A. De Luca, S. Egelman, Technologies and applications for active and assisted living. current situation, Pragmatics 27 (3) (2017) 447–474.
[5] J. Michael, H.C. Mayr, Creating a domain specific modelling method for ambient assistance, in: Proceedings of the Fifteenth International Conference on Advances in ICT for Emerging Regions (ICTer), 2015, 2015, pp. 119–124.
[6] H.C. Mayr, F. Al Machot, J. Michael, G. Morak, S. Ranasinghe, V.A. Shekhovtsov, C. Steinberger, HCM-L: domain-specific modeling for active and assisted living, in: Domain-Specific Conceptual Modeling, Springer, 2016, pp. 527–552.

[7] H.C. Mayr, J. Michael, S. Ranasinghe, V.A. Shekhovtsov, C. Steinberger, Model centered architecture, in: Conceptual Modeling Perspectives, Springer, 2017, pp. 85–104.
[8] J. Michael, C. Steinberger, V.A. Shekhovtsov, F. Al Machot, S. Ranasinghe, G. Morak, The HBMS Story, Enterp. Model. Inf. Syst. Archit. 13 (2018) 345–370.
[9] G.M. Lunardi, G.M. Machado, F.A. Machot, V. Maran, A. Machado, H.C. Mayr, V.A. Shekhovtsov, J.P.M. de Oliveira, Probabilistic ontology reasoning in ambient assistance: predicting human actions, in: Proceedings of the 2018 IEEE 32nd International Conference on Advanced Information Networking and Applications (AINA), IEEE, 2018, pp. 593–600, doi:10.1109/AINA.2018.00092.
[10] D.L. McGuinness, F. Van Harmelen, et al., OWL web ontology language overview, W3C Recomm. (10) (2004) https://www.w3.org/TR/2004/REC-owl-features-20040210/.
[11] K.B. Laskey, MEBN: a language for first-order Bayesian knowledge bases, Artif. Intell. 172 (2–3) (2008) 140–178.
[12] R. Li, B. Lu, K.D. McDonald-Maier, Cognitive assisted living ambient system: a survey, Digit. Commun. Netw. 1 (4) (2015) 229–252.
[13] D. Monekosso, F. Florez-Revuelta, P. Remagnino, Ambient assisted living, IEEE Intell. Syst. 30 (4) (2015) 2–6.
[14] S. Katz, Assessing self-maintenance: activities of daily living, mobility, and instrumental activities of daily living, J. Am. Geriatr. Soc. 31 (12) (1983) 721–727.
[15] M.P. Lawton, E.M. Brody, Assessment of older people: self-maintaining and instrumental activities of daily living, Gerontologist 9 (3_Part_1) (1969) 179–186.
[16] D. Burmeister, B. Altakrouri, A. Schrader, Ambient reflection: towards self-explaining devices, in: Proceedings of the LMIS@ EICS, 2015, pp. 16–20.
[17] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, M. Ayyash, Internet of things: a survey on enabling technologies, protocols, and applications, IEEE Commun. Surv. Tutor. 17 (4) (2015) 2347–2376.
[18] S. Mayer, A. Tschofen, A.K. Dey, F. Mattern, User interfaces for smart things–a generative approach with semantic interaction descriptions, ACM Trans. Comput.-Hum. Interact. (TOCHI) 21 (2) (2014) 12.
[19] S. Mayer, J. Hodges, D. Yu, M. Kritzler, F. Michahelles, An open semantic framework for the industrial Internet of Things, IEEE Intell. Syst. 32 (1) (2017) 96–101.
[20] P.-Y.P. Chi, Y. Li, Weave: scripting cross-device wearable interaction, in: Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems, 2015, pp. 3923–3932.
[21] S. Ranasinghe, F. Al Machot, H.C. Mayr, A review on applications of activity recognition systems with regard to performance and evaluation, Int. J. Distrib. Sens. Netw. 12 (8) (2016).
[22] V.A. Shekhovtsov, H.C. Mayr, S. Ranasinghe, J. Michael, Domain specific models as system links, in: Proceedings of the ER 2018 Workshops, Springer, 2018. Accepted for publication
[23] C. Perera, A. Zaslavsky, P. Christen, D. Georgakopoulos, Context aware computing for the Internet of Things: a survey, IEEE Commun. Surv. Tutor. 16 (1) (2014) 414–454.
[24] Q. Ni, A.B. García Hernando, I.P. de la Cruz, The elderly's independent living in smart homes: a characterization of activities and sensing infrastructure survey to facilitate services development, Sensors 15 (5) (2015) 11312–11362.
[25] G. Abowd, A. Dey, P. Brown, N. Davies, M. Smith, P. Steggles, Towards a better understanding of context and context-awareness, in: Proceedings of the Handheld and Ubiquitous Computing, 1999, pp. 304–307.
[26] N.D. Rodríguez, M.P. Cuéllar, J. Lilius, M.D. Calvo-Flores, A survey on ontologies for human behavior recognition, ACM Comput. Surv. (CSUR) 46 (4) (2014) 43.
[27] P. Moore, B. Hu, X. Zhu, W. Campbell, M. Ratcliffe, A survey of context modeling for pervasive cooperative learning, in: Proceedings of the 2007 First IEEE International Symposium on Information Technologies and Applications in Education, 2007.
[28] O. Yurur, C.H. Liu, Z. Sheng, V.C.M. Leung, W. Moreno, K.K. Leung, Context-awareness for mobile sensing: a survey and future directions, IEEE Commun. Surv. Tutor. (2016) 68–93.
[29] C. Bettini, O. Brdiczka, K. Henricksen, J. Indulska, D. Nicklas, A. Ranganathan, D. Riboni, A survey of context modelling and reasoning techniques, Pervas. Mob. Comput. 6 (2) (2010) 161–180.
[30] Y. Li, J. Chen, L. Feng, Dealing with uncertainty: a survey of theories and practices, IEEE Trans. Knowl. Data Eng. 25 (11) (2013) 2463–2482.
[31] P.C.G. Costa, Bayesian Semantics for the Semantic Web, George Mason University, 2005.
[32] C. Howard, M. Stumptner, A survey of directed entity-relation–based first-order probabilistic languages, ACM Comput. Surv. (CSUR) 47 (1) (2014) 4.
[33] R.N. Carvalho, K.B. Laskey, P.C. Costa, PR-OWL 2.0–bridging the gap to OWL semantics, in: Uncertainty Reasoning for the Semantic Web II, Springer, 2013, pp. 1–18.
[34] P.C. Costa, K.B. Laskey, PR-OWL: A framework for probabilistic ontologies, Front. Artif. Intell. Appl. 150 (2006) 237–249.
[35] S. Matsumoto, R.N. Carvalho, P.C. Costa, K.B. Laskey, L.L. Dos Santos, M. Ladeira, There's No More Need to be a Night OWL: on the PR-OWL for a MEBN Tool Before Nightfall, 2011.
[36] R.N. Carvalho, S. Matsumoto, K.B. Laskey, P.C. Costa, M. Ladeira, L.L. Santos, Probabilistic ontology and knowledge fusion for procurement fraud detection in Brazil, in: Uncertainty Reasoning for the Semantic Web II, Springer, 2013, pp. 19–40.
[37] H.C. Mayr, J. Michael, V.A. Shekhovtsov, S. Ranasinghe, C. Steinberger, A model centered perspective on software-intensive systems, in: Proceedings of the EMISA 2018, CEUR-WS.org, 2018, pp. 58–64.
[38] Object Management Group, Meta Object Facility (MOF) Core, 2016. http://www.omg.org/spec/MOF/, (Accessed August 31.08.2018).
[39] A.G. Kleppe, J. Warmer, J.B. Warmer, W. Bast, MDA Explained: The Model Driven Architecture: Practice and Promise, Addison-Wesley Professional, 2003.
[40] N. Bencomo, R.B. France, B.H. Cheng, U. Aßmann, Models@run.time: foundations, applications, and roadmaps, Lecture Notes in Computer Science, 8378, Springer, 2014.
[41] T. Parr, The Definitive ANTLR 4 Reference, Pragmatic Bookshelf, 2013.
[42] J. Michael, Using cognitive models for behavioral assistance of humans, it-Inf. Technol. 58 (1) (2016) 44–48.
[43] A.N. Leontyev, Activity, Consciousness, and Personality, Prentice-Hall, 1978.
[44] A. Süssl, ADOxx based HCM-L Model transformation to OWL 2.0 (for Reasoning), Master Thesis, Universität Klagenfurt, 2018.
[45] R. Filman, T. Elrad, S. Clarke, M. Akşit, Aspect-Oriented Software Development, Addison-Wesley Professional, 2004.
[46] J. Kienzle, W. Al Abed, F. Fleurey, J.-M. Jézéquel, J. Klein, Aspect-oriented design with reusable aspect models, in: Transactions on Aspect-Oriented Software Development VII, Springer, 2010, pp. 272–320.
[47] P. Tarr, H. Ossher, W. Harrison, S.M. Sutton, N degrees of separation: multi-dimensional separation of concerns, in: Proceedings of the 1999 International Conference on Software Engineering, IEEE, 1999, pp. 107–119.
[48] P. Rashidi, D.J. Cook, Keeping the resident in the loop: adapting the smart home to the user, IEEE Trans. Syst. Man Cybern.-Part A: Syst. Hum. 39 (5) (2009) 949–959.
[49] P. Rashidi, D.J. Cook, L.B. Holder, M. Schmitter-Edgecombe, Discovering activities to recognize and track in a smart environment, IEEE Trans. Knowl. Data Eng. 23 (4) (2011) 527–539.
[50] E. Serral, P. Valderas, V. Pelechano, Improving the cold-start problem in user task automation by using models at runtime, in: Information Systems Development, Springer, 2011, pp. 671–683.
[51] E. Serral, P. Valderas, V. Pelechano, Addressing the evolution of automated user behaviour patterns by runtime model interpretation, Softw. Syst. Model. 14 (4) (2015) 1387–1420.
[52] L. Chen, C. Nugent, G. Okeyo, An ontology-based hybrid approach to activity modeling for smart homes, IEEE Trans. Hum.-Mach. Syst. 44 (1) (2014) 92–105.

[53] H.G. Ceballos, V. Flores-Solorio, J.P. Garcia, A probabilistic BPMN normal form to model and advise human activities, in: Proceedings of the International Workshop on Engineering Multi-Agent Systems, 2015, pp. 51–69.
[54] A. Machado, V. Maran, I. Augustin, L.K. Wives, J.P.M. de Oliveira, Reactive, proactive, and extensible situation-awareness in ambient assisted living, Expert Syst. Appl. 76 (2017) 21–35.
[55] D.J. Patterson, O. Etzioni, D. Fox, H. Kautz, Intelligent ubiquitous computing to support Alzheimer's patients: Enabling the cognitively disabled, in: Adjunct Proceedings, 2002, p. 21.
[56] J.-P. García-Vázquez, M.D. Rodríguez, M. Tentori, D. Saldaña-Jimenez, Á.G. Andrade, A.-N. Espinoza, An agent-based architecture for developing activity-aware systems for assisting elderly, J. UCS 16 (12) (2010) 1500–1520.
[57] H. Ceballos, J.P. García-Vázquez, R. Brena, Using activity theory and causal diagrams for designing multiagent systems that assist human activities, in: Proceedings of the Mexican International Conference on Artificial Intelligence, 2013, pp. 185–198.
[58] F. Al Machot, H.C. Mayr, S. Ranasinghe, A Hybrid reasoning approach for activity recognition based on answer set programming and Dempster–Shafer theory, in: Recent Advances in Nonlinear Dynamics and Synchronization, Springer, 2018, pp. 303–318.