Chapter 11
Using Semantic Markup to Boost Context Awareness for Assistive Systems

Claudia Steinberger and Judith Michael

Abstract Considerable effort to manually configure the user's context and too coarse-grained activity recognition results often make it difficult to set up and run an assistive system. This chapter is the result of our experience with the Human Behavior Monitoring and Support (HBMS) assistive system, which monitors user's activities of daily life and supports the user in carrying out these activities based on his own behavior model. To achieve the required context awareness, we join assistive systems with the semantic web to (1) simplify the construction of a personalized context model and to (2) improve the system's activity recognition capabilities. We show how to semantically describe devices and web applications including their functionalities and user instructions and how to represent these descriptions in the web. The advantages of this semantic markup approach for the application of HBMS-System and beyond are discussed. Moreover, we show how personalized and adaptive HBMS user clients and the power of the context model of HBMS-System can be used to bridge an existing activity recognition gap.

Keywords Semantic manual · Behavioral assistance · Assistive system setup · Application markup · Context model

11.1 Motivation

Assistive systems that monitor user's behavior and support them in performing activities based on behavioral patterns can be very useful in helping people who suffer from any kind of cognitive decline. However, adequate context awareness and a smooth adaptation of an assistive system to the user's environment are prerequisites for its successful installation.

C. Steinberger (✉)
Department of Applied Informatics, Universität Klagenfurt, Klagenfurt, Austria
e-mail: claudia.steinberger@aau.at

J. Michael
Software Engineering, RWTH Aachen University, Aachen, Germany
e-mail: michael@se-rwth.de

F. Chen et al. (eds.), *Smart Assisted Living*, Computer Communications and Networks, https://doi.org/10.1007/978-3-030-25590-9_11

The demands for context awareness are high when it comes to behavioral assistance. These assistive systems can be seen as cyber-physical systems [13]. Every system instance owns its own physical user environment, in which the actions and environmental conditions of the user have to be monitored and supported based on a context model. To record data on situations in the physical world, human activity recognition (HAR) systems are used applying location-, wearable- or object-based sensors or video observation [36]. Assistive systems are not always modular and interoperable with already available open HAR-Systems. They often implement their own HAR-System because the capabilities of existing open ones do not meet their activity recognition requirements.

In any case, the system setup requires adjustment effort for every user in practice, so that the assistive system can achieve the necessary context awareness. Video observation is strongly rejected by users and smart devices with built-in sensors are often not (yet) available in and for households. However, assistive system research projects hardly report on the effort and problems involved in adapting an assistive system and its HAR-Systems to an end user's environment during system setup.

Over the last ten years, the authors have been working on an assistive system project called HBMS (Human Behavior Monitoring and Support) [28]. HBMS-System aims to actively support people in their daily activities including also the use of their preferred web applications. HBMS-System is interoperable with different open HAR-Systems but uses an own internal knowledge base called Human Cognitive Model (HCM), to define, integrate, and store the user's context model [36]. This contribution is the result of the experiences gained in setting up HBMS-System for user's physical environment, observing and supporting user's behavior and shows our solutions to eliminate encountered weaknesses.

An evaluation of HBMS-System revealed in particular the following weaknesses: (1) In order to provide assistance, information about the user's resources in the physical environment like devices or web applications, his or her social and personal situation, and the spatial environment had to be analyzed and mapped to the HCM, when setting up the system. Especially for modeling the devices with their functionalities and user instructions as well as modeling the functionalities and interaction possibilities with web applications, considerable effort was required. (2) The activity recognition results of the used HAR-Systems concerning fine-grained user interactions with devices or web applications were not satisfying for the intended behavioral assistance [40].

In the following, we show the gaps and challenges in the area of context perception in detail and present an approach to overcome these challenges joining assistive systems with the semantic web to simplify the construction of user's context model.

This chapter is structured as follows: Sect. 11.2 presents the context requirements of assistive systems that aim to provide behavior-based support. Section 11.3 introduces the HBMS-System and HCM as well as our evaluation experiences in more detail. It works out two research questions: *RQ1: How is it possible to support the setup of personalized context models for assistive systems in order to keep the manual effort as low as possible? RQ2: How is it possible to improve the fine-grained activity recognition capability of assistive systems to be able to support the user in* *multiple situations?* Section 11.4 deals with related work to answer these research questions. Section 11.5 presents answers to RQ1 and shows how to semantically markup devices and web applications and deals with related advantages. Section 11.6 presents answers to RQ2 and shows how the results of Sect. 11.5 can be used for personalized and adaptive user clients to boost activity recognition results. Section 11.7 summarizes our findings and gives an outlook to identified further research challenges.

11.2 Context Requirements of Assistive Systems

An assistive system designed to support a person in carrying out activities in the physical environment must recognize where the user is doing what, how, and by what means. Therefore, it is necessary that the assistive system has access to a suitable abstraction of the user's context: a personalized context model, which can be used to characterize the user's situation and his interactions with elements of his environment.

At a meta-level [30] different but related context types can be identified that influence the setup process and the ongoing interaction between a user and the assistive system [20]. The manifestation of these context types must be *configured or learned* for every individual user installation during the *setup process of the assistive system* [see (1,2) in Fig. 11.1] and stored as his *personalized context model*. To be able to assist the user in his activities, he is to be *monitored* according to his personalized context model. The *support* opportunities depend essentially on the quality of this context knowledge [see (3, 4) in Fig. 11.1]. The following provides a brief overview

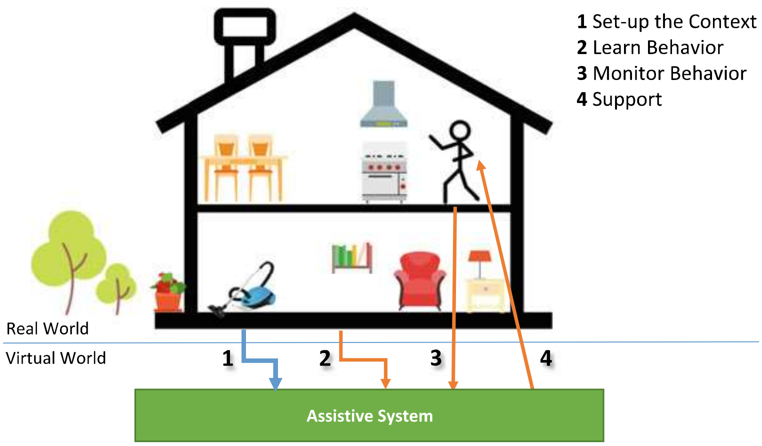

Fig. 11.1 Significance of user's context for assistive systems

of the elements of a context model, their configuration effort, and the options for automatically monitoring user interactions with these elements.

In the real world, a person moves spatially in a certain environment, such as through rooms in an apartment or a house. The user also has special personal characteristics and abilities that can be valuable for the assistive system (e.g., blood pressure, motion profile). When setting up an assistive system, it is necessary to provide the assistive system with a model of the user's *personal* and *spatial context,* including locations, areas and their connections, and the equipment. Since the spatial environment of a user rarely changes, it can be described in the form of a floor plan including also the fixed equipment. Later, during the assistance process, it is relevant to the assistive system where the person is spatially located and how well the person is doing. Vital parameters, the position, and the movement of the person can already be monitored automatically well by HAR-Systems using built-in sensors, wearable sensors, or video observations.

It becomes more difficult for the assistive system to recognize what activity the user is performing *precisely* within his spatial environment. This ability is required both to learn the typical user behavior and provide fine-grained support if he is unable to repeat a previous behavior.

The user's interaction with resources, that make up the equipment of his spatial contexts, such as devices, applications, items, or furniture, can be used as a basis for activity recognition. Setting up the assistive system means that it is necessary to provide a model of the user's *environmental context*, including resources with their interaction possibilities and instructions. As a user can interact with a lot of different, sometimes complex resources performing his activities, the process of model construction can become very time-consuming and tedious. This is especially the case when it comes to model non-smart devices or web applications with all their user functionalities and related user instructions.

Later, during the *assistance process*, it is relevant for the assistive system to recognize what and when interactions between user and resources really take place. Only smart devices are able to provide information on their current status to the assistive system via a HAR-System. The interaction with non-smart devices still causes problems or can hardly be grasped or monitored automatically at all.

In the following, we show how HBMS-System fulfilled these context requirements. We present the context meta-model of the HBMS-System, our approach to set up personal context models in HBMS-System and derive enhancements based on our experiences regarding configuration effort and activity recognition.

11.3 The HBMS-System and HCM

Over the last ten years, the authors have been working on an assistive system project called *Human Behavior Monitoring and Support* (*HBMS*) [28]. It aims to actively support people in their daily activities (e.g., morning procedure) including also the use of their preferred web applications (e.g., e-Banking procedure). Giving support means in HBMS to help people to remember how they once performed a particular activity by reactivating already existing memory anchors. A memory anchor is a simple stimulus that influences the state of mind. By providing retrieval cues, it is easier to remember experienced situations (cued recall) [41]. Thus, HBMS supports the autonomy of a person with decreasing memory.

In HBMS, this support functionality is based on conceptualizing a person's episodic memory, establishing a model of that knowledge, and using that model for support. As described in Sect. 11.2, human actions take place in a physical environment. Therefore, the episodic memory of a person refers to an abstraction of his or her physical environment and user support affects again the physical environment. The personal user context is described in HBMS in the form of a suitable context model, the *Human Cognitive Model* (*HCM*), building the core of the HBMS-System (see Fig. 11.2).

The user's episodic knowledge (behavior context) as well as information about user's environmental resources, social and personal situation, and location is used and contained in the HCM. Figure 11.3 shows a simplified meta-model of HCM. Particularly, high demands are placed on the description of the personal environmental context, including devices, household appliances, and (web) applications used [27]. The HBMS-System was realized as a model centered architecture [24]. A personalized user context model is set up in the HBMS-System in the form of models, each of which is formed tool-assisted (HCM-L Modeller) with the means of the domain-specific modeling language *HCM-L* (*Human Cognitive Modeling Language*) [23, 25]. It preserves knowledge in human-readable as well as in computer-readable representation form.

As shown in Fig. 11.2, when setting up the HBMS-System (1) information about the assisted person and his social context and (2) information about the user's envi-

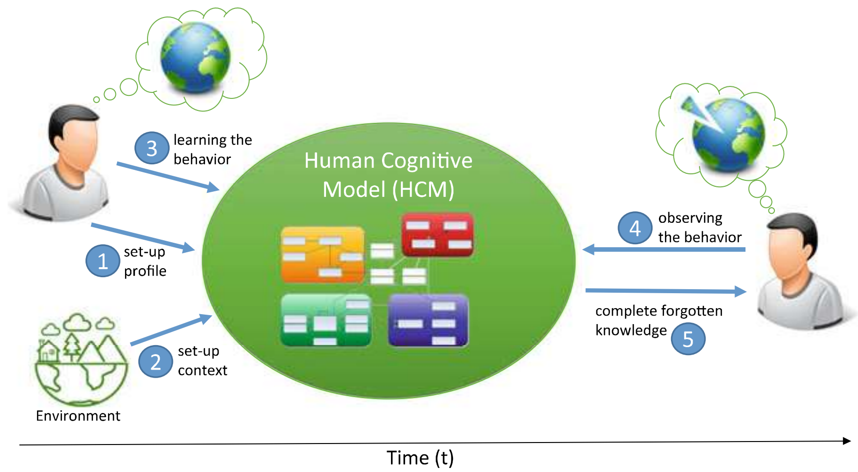

Fig. 11.2 Overall idea of the HBMS-system

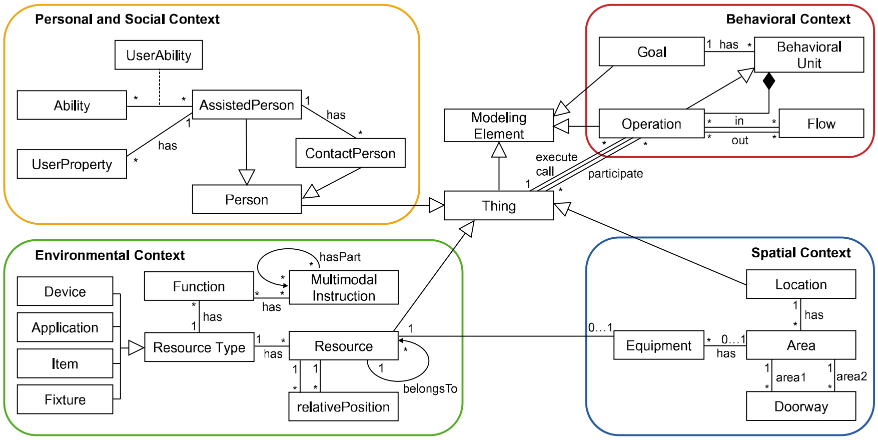

Fig. 11.3 HCM-L meta-model (excerpt from [27])

ronment and spatial context have to be added. The HBMS-System can *learn* the typical behavior (*individual episodic memory*) of the person by manual setup or by observation (3). HBMS-System is interoperable with different open HAR-Systems [36]. On demand, the HBMS-System uses this knowledge to *compensate gaps in the episodic memory* of the supported person by (4) *observing* the behavior of a person and (5) providing smart *advice and support*. Reasoning over the HCM allows the HBMS-System to predict user actions and to guide him [1].

The first evaluation of HBMS-System revealed in particular the following weaknesses:

(1) When setting up the HBMS-System (steps 1 and 2 in Fig. 11.2), the user's context has to be modeled manually or taken from the used HAR-System to build up the user's personalized context model. Particularly, considerable effort is needed to model user's resources and their functionality and handling like for household appliances and other devices [40] and web applications, as this data is not available via conventionally HAR-Systems. Thus, large parts of user manuals (instructions on how to operate the resources) have to be *captured manually during system setup* (step 2 in Fig. 11.2). This leads us to research question 1:

- *RQ1: How is it possible to improve the setup of personalized context models for assistive systems in order to keep the manual effort as low as possible?*

(2) The HBMS-System provides interfaces for conventional HAR-Systems to be able to integrate information about users' activities from various systems [36]. This activity information is transformed into the personalized context model (a) during the learning phase (step 3 in Fig. 11.2) to train behavior which should be supported later on and (b) during the support phase (step 4 in Fig. 11.2), where ongoing behavior is monitored step-by-step. The activity recognition results of

the used HAR-Systems concerning fine-grained user interactions with devices or web applications are not satisfying for the intended behavioral assistance [40]. For example, to pick the cutlery out of a cupboard and place it on a table would need several sensors on the cupboard, the cutlery itself and the table to be able to detect it as fine-grained as needed. It would be possible to reach better results by using depth cameras and videos, but the private home environment is too sensitive to use these technologies and end users are hardly willing to accept such solutions [2]. The need for better recognition of fine-grained user interactions with environmental resources (steps 3 and 4 in Fig. 11.2) is therefore still present in HBMS. This leads us to research question 2:

- *RQ2: How is it possible to improve the fine-grained activity recognition capability of assistive systems?*

To answer RQ1, we *investigated if the semantic markup of devices and web application functions can help us to overcome the high effort during HBMS setup.* Figure 11.4 shows the main idea of our approach which simplifies the construction of a personalized context model: Semantic markup data of web applications (e.g., the e-Banking application of the user) and devices (e.g., the vacuum cleaner, washing machine, mower of the user) as additional sources to set up a personalized context model (step 2a in Fig. 11.4). Nevertheless, there remain still things in the environment, such as items or fixture, which have still to be included manually (step 2b in Fig. 11.4).

To answer RQ2, we investigated if the use of *personalized and adaptive user clients can improve the activity recognition capability of assistive systems with regard to the detection of fine-grained activities.* Figure 11.5 shows the main idea of our solution: (4a) designates the sector, where behavior observation already worked well

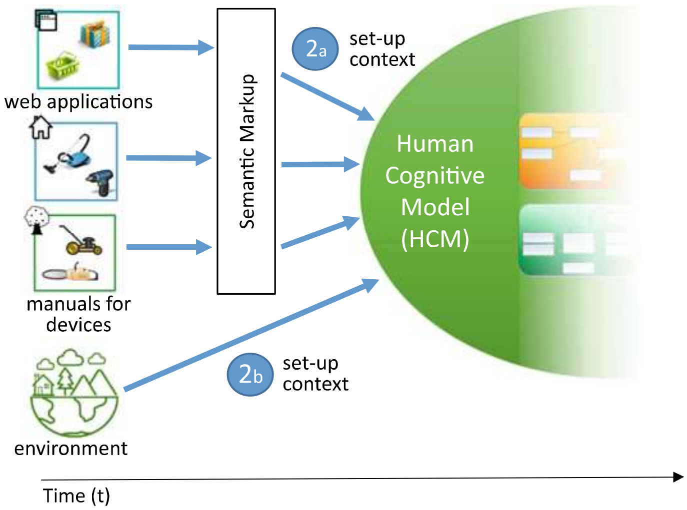

Fig. 11.4 System setup with semantic markups

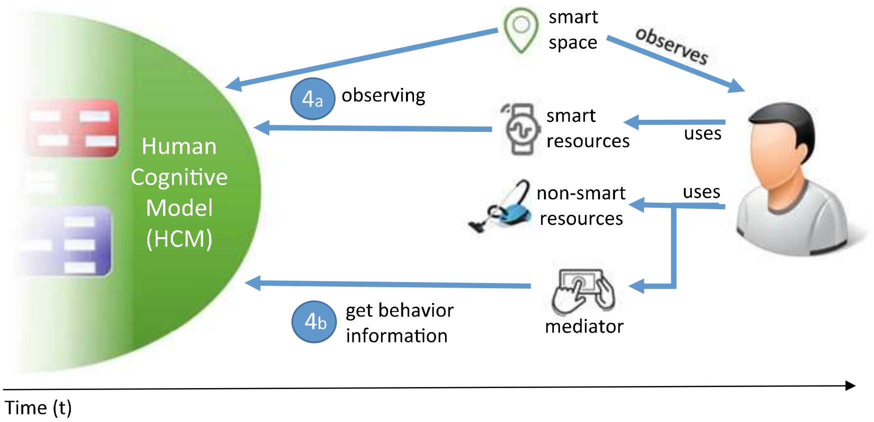

Fig. 11.5 Personalized and adaptive user clients

in HBMS. Information about the movements of a person is observed via the smart space, e.g., sensors which detect where the person is at a certain moment. Information from smart devices such as smartphones or smart watches can be easily included as well. (4b) in Fig. 11.5 shows our extension, where interactions with non-smart devices are communicated via an additional mediator, which is included in a multimodal user client of the HBMS application.

Sections 11.5 and 11.6 describe in detail how we have proceeded in answering these research questions. The next section discusses related work we need as a basis for this work.

11.4 Related Work and Possible Solutions

This section summarizes related work to join an assistive system with the semantic web in order to simplify the construction of the user's context model and to improve its activity recognition capability.

11.4.1 Assistive Systems

The term *assistance system*, as it is understood in this chapter, is intended to provide *just-in-time activity guidance for people as they complete their activities*. A smart home is a residential home setting augmented with a diversity of multimodal sensors, actuators, and devices where this cognitive assistance can be given based on ICT services and systems. By monitoring environmental changes and inhabitants' activities, an assistive system can process perceived sensor data, make timely decisions, and take appropriate actions to assist an inhabitant to perform activities, thus extending their period of time living independently within their own home environment. In their survey [29], Nie et al. summarize the most recent smart home projects, and they give a classification of the main activities considered in smart home scenarios and review the different types of sensors that can be used to monitor these activities. In general, to achieve this objective, a bottom-up, sensor-centric approach is used covering the following levels:

(1) *Monitoring*: Sensors monitor inhabitant behavior and their situated environment in real time and dynamically fuse and interpret the multiple modalities of signals. To monitor inhabitant behavior and environmental changes, visual/audial sensing facilities and networked sensor technologies can be used. Sensors can be attached to an actor under investigation (wearable sensing) for recognizing physical movements or to objects that constitute the activity environment (dense sensing) for recognizing human-object interactions [4, 29].
(2) *Activity Recognition*: To recognize inhabitants' activities, data-driven and knowledge-driven approaches are applied [6]. Chen et al. [4] summarizes activity recognition approaches in correspondence with relevant sensors and monitored activities.
(3) *Assistance*: Provide assistance to help the inhabitant to perform the intended activity based on recognized activities. Rafferty et al. [34] move from a sensor-centric approach for activity recognition to a top-down approach for *intention recognition*, where the inhabitants intended goals are the focus of the assistance system. The HBMS-System applies this approach too.

11.4.2 Semantic Markup and Web Ontologies

The semantic web offers a large variety of technologies for semantic data enrichment and semantic interoperability [3]. Jovanovic and Bagheri [18] summarize some of them in a technology stack for semantically rich e-commerce applications by dividing them into six layers:

1. ***Syntactic interoperability*** by using *standard data-exchange formats* (e.g., JSON, XML, RDF) and *specifications for embedding semantic markup in web pages* (e.g., Microformats, Microdata, RDFa, JSON-LD);
2. ***Schema-level semantic interoperability*** by using *vocabularies* (e.g., Schema.org, Open Graph Protocol, Friend of a Friend);
3. ***Product identity resolution*** by using *strong product identifiers* (e.g., GTIN, UPC, EAN, ISBN);
4. ***Value-level semantic interoperability*** by using *product catalogues/ontologies*;
5. ***Advanced data search and manipulation*** by using *semantic technologies* for these functionalities (e.g., RDF data storage in triple stores, SPARQL data query, and manipulation language);

6. ***Improved e-commerce experience*** by creating intelligent e-commerce applications (e.g., product recommendation apps).

We focus on the *second of these layers*, the *schema-level semantic interoperability*, as it comprises vocabularies for describing things on the web. These vocabularies allow for establishing syntactic and semantic interoperability. Within this layer, several ontologies for marking up things on the web exist in a large range of domains. To solve our research questions RQ1 and RQ2 formulated in Sect. 11.3, we investigated those ones, which seemed to be most promising to describe user functionalities and the handling of devices and web applications semantically.

As a source to identify candidates for our investigations, we used though not only the Linked Open Vocabularies (LOV) catalogue [22] and bundled promising candidates into the following domain categories, showing some similarities with our device and web application domain. The following ontologies have been identified as possible candidates for being a basic ontology for semantically marking up devices, user manuals, and web applications in a first step.

- **Bibliography Ontologies**: *FaBiO* [10] the "Functional Requirements for Bibliographic Records"-aligned Bibliographic Ontology is an ontology for describing publications and that contain or are referred to by bibliographic references.
- **Product and e-Commerce Ontology**: [18] discusses the advantages of using the semantic web for e-commerce. *eClassOWL is used* for describing the types and properties of products and services on the semantic web [9] and is used in combination with *goodRelations* [15], which covers commercial aspects of offers and demand, e.g., prices, payment, or delivery options. *Provoc* [33], as a product vocabulary, extends goodRelations for company hierarchies and production aspects.
- **Internet of Things (IoT) Ontologies**: *IoT-O* [17] is a core domain "Internet of Things" ontology, which is composed of different modules that are based on already existing ontologies [35]. *SAREF* [37] is an ontology in the smart appliances domain.
- **Web Service Ontologies**: *WSMO-Lite* Ontology [44] is a lightweight approach to semantic annotation of web service descriptions [43].
- **Geographic Information Ontologies**: There exists an OWL representation of *ISO 19115* (Geographic Information—Metadata) concerning about ordering instructions [32].
- **Web Resources/SEO**: *Schema.org* provides a single schema for structured data on the web, and its' vocabularies include a large variety of domains (e.g., health and medicine, persons, places, products, organizations, events, actions, creative work media-objects, or recipes) [38]. *Dublin Core Schema* [8] is a small set of vocabulary terms that can be used to describe web resources like video, images, web pages, etc., and physical resources such as books, CDs, or artworks. Schema.org is edging Dublin Core out because it's being solely and specifically created for SEO purposes by actual search engine operators [16].
- **Social media Ontologies**: The *Open Graph Protocol* (*OGP*) [31] allows a basic object description (e.g., Web site, image, music, video) with metadata. OGP is

currently used by Facebook, Google, and Mixi. *Friend of a Friend* (*FOAF*) [11] concentrates on relations between agents (persons, groups, organizations) and documents.

- **Operating instructions**: no specific vocabulary for operating instructions could be identified in *LOV-catalogue*; [42] presents a markup of medical guidelines which is not applicable to our requirements; and the *Learning Resource Metadata Initiative* (*LRMI*) *ontology* [21] is a collection of classes and properties for markup and description of educational resources. The specification builds on the extensive vocabulary provided by Schema.org and other standards.

In [12, 39], we specified the *concepts needed to describe devices and web applications semantically* and *evaluated* the most promising ontology candidates. *Schema.org* seems to be the best approach, due to the concepts overlap, its wide distribution, its extensibility, and the support of search engine optimization. In Sect. 11.5, we show in more detail how to semantically describe devices and web applications, their components and functions, and their handling using the schema.org vocabulary.

11.5 Semantic Markup of Devices and Web Applications

Section 11.3 describes the problems that have occurred with our assistive system HBMS when setting up the user environment and observing interactions with non-smart devices and web applications. We hypothesized that *semantic markup* could help to overcome these problems.

Information regarding the general handling of a device or a web application represents domain knowledge. It is independent of a certain user and usually described by the manufacturer or developer in the form of a user manual.

For devices, manufacturers provide manuals to inform a user how to prepare, use, transport, store, and maintain them. These manuals are typically described in the form of multipage, multilingual, detailed documents, made available for the user by hardcopy or online. Sometimes explanatory videos are additionally offered on social media channels. These manuals are used in the following to describe devices semantically.

In contrast to devices, web applications usually offer context-sensitive user manuals online. Interactive user elements can be automatically identified in HTML code, but the semantics of what actions are triggered when interacting with these elements are not included in HTML code. Schema.org metadata is already used on web pages to mark products, events, organizations, or even some actionable elements, but it is not used to semantically describe the *entire user interface of a web application*.

This section shows how to use schema.org to semantically describe properties, functions, and operating instructions of devices and web applications. A web-based tool called "Schemator" has been developed yet [12] to semi-automatically create appropriate schema.org semantic markups for web applications. The semantic description created by the manufacturer or developer in such a manner can be down-

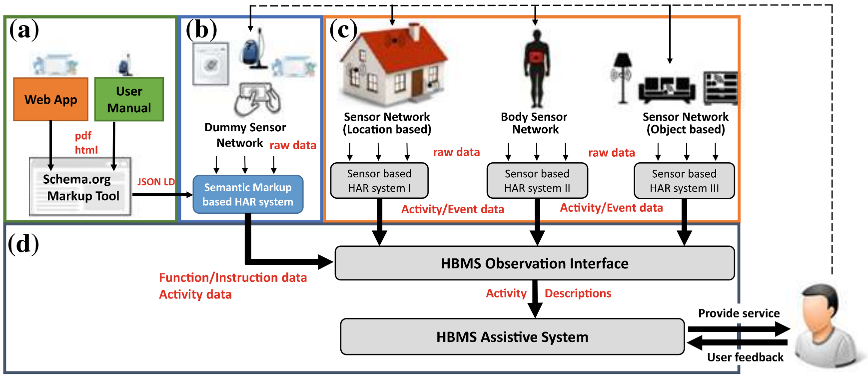

Fig. 11.6 Semantic markup interoperability scenario

loaded from the web in JSON-LD format (see Fig. 11.6a). This enables interoperability with various assistive systems, and the environmental context knowledge needed for a specific user can be imported from the web into the personalized context model of the assistive system.

To improve the activity recognition power and interaction granularity, we propose personalized and adaptive user clients simulating sensor data (see Fig. 11.6b). This approach will be detailed in Sect. 11.6. Via the HBMS Observation Interface, this "semantic based" HAR can be integrated like location or object-based sensor-based HARs [4] (see Fig. 11.6c) to work interoperable with the assistive system and improve its activity recognition capability (see Fig. 11.6d).

11.5.1 Semantic Markup of Web Applications

To semantically describe the interaction possibilities of a user with a web application (see Fig. 11.6a), we examined several web applications and modeled web applications at the meta-level using suitable *schema.org types*. Figure 11.7 sketches the result of this process.

A web application can be described using *WebApplication*. Each page of the web application can be described using *WebPage* and its interactive elements using *WebPageElement*. As all these types are subtypes of *CreativeWork*, properties are available to describe *identifier, name,* and *description*. *MediaObjects* and more specific types like *AudioObject*, *VideoObject,* and *ImageObject* can be linked too using the property *associatedMedia*. What triggers the interaction with a *WebPageElement* can be described using the property *hasPotentialAction* referencing an *Action*. Actions were introduced in schema.org to describe the ability to perform a certain operation on a *Thing*. Each Action has a *name*, a *description,* and a *result*. Latter is of type

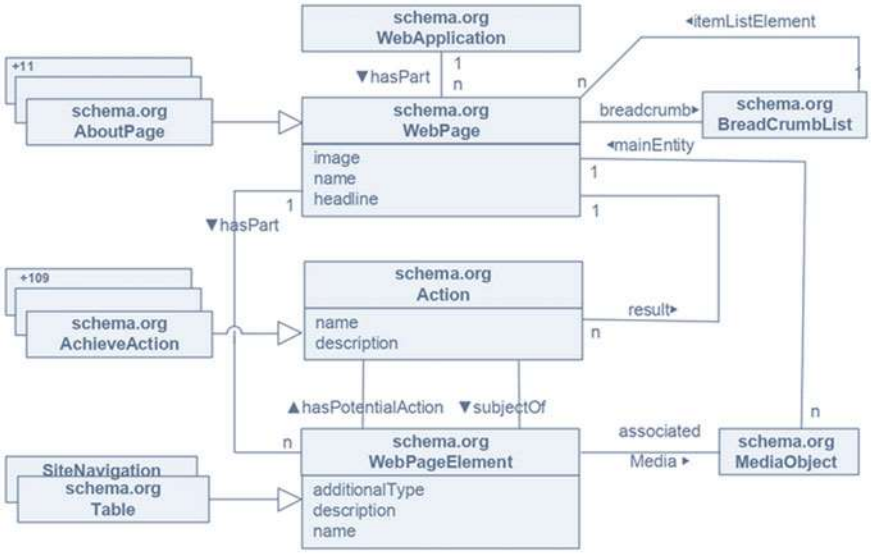

Fig. 11.7 Meta-model for web application markup

CreativeObject (and thus, e.g., again a *WebPage*). Several specific *Action* types like *AchieveAction*, *SearchAction, CreateAction,* and more are predefined in schema.org.

The schema.org markup tool "Schemator" (see Fig. 11.6a) supports generating schema.org metadata of web applications semi-automatically: In a first step, it reads HTML and JavaScript elements of a given web page. In the second step, it provides a user-friendly interface to add semantic information that cannot be generated automatically. Compared to conventional schema.org tools, only the data necessary in this context is requested from the tagging user. The schema.org data of the marked-up web application is centrally stored and can be downloaded afterward in JSON-LD format. Figure 11.8 shows an excerpt of the JSON-LD code describing an example web application.

11.5.2 Semantic Markup Data of Devices

While human readers understand the handling of a good user manual mostly at a glance, an assistive system needs extra information. In this section, we show how to describe user manuals semantically including functions, problem situations, warnings, and instructions.

Although users are most widely interested in device's core functionalities to support them in their activities, also "support" functions (installation, maintenance) have to be carried out and described in manuals. After having examined several manuals, we figured out the needed information for manual's semantic comprehension and

```
"@context"                            : "http://schema.org",
"@id"                                 : "https://developer.mozilla.org/en-US/docs/Web/HTML/Element/html?1",
"type"                                : "WebApplication",
"url"                                 : "https://www.basicwebshopexample.com",
"name"                                : "Basic Webshop Example",
"creator"                             : "John Doe",
"applicationCategory"   : "Webshop",
"about"                               : "This is an basic annotation example",
"hasPart":
{
        "@type"         : "WebPage",
        "@id"           : "https://developer.mozilla.org/en-US/docs/Web/HTML/Element/body?1",
        "hasPart" :
        {
        "@id"           : "https://developer.mozilla.org/en-US/docs/Web/HTML/Element/main?1",
        "@type"         : "WebPageElement",
        "hasPart"       :
        [
        {
        "@id"           : "https://developer.mozilla.org/en-US/docs/Web/HTML/Element/i?1",
        "@type"         : "SiteNavigationElement",
        "name"          : "i.fa.fa-shopping-bag",
        "additionalType": "Brand",
        "description": "The brand logo of this webshop. Click on it to return to the start site.",
        "potentialAction" :
                        {
                        "@type"                 : "Action",
                        "name"                  : "onclick.navigateToHome",
                        "description"   : "Return to start site.",
                        "result"                :
                        {
                        "@type" : "WebPage",
                        "@id"   : "https://developer.mozilla.org/en-US/docs/Web/HTML/Element/body?2"
                        }
```

Fig. 11.8 Excerpt of the JSON-LD code describing a web application

modeled devices, their functionality and associated instructions at the meta-level. As in Sect. 11.5.1 for web applications, we reused schema.org as far as possible for the semantic markup of devices.

Figure 11.9 sketches the result of this process and focuses on those schema.org classes and properties, which are suitable for this purpose. The similarity with the meta-model for the description of web applications in Fig. 11.7 is obvious.

In contrast to the latter, we also used the extension mechanism of schema.org. In the following, we focus on the identified *existing* (shadowed in Fig. 11.9) and extended schema.org classes and properties (white in Fig. 11.9) to semantically markup devices and their handling. As the e-commerce schema from the goodRelations project [15] has been integrated into schema.org in 2012, it is easy to express structured data about products and related facts with schema.org vocabulary. Thus, a device can be easily described as a *Product*, with the properties *name, description, identifier, category, and image*. For the specialization of a product into different models, schema.org offers the subclass *ProductModel* and the property *successorOf* to tag developments of a product over time. Complex devices can be mapped to *Product* as well, as *Product* includes a property *isAccessoryOrSparePartFor* referring to another product (or multiple products) for which this product is an accessory or spare part.

Every *Thing* and thus, also *Product* can have a set of *potentialAction* of type *Action*. This property references an idealized action in which this thing would play an "object" role. There exist many action subclasses, and *useAction* seems to be well suited for tagging **Core Functions of a device**. A fitting *Action* subclass to map

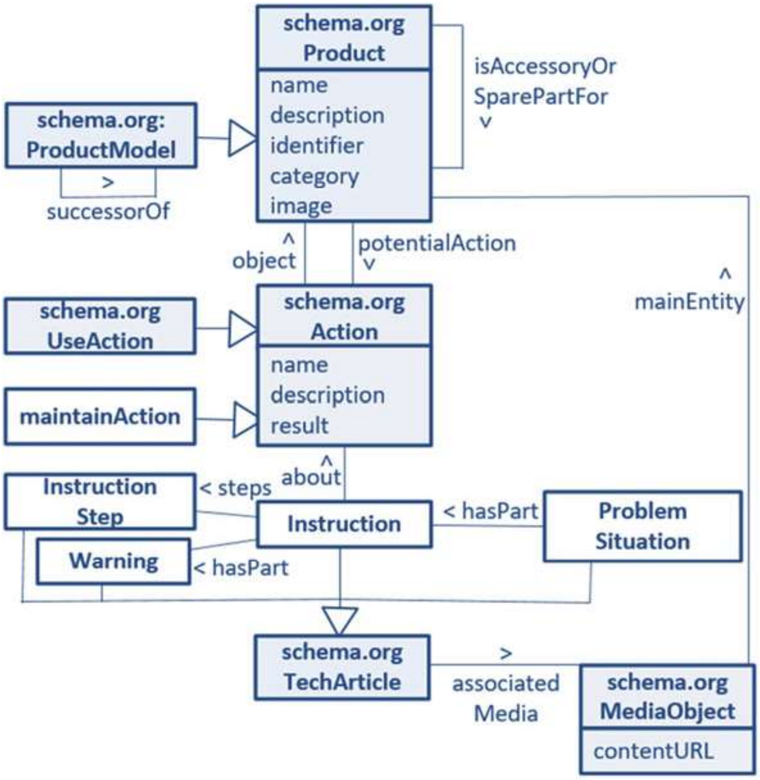

Fig. 11.9 Schema.org excerpt (colored fields) and our extensions (white)

Support Functions has not been defined yet. Thus, a specialization of *Action* to *maintainAction* is helpful to tag them.

Manuals can be considered as creative work. Schema.org offers *CreativeWork* with a lot of subclasses to tag web content like *Article*, *Book*, *MediaObject*, *Movie*, *Recipe*, *Web site,* and more. Although no schema.org subclasses for manuals have been defined yet, *TechArticle*, a subclass of *Article*, can be used to map the concepts of Instruction, Instruction Step, Warning, and Problem Situation, which can be found in almost every user manual. Thus, a specialization of *TechArticle* into *Instruction*, *InstructionStep*, *Warning,* and *ProblemSituation* and the extension of *Instruction* with the property *steps* do make sense using *additionalType.*[1] The existing property *hasPart* allows to indicate other CreativeWorks that are parts of this *CreativeWork*, so it allows to map the interrelationship between *ProblemSituation*, *Instruction,* and *Warning*. *MediaObject* and more specific types like *AudioObject*, *VideoObject,* and *ImageObject* can be associated with *TechArticle* using the property *associatedMedia*.

[1] See, e.g., http://sdo-schemaorgae.appspot.com/TechArticle.

As every *CreativeWork* can have a subject matter of the content using the property *mainEntity*, this property can be used to assign it to a *Product* or *Action* tagged before.

Like sketched in Fig. 11.8 for a web application also the semantic description of a device can be represented in JSON-LD format. During HBMS-System setup, this device knowledge can be imported from the web into the personalized context model.

11.6 Personalized and Adaptive User Clients

This section uses an example to explain how semantic data is integrated into the HBMS-System and how this information is used for active support. As we see in Fig. 11.10, the assumption for the following example is, that manufacturers semantically marked up their devices (e.g., all types of vacuum cleaners) online using the "Schemator" tool (see Sect. 11.5).

During HBMS-System setup and customization, the personalized user context has to be defined (see Sect. 11.3). This covers, e.g., also the description of the functionality and handling of a vacuum cleaner of the supported user. As mentioned, the manual resource definition and update of the environmental context is a considerable effort we wanted to reduce. Thus, the automatic import of the semantic data about the given type of vacuum cleaner into the personalized context model highly facilitates this process.

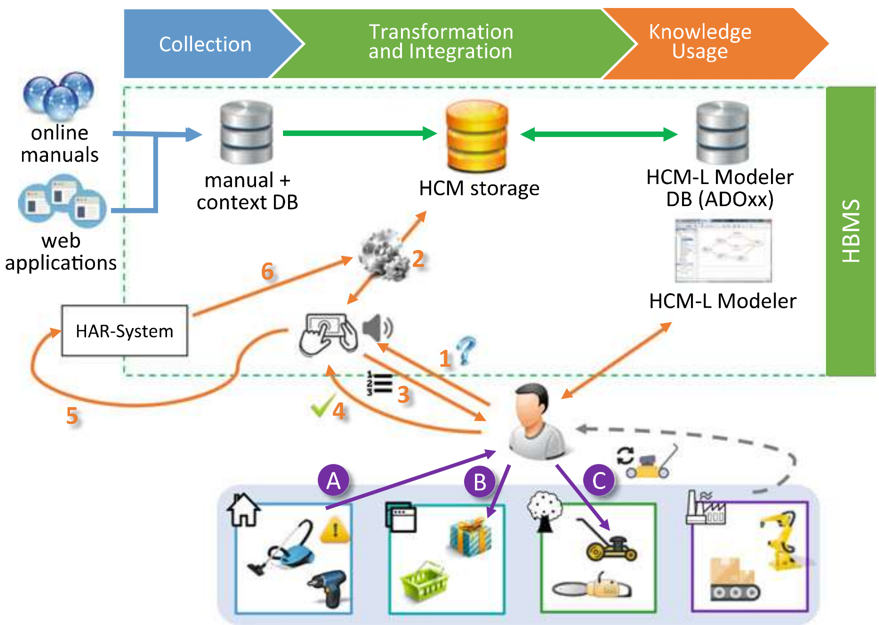

Fig. 11.10 Personalized and adaptive user clients in HBMS

The data is taken from the web, transformed, and integrated into the HBMS-System data stores mapping tagged elements to HCM-L *resource types*, HCM-L *functions* or *multimodal instructions* (see Fig. 11.3). The HCM-L Modeler [1] enables to visualize the imported information.

During the HBMS-support process, the user is guided on demand through behavioral units (BUs) and their operations. These BUs can be indoor or outdoor, at home or in business. We assume in our example the user wants to clean up his apartment and uses his vacuum cleaner.

If one operation of a BU includes the use of a special resource functionality or gives a warning [e.g., (A) in Fig. 11.10, dust bag of the vacuum cleaner has to be replaced], (1) the *user contacts* the HBMS-System, which (2) *retrieves* the demanded *knowledge* regarding this vacuum cleaner and (3) *passes* the *appropriate information* via a client (e.g., a tablet or a voice assistant) to instruct the user step-by-step. After each instruction step, (4) the *user feedbacks* the HBMS-System about the current execution status (e.g., via click or voice assistant). The HBMS-System (5) *simulates sensor data* based on this user feedback and sends it to the related HAR-System. From the point of view of the HAR-System, it seems as if the data had come from a genuine sensor. Thus, (6) *activity data* is handled by the HBMS-System as well as if it were recognized via a real sensor. It is *included* in the HBMS knowledge base and used in succeeding operations. (2)–(6) are now repeated *until the problem is solved* and the warning is gone.

The steps (1)–(6) can also be applied for web applications or web pages [see (B) in Fig. 11.10] which are *semantically marked up*, as the information about the steps needed to use the web application or web page is included in the HCM-storage such as other data.

Moreover, automated mapping from old to renewed devices is possible. If a device is replaced by a new one or a new device is added [e.g., (C) in Fig. 11.10], the update of the manual goes the same "collection, transformation, integration" way until the manual information is contained in the knowledge base and related to BUs using the former device. Consequently, it can be used for support.

11.7 Summary and Outlook

This contribution combines assistive systems with the semantic web using HBMS-System as an example, in order to simplify the creation of a personalized context model during system setup and to improve the system's activity recognition capabilities.

This approach makes interaction possibilities and functions of devices and web applications semantically *understandable* for the HBMS-System and *interoperable* with its environmental context model. We show how to semantically describe devices and web applications including their functionalities, results, and user instructions and propose to use the vocabulary of schema.org. Since schema.org is very extensive, we define a vocabulary subset, but also a small extension of schema.org to fill in gaps. The result is presented in the form of two meta-models. According to these meta-models, devices and web applications can be marked up and represented in JSON-LD format on the web. In this way, "semantic manuals" are created. A tool therefor ('Schemator') is in development.

Furthermore, this contribution explains how *personalized and adaptive* HBMS *user clients* and the power of the *HBMS environmental context* model can be used to *bridge* an existing *activity recognition gap*. User feedback after each instruction step is included in the HBMS-System by simulation of sensor data according to this user feedback. We are currently working on the implementation of the capture, transformation, and integration process in the HBMS-System.

Although we have used a domestic example, the idea of semantic manuals is transferable to other areas such as software applications or machines in production halls and their manuals. Describing industrial information models with ontologies and constraints is an occurring research topic [19]. To provide active assistance for such processes in the industrial context (e.g., for manufacturing processes), semantic markup of *industrial manuals* is a promising approach. In addition, *software applications* could benefit from semantic manuals. Consequently, further development of the HBMS-System will *focus on these domains*.

Even better tools are needed to facilitate markup for manufacturers. A natural language approach that generates the required structured metadata from text documents on the web will be helpful for manufacturers and is another interesting area of research.

Furthermore, the *Internet of Things* (*IoT*) world offers a large variety of open issues for the research community. First adaptations of the context model toward IoT manufacturing processes and privacy concerns are already in progress [26]. This also includes approaches to supporting interoperability through ontologies [7]. Working with heterogeneous intelligent systems in assistive systems remains a challenge where the use of ontologies to standardize and support semantic interoperability would bring great benefits to ongoing projects [5]. Furthermore, the ontological foundation of the HBMS meta-model in a basic ontology [14] by the research team should improve the quality of conceptual modeling languages and models.

References

1. Al Machot F, Mayr HC, Michael J (2014) Behavior modeling and reasoning for ambient support: HCM-L modeler. In: Hutchison D et al (eds) Modern advances in applied intelligence. International conference on industrial engineering and other applications of applied intelligent systems, IEA/AIE 2014, Kaohsiung, Taiwan, proceedings, Part II. Springer International Publishing, Cham, pp 388–397
2. Arning K, Ziefle M (2015) "Get that camera out of my house!" Conjoint measurement of preferences for video-based healthcare monitoring systems in private and public places. In: Geissbühler A et al (eds) Inclusive smart cities and e-health. 13th international conference on smart homes and health telematics, ICOST 2015, Geneva, Switzerland, LNCS 9102. Springer, Cham, pp 152–164

3. Chan LM, Zeng ML (2006) Metadata interoperability and standardization-a study of methodology part I: achieving interoperability at the schema level. D-LIB Mag 12(6)
4. Chen L, Hoey J, Nugent CD, Cook DJ, Yu Z (2012) Sensor-based activity recognition. IEEE Trans Syst, Man, and Cybern-Part C 42(6):790–808
5. Chen L, Nugent CD, Okeyo G (2014) An ontology-based hybrid approach to activity modeling for smart homes. IEEE Trans Hum-Mach Syst (THMS) 44(1):92–105
6. Chen L, Nugent CD, Wang H (2012) A knowledge-driven approach to activity recognition in smart homes. IEEE Trans Knowl Data Eng 24(6):961–974
7. Daniele L, Solanki M, den Hartog F, Roes J (2016) Interoperability for smart appliances in the IoT world. In: Groth P et al (eds) The semantic web—ISWC 2016, LNCS 9982, Springer, pp 21–29
8. Dublin Core Metadata Initiative (DCMI) (2019) URL: goo.gl/gy4P9Q. Last access: 11.3.2019
9. eClassOWL—The Web Ontology for Products and Services (2007) URL: goo.gl/OgEJFg. Last access: 11.3.2019
10. FaBiO (2016) The FRBR-aligned Bibliographic Ontology. URL: goo.gl/CW9QfL. Last access: 9.5.2017
11. Friend of a Friend (FOAF) (2015) URL: goo.gl/PH5YBR. Last access: 11.3.2019
12. Friesser J (2019) Semantische Aufbereitung von Benutzeroberflächen von Webapplikationen für die aktive Assistenz mit HBMS, diploma thesis
13. Geisberger E, Broy M (eds) (2012) agendaCPS: Integrierte Forschungsagenda Cyber-physical systems, vol 1. Springer
14. Guizzardi G, Falbo R, Guizzardi RS (2008) Grounding software domain ontologies in the unified foundational ontology (UFO): the case of the ODE software process ontology. In: M Lencastre et al (eds) 11th Ib. WS on RE and SW Environments, pp 127–140
15. Hepp M (2008) GoodRelations: an ontology for describing products and services offers on the web. In: Gangemi A, Euzenat J (eds) Knowledge engineering: practice and patterns, vol 5268. Springer, Berlin, Heidelberg, pp 329–346
16. Hisham A (2015) Dublin core versus Schema.org: a head-to-head metadata comparison. URL: goo.gl/btkd91. Last access: 11.3.2019
17. IoT-O (2017) a core domain Internet of things ontology. URL: goo.gl/CH1cdX. Last access: 11.3.2019
18. Jovanovic J, Bagheri E (2016) Electronic commerce meets the semantic web. IT Prof 18(4):56–65
19. Kharlamov E et al (2016) Capturing industrial information models with ontologies and constraints. In: Groth P et al (eds) The semantic web—ISWC 2016, LNCS 9982, Springer, pp 325–343
20. Kofod-Petersen A, Cassens J (2006) Using activity theory to model context awareness. In: Roth-Berghofer TR, Schulz S, Leake DB (eds) Modeling and retrieval of context. MRC Workshop 2005, Edinburgh, UK, revised selected papers, LNCS 3946. Springer, Berlin, pp. 1–17
21. Learning Resource Metadata Initiative (LRMI) (2018) URL: goo.gl/APM6lE. Last access: 8.3.2019
22. Linked Open Vocabularies (LOV) (2019) URL: goo.gl/JWhNDU. Last access: 11.3.2019
23. Mayr HC, Al Machot F, Michael J, Morak G, Ranasinghe S, Shekhovtsov V, Steinberger C (2016) HCM-L: domain-specific modeling for active and assisted living. In: Karagiannis D, Mayr HC, Mylopoulos JP (eds) Domain-specific conceptual modeling. Concepts, methods and tools, vol 5. Springer, Cham, pp 527–552
24. Mayr HC, Michael J, Ranasinghe S, Shekhovtsov VA, Steinberger C (2017) Model centered architecture. In: Cabot J et al (eds) Conceptual modeling perspectives. Springer International Publishing, Cham, pp 85–104
25. Michael J, Mayr HC (2013) Conceptual modeling for ambient assistance. In: Ng W, Storey VC, Trujillo J (eds) Conceptual modeling—ER 2013. 32th International conference on conceptual modeling. Hong-Kong, China, LNCS 8217. Springer, pp 403–413
26. Michael J, Koschmider A, Mannhardt F, Baracaldo N, Rumpe B (2019) User-centered and privacy-driven process mining system design. To appear in CAiSE Forum

27. Michael J, Steinberger C (2017) Context modelling for active assistance. In: Proceedings of the ER Forum 2017 and the ER 2017 demo track co-located with ER 2017, CEUR workshop proceedings (CEUR-WS.org), pp 221–234
28. Michael J, Steinberger C, Shekhovtsov VA, Al Machot F, Ranasinghe S, Morak G (2018) The HBMS story. Enterprise modelling and information systems architectures. Int J Concept Model 13:345–370
29. Ni Q, García Hernando AB, Pau de la Cruz I (2015) The elderly's independent living in smart homes: a characterization of activities and sensing infrastructure survey to facilitate services development, Sensors 15(5):11312–11362
30. Object Management Group OMG (2016) Meta Object Facility (MOF) Core, URL: http://bit.ly/OMG-MOF, Version 2.5.1. Last accessed 11.3.2019
31. Open Graph Protocol (OGP) (2014) URL: goo.gl/5lKfLM. Last access: 11.3.2019
32. OWL Representation of ISO 19115 (Geographic Information-Metadata) (2014) URL: goo.gl/ygbQZ7. Last access: 11.3.2019
33. Provoc—Product Vocabulary (2016) URL: goo.gl/HGm3u1. Last access: 11.3.2019
34. Rafferty J, Nugent CD, Liu J, Chen L (2017) From activity recognition to intention recognition for assisted living within smart homes. IEEE Tr. Hum-Mach Syst 47(3):368–379
35. Seydoux N, Drira K, Hernandez N, Monteil T (2016) IoT-O, a core-domain IoT ontology to represent connected devices networks. In: Blomqvist E et al (eds) European knowledge acquisition workshop (EKAW 2016): knowledge engineering and knowledge management, Springer, pp 561–576
36. Shekhovtsov VA, Ranasinghe S, Mayr HC, Michael J (2018) Domain specific models as system links. In: Woo C, Lu J, Li Z, Ling TW, Li G, Lee ML (eds) Advances in conceptual modeling. ER 2018, Xian, China. Springer International Publishing, Cham, pp 330–340
37. Smart Appliances REFerences (SAREF) Ontology (2013) URL: goo.gl/Y93h64. Last access: 11.3.2019
38. Stavrakantonakis I, Fensel A, Fensel D (2014) Matching web entities with potential actions. In: SEMANTiCS 2014—posters & demos track. CEUR
39. Steinberger C, Michael J (2017) Semantic mark-up of operating instructions. Tech. Report. URL: http://bit.ly/semanticMarkup. Last access: 11.3.2019
40. Steinberger C, Michael J (2018) Towards cognitive assisted living 3.0. In: IEEE international conference on pervasive computing and communications workshops (PerCom workshops), Athens. IEEE, Piscataway, NJ, pp. 687–692
41. Strube G (ed) (1996) Wörterbuch der Kognitionswissenschaft. Klett-Cotta, Stuttgart
42. Svátek V, Růžička M (2003) Step-by-step mark-up of medical guideline documents. Int J Med Inf 702(3):329–335
43. Vitvar T, Kopecký J, Viskova J, Fensel D (2008) WSMO-lite annotations for web services. In: Bechhofer S (ed) The semantic web: research and applications. 5th European semantic web conference, ESWC 2008, Tenerife, Spain. Springer, Berlin, pp 674–689
44. WSMO-Lite Ontology (2013) URL: goo.gl/5y5uxo. Last access: 11.3.2019