Business Process Modeling and Instantiation in Home Care Environments

Júlia K. Kambara da Silva, Guilherme Medeiros Machado,
Lucinéia Heloisa Thom and Leandro Krug Wives
Instituto de Informática, Universidade Federal do Rio Grande do Sul (UFRGS), Caixa Postal 15.064, CEP 91.501-970, Porto Alegre, RS, Brasil

Keywords: Home Care, Web Service, Business Process, Modeling, Device Instantiation.

Abstract: There are many studies currently being conducted within the field of Home Care, where houses fulfilled with devices and sensors can help users in their daily lives, even the ones with chronicle diseases and disabilities. One important challenge in this area refers to the selection of the device and functionalities that best meets users' needs based on their context, location and disabilities. In this sense, this paper presents a novel approach for selecting the most appropriate device for the current user context. In our approach, devices and their functionalities are described and represented by Web services, and business processes are used as guidelines that specify procedures that should be taken in the treatment of a home care patient. Therefore, the issue of what device and which of its corresponding functionalities should be selected is treated as an approach to discover and select Web services based on its syntactic and semantic aspects as well as the user context.

1 INTRODUCTION

The world population is currently over 7 billion inhabitants, which 8.2% of these are people over 65 years (WorldoMeters, 2013). Besides, according to the Department of Economic and Social Affairs of the United Nations (2013), the life expectancy is augmenting. While there is no alternative for hospitals in the treatment of patients with terminal illnesses, elderly who are healthy still need assistance to live independently (Pung et al., 2009). Hence, innovative technologies are needed to enable keeping these elderly at their own homes (Bastide et al., 2010).

In this context, Home Care Systems (HCS) emerged. According to McGee-Lennon (2008), HCS can be defined as a technology used to support the accomplishment of networking tasks, providing the means to collect, distribute, analyze and manage information related to care. Such technology typically includes sensors, devices, displays, data, networks and computing infrastructure. Thus, HCS aim to empower people in the need of assistance to continue living in their own homes, while their health and autonomy deteriorates (Auvinen et al., 2011).

In order to efficiently manage home healthcare assistance, a virtual connection between the patient and the hospital has to be established, which supports monitoring activities, requesting services and management of medical protocols that the patient undergoes (Ardissono et al., 2006).

In this sense, Ardissomo et al. (2006) describe that the development of this type of application is far from trivial because the service should be tailored to different actors (patient and/or relatives, nurses, doctors, etc.), and it should integrate distributed, heterogeneous subservices to mediate the interaction between users and service suppliers. Moreover, the service has to cope with medical guidelines in a context-aware way in order to provide users with instructions that are appropriate to the patient's situation.

In this paper, we use business processes (BP) designed with the Business Process Management Notation (BPMN) to describe the guidelines of how the user's smart home could manage undesired events or situations, i.e., those that would lead the users to a dangerous situation or may debilitate their health condition. In this context, a situation is a description of the states of the entities managed by a

K. Kambara da Silva J., Medeiros Machado G., Heloisa Thom L. and Krug Wives L..
Business Process Modeling and Instantiation in Home Care Environments.
DOI: 10.5220/0004886905130525
In *Proceedings of the 16th International Conference on Enterprise Information Systems* (ICEIS-2014), pages 513-525
ISBN: 978-989-758-028-4

system, where an entity is a person, place, or object that is considered relevant to the interaction between the user and an application (Dey, 2001). Therefore, examples of situations include “Patient is sleeping” and “Door is open”.

In our approach, the expression *situation of interest* refers to undesirable situations that can happen to an elderly in a home care context and are relevant to be monitored in our system. Therefore, when a situation of interest happens, the application should react in order to restore its situation to normal.

In this sense, the activities specified by a BP should be performed by home devices. However, in an environment where devices may have different features and are scattered around the house, the challenge is to identify which device should be chosen to perform a particular activity, specially based on the characteristics of the patient. Thus, this is the research question addressed in this paper. In such approach, each device is represented by a Web service, and then the instantiation of such Web services from BP becomes an issue.

To accomplish Web Service’s selection and instantiation, we take into account both the concepts used in semantic Web services’ descriptions and their data types. The idea consists on performing the closest possible semantic and syntactic matching. In addition, we consider contextual aspects, and according to Dey (2001), context is any information that can be used to characterize the situation of an entity.

In this context, the main contributions of this paper are the specification of an ontological model and of business processes models to support a framework that can be used to guide the selection and instantiation devices functionalities in a home care environment. Such selections are based on both the user's context and the syntactic and semantic aspects of Web services, allowing the system to be adaptive to the user. Therefore, houses can incorporate new devices and still manage them accordingly.

The remaining of the paper is structured as follows. Section 2 provides a motivation scenario. In Section 3, we discuss related work. Section 4 shows the underlying concepts used and Section 5 describes our approach. Section 6 indicates how our approach would execute into two fictitious scenarios. Finally, Section 7 concludes the paper.

2 MOTIVATING SCENARIO

Machado et al. (2013) describe a scenario in which applications for home care use pervasive BP. These BP describe guidelines for managing situations of interest. Therefore, when a situation of interest is detected, the system triggers the execution of a business process that addresses such situation. In this sense, smart environments equipped with sensors and actuators represented by Web services (WS) could be managed by a controller that is responsible by the execution of the business processes, therefore assisting the daily life of people.

In this scenario, BP are specified in a conceptual level, and are related to the operational level. Therefore, they must be instantiated and adapted according to the environment of each home, being managed by a controller. This controller requires identifying the most suitable WS to run in every situation of interest. For instance, a house controller could be instructed to send an alert to the patient informing him that it is time to take his medication.

Since in a house there are several devices that could perform such task (e.g., radio, television or mobile phone), and the patient may have specific restrictions (e.g., being deaf), television and mobile phone would be the best options. However, if the patient is not in the same room where the TV or phone is, another device should be selected. The problem in this case becomes the identification and instantiation of any WS that performs the requested functionalities or is compatible with the provided operations. This paper intends to act rightly in this matter, assisting in the identification and instantiation of the WS that performs the activity described by the business process.

3 RELATED WORK

Many works related to Home Care, e.g., Machado et al. (2013), Paganelli and Giuli (2011), address the instantiation of a device to perform a certain action inside a house, but they do not state how it is made.

Works as Ardissono et al. (2006), Bastide et al. (2010), and Gassen et al. (2012) use workflows to describe a guideline to assist in home caring using actions to achieve a certain goal, but do not specify which device should perform a certain action. Other works like the ones of Wang and Turner (2008) and Kaldeli et al. (2013) propose approaches for smart home-based rules, which, when a certain condition is met, an action is performed through a device.

However, these works do not take into account syntactic or semantic WS, i.e., there is no guarantee that the datatypes of Input and Outputs are compatible (syntactic), nor that the service perform the same function (semantic).

4 RELATED CONCEPTS

The approach proposed in this paper is able to identify and dynamically instantiate WS to execute applications based in BP for Home Caring, and it is built upon the following concepts: ontology, which models entities and house devices (also known as appliances); business processes, used for modeling the activities that a controller must perform; and Web services, which are used to communicate with a controllable device and ask them to perform a particular functionality.

4.1 Ontology

An ontology is a formal explicit specification of a shared conceptualization (Gruber, 1993). One of the languages most used for describing ontologies is the Web Ontology Language (OWL). By using OWL we can define the following concepts, according to (Bechhofer et al., 2004) and (Bock et al., 2012): class, also known as concept, promotes abstraction mechanism for grouping resources with similar characteristics; individual (instance of a class); datatype (refers to sets of data values); object property (links individuals to individuals); and datatype property (links individuals to data values), which is also called attribute.

4.2 Business Processes

A business process (BP) consists of a set of activities that are performed and coordinated in an organizational and technical environment, which together aim to realize a business goal (Weske, 2012).

A BP model consists of a set of activity models and execution constraints between them (Weske, 2012). In this context, the Business Process Model and Notation (BPMN) is the *de-facto standard* for representing in graphical way the processes occurring in virtually every kind of organization (Chinosi and Trombetta, 2012).

Moreover, according to Weske (2012), BP can be classified by levels of abstraction into: business goals and strategies, which refers to long-term objectives to the company; organizational business processes, which are typically specified in textual form by their inputs, outputs, expected results, and dependencies on other business process; operational business processes, where activities and theirs relationships are specified by business process models, but implementation aspects of the business process are disregarded; and implemented business processes, which refers to a specification that allows the enactment of the process on a given platform, being it organizational or technical. In this paper we only address the operational level.

4.3 Web Services

Su and Wang (2010) define WS as an interoperable unit of application logic that transcends programming languages, operating systems, communication protocols, network and data representation dependencies and issues. WS are formed by a set of open standards that define how these components should be specified (through WSDL), and how they should be announced for being discovered and reused (via UDDI API), and how they should be invoked at runtime (via SOAP API) (Stroulia and Wang, 2003).

WS standards solve many problems at the technical level, since they describe *how* the service can be accessed, but they do not describe WS's semantics. The description of *what* the service does and in what order its operations have to be called is described in inputs or comments presented (if any) in a WSDL description or UDDI entry (Fensel et al., 2007; Wang et al., 2004).

The Semantic Web Service (SWS) initiative aims to address the problem of WS semantics. Want et al. (2004) define SWS as a description of the skills and content of a WS in a computer-interpretable language.

One of the main description initiatives to enable SWS is the Web Ontology Language Schema (OWL-S), which is an ontology with a framework based on OWL to describe WS. OWL-S describes the inputs, outputs, preconditions, effects and the WS in terms of concepts defined in an OWL ontology.

5 PROPOSED APPROACH

In our scenario, a house has devices, each represented by a WS, and a controller, which is a central system responsible for the management of the house's devices. Therefore, when a situation of interest happens, the controller invokes the suitable set of WS to treat such situation.

Briefly, our proposal is that, given an ontology that describes the controller and devices in a house, this ontology must be used when the modeler is defining the BP that describes the guidelines that will help the patient in a situation of interest. Therefore, when the BP is instantiated and running, the set of WS representing the house's devices should perform any needed action. The instantiation of this WS is made with the aid of the ontology, taking into account the user context.

In many cases, we have to specify the appropriate input parameters before calling a WS. Since WS represent devices in a home, and they are not necessarily made by the same manufacturer, each manufacturer is responsible for creating the corresponding WS for their device. Therefore, two or more WS that do the same thing in different devices may have different input and output parameters, and the most appropriate must be selected. For instance, let us consider the following devices' definitions (Figure 1).

```
Device 1
Service: showMovie
Functionality: "Play Video"
Input: { variable: v,    concept: Video,      datatype: base64Binary
         variable: s,    concept: Subtitle,   datatype: string }
Output: –

Device 2
Service: play
Functionality: "Play Video"
Input: { variable: site,    concept: Link,      datatype: string }
Output: –

Device 3
Service: runMovie
Functionality: "Play Video"
Input: { variable: id,    concept: Movie,      datatype: string }
Output: –
```

Figure 1: Example of the same service in different devices.

Figure 1 shows three devices that could be instantiated to show a Video. In all of them, notice that the input element is formed by a variable, the ontological concept that the variable represents, and its datatype (in SWS). Also notice that the same functionality can have different inputs (to simplify we are focusing only on inputs, but the same may happen to outputs). Thus, how the controller passes these parameters for the WS, since they may change from one device to another? The solution chosen in this paper is that the controller already has a default abstract input set for each action and use syntactic and semantic matching techniques to choose the most appropriated one.

For instance, we could have a URL of normalizedString datatype as the default abstract input to the "Notifies Agitation" action, and the "Play Video" functionality could execute this action. Therefore, all devices that have a service with this "Play Video" functionality would be analyzed through our WS matching technique (see sections 5.1.5 and 5.3.1 for details).

It is important to state that the semantic WS matching is concerned with the distance between the concepts used in of the parameters. It compares the concepts of the available services' input with the concepts of the default abstract service input, i.e. (URL, video) (URL, Subtitle) (URL, Link), (URL, Movie). This is done using the ontology where these concepts were referenced. The syntactic matching however is concerned with the distance between the parameters datatypes, which, in this case, are (normalizedString, base64Binary) and (normalizedString, string).

Using semantic and syntactic matching, one arrives at the result that the *play* service of Device 2 is most similar to the default abstract service given in the previous example, since it has the same number of input arguments and its input concepts (Link and URL) and data types are similar (string and normalizedString).

This allows the house to have various devices made by different manufacturers, but all manageable by the same controller. In our approach, we assume that manufacturers specify these functionalities in the WS description, but one is free to implement them in any manner. So, when a device is purchased and connected into the house, the controller recognizes that a new device was acquired, checks its functionalities, and registers it in its database (these parts are not covered in this paper).

5.1 Home Care Applications Modeling

The models observed for the development of the proposed approach are:

- Person: characterizes a person, its disabilities and location;
- Organization: characterizes the organizations involved (e.g., Health provider);
- Location: represents people and devices' locations in a higher level of abstraction.
- Controller: describes the capabilities of the house's controller;
- Device: describes devices, their features, parameters and restrictions.

Such models compose the base of an ontology, i.e., it is not a complete finished domain ontology, but it can be the basis to a new ontology or be included into another. Such models will be detailed in the following sections and we will use capital letters in the ontology concepts names to distinguish them from their respective real world individuals.

5.1.1 Person

The information presented in our modeling of persons are straightforward, and contains only the characteristics of the users, their disabilities, level and location within the house.

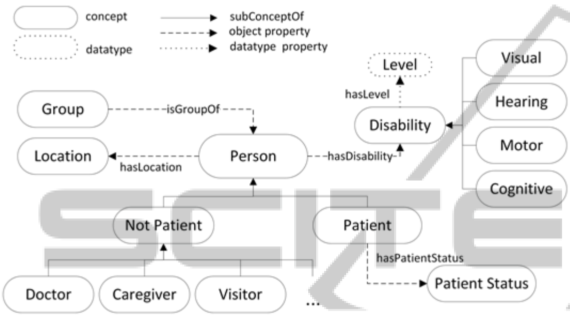

Figure 2: Person model.

As shown in Figure 2, a Person is categorized into several sub-concepts in order to clarify who a particular person is. Person has a Location, which is a place in or out the house (details in section 5.1.3); and Disability, which is an incapacity that the person may have, affecting how the device communicates with her.

Disability has the datatypeProperty Level that indicates the degree of disability, and is subdivided into four categories, taken from (Crow, 2008):

- Visual: people with visual disability may have difficulty understanding written text and graphic content, and both are the main ways for presenting information;
- Hearing: people with this disability have a decreased ability to hear certain frequencies, or have difficulty hearing all frequency levels, which affect the reception of auditory information;
- Engine: people may have limited use of their hands, or cannot use them, which affects their interaction with a device;
- Cognitive: involves a wide range of memory, perception, problem solving and conceptualization of change, this could affect any interaction with the person, since information should be repeated more often.

Although the work of Crow (2008) reports the impact of these disabilities in online learning, it is understood that these disabilities are also important in daily life of person.

Notice that not only Patient may have a Disability, but also of Doctor, Visitor, or other person. Since even a young person can have disabilities. Thus, there is no use sending a sound message to the caregiver if he/she has hearing problems.

Person of Patient type has a Patient Status, i.e., the state in which the patient lies, for instance: agitation, shock, stroke, acute state of diabetes, etc. Finally, Person may be related to the concept of Group, e.g., People are a group of Person.

5.1.2 Organization

Organization models entities that are not part of the house, but somehow interact with the controller, for example, the health care provider, pharmacy, etc.

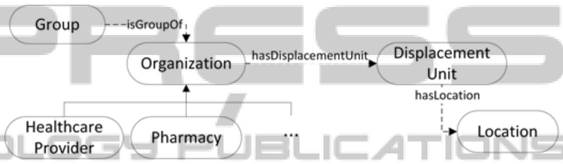

Figure 3: Organization model.

As shown in Figure 3, in the organization model only "Displacement Unit" has Location (that will be more explained in section 5.1.3), since we understand that no organization is located in the patient's home ("Not at Home"), but the "Displacement Unit", because it moves, could be. For instance, if the organization needs to send a unit to the patient's house (e.g., an ambulance), and want to know its location (or if it has already reached the patient's house, e.g. Garden, or not, "Not at Home"). Similarly as in the concept of Person, Organization may be related to a Group.

5.1.3 Location

The Location model (Figure 4) specifies at a high level of abstraction if something is located within the house (At Home) or not (Not at Home), and, if

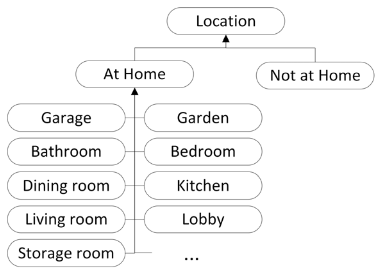

Figure 4: Location model.

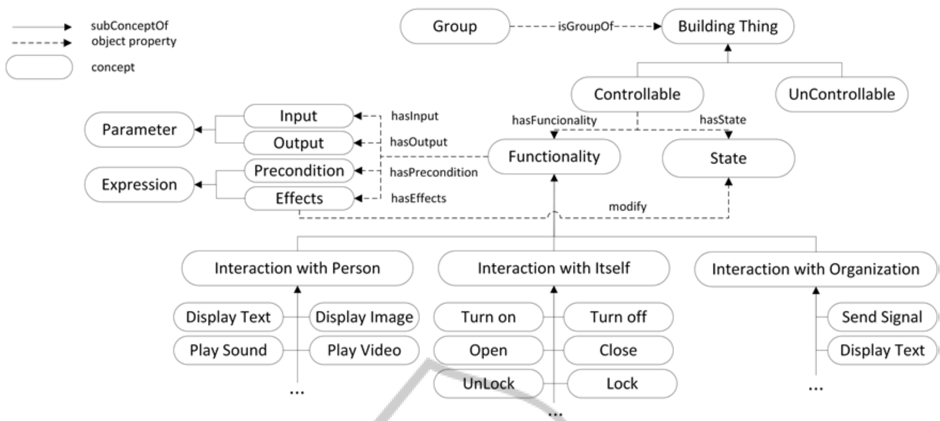

Figure 5: Device model.

inside the house, where specifically it lies. The part that characterizes the inside house location of an entity was taken from the DogOnt ontology (Bonino and Corno, 2008).

5.1.4 Device

The device model (Figure 5) is used both to categorize which built thing is controllable or not, and to report the functionalities of controllable things.

The concepts Building Thing, Controllable, Uncontrollable, Functionality and State were also taken from DogOnt (Bonino and Corno, 2008). Building Thing refers to everything available inside the house. Uncontrollable refers to any object that cannot be managed, and Controllable refers to any object that can be under control. From now on, in order to facilitate reading, all Controllable Things will be simply referred as "Device". Furthermore, in this model, there exists a relationship between the concepts of Group and Building Thing (either controllable or not).

Device has Functionality and State, where State refers to the internal configurations that the device can assume in a time instance, and Functionality refers to what the device can do to change the State values. For example, a lamp can have lamp_state = off, but once its "turn on" functionality is triggered, its status will be changed to lamp_state = on.

Based on OWL-S (Martin et al., 2004), Functionality has zero or more Inputs and Outputs (both are Parameters), besides zero or more Preconditions and Effects, which are Expressions. Input represents an entry concept, which is used for the execution of functionality; Output represents a concept that is generated after the execution of functionality. Preconditions represent rules that must be accomplished in order to perform a functionality, and Effects represents rules that change the status of a device.

Functionality can be classified according to the target object of interaction into the following categories:

- Interaction with Person: when one wants to interact with a person through a device;
- Interaction with Itself: when a device wants to perform some action in itself;
- Interaction with Organization: usually when one wants to communicate / ask something to an entity that is outside the house.

In this model, "Interaction with Person" is a type of Functionality, hence it inherits their relationships (hasInput, hasOutput, hasPrecondition, hasEffects), and thus some specific characteristics of a given functionality may be previously indicated.

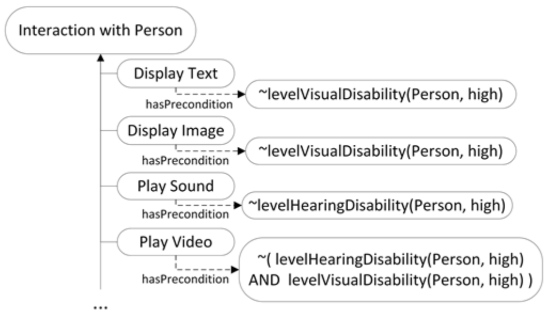

Figure 6: Example of functionality with pre-defined characteristics

As shown in Figure 6, any Device that has "Display Text" as Functionality will have Precondition that has a rule that informs that the targeted person, whose text will be shown (i.e., the target entity), cannot have high visual disability.

5.1.5 Controller

The Controller model is used to indicate the capabilities of a controller and helps the modeler specifying the BP that will perform the more appropriate activities related to each situation of interest. Thus, as shown in Figure 7, the Controller has Action.

An Action is understood as something that is done willingly, executes, or entails something in an intentionally, deliberately and effortful way (Zhu, 2004). In our context, we define action as:

Definition 1: Action is something that an entity executes, does, or performs, either manually or automated.

Thus, in the specific case of the Controller, the actions that can be performed are divided in Sensing and Acting, which correspond to the actions of Getting and Setting (Kaldeli et al., 2013):

Definition 2: Sensing actions are those that return the status of an entity.

Definition 3: Acting actions are those that change the status of one or more entities.

Acting can be further classified into the following categories, considering to whom it is applied:

- Regarding a Person: refer to actions that are performed in order to communicate something to a person;
- Regarding a Device: refer to actions that only modify a device;
- Regarding an Organization: refer to actions to inform/call an organization.

All Actions of Acting type are executed by the Functionality of a device. Thus, when the Acting is "Regarding a Person", it will be executed by a Functionality of "Interaction with Person"; when it is "Regarding an Organization", it will executed by a Functionality of "Interaction with Organization", and when "Regarding a Device", the acting will be its Functionality of "Interaction with Itself" (Figure 7).

Finally, each Acting leaf contains aggregated information so one can choose more than one functionality to perform such actions. Figure 8 illustrates this point: "Notifies Agitation" has Text, Image, Video and Sound, and, so this acting can be executed by any device that has the following functionalities: "Display Text", "Display Image", "Play Sound", or "Play Video". That is, from Information of Acting leafs is created the set of input of default abstract service.

In this model, it is not necessary to provide all the information requested by the Functionality and used by the Acting leaf. However, when it happens, it will affect the functionality that can execute them. For example, the "Calm Agitation" action (Figure 8) does not have all kinds of information (e.g., Text,

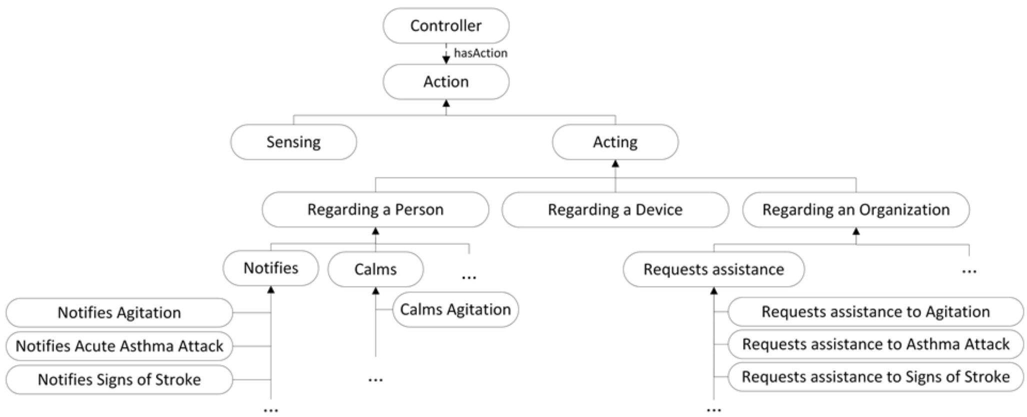

Figure 7: Controller model.

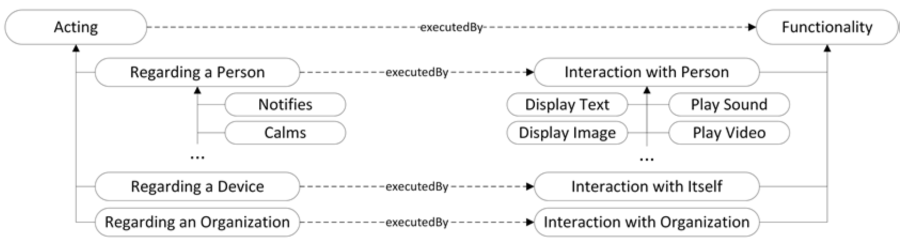

Figure 8: Acting is performed by some kind of device functionality.

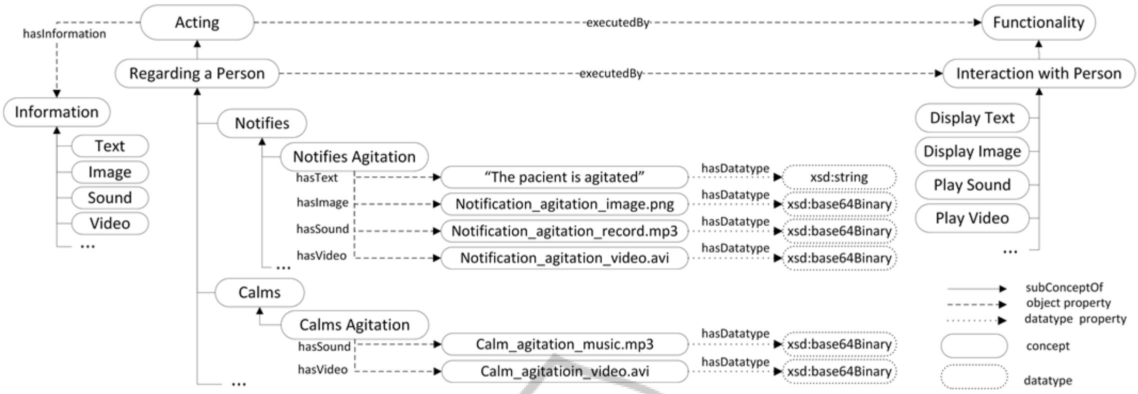

Figure 9: The controller model has information to choose more than one functionality to perform the acting.

Image, Sound, Video), which in this case makes sense because it is unlikely to appease a patient in a state of agitation with a text or a picture. Thus, only the Devices that have the "Play Sound" and/or "Play Video" functionality can perform this action.

5.2 Business Process Modeling

Gassen et al. (2012) propose a methodology in which ontologies are used to support business processes. In their approach, each activity label is composed of a triple <Subject, Action, Object>, where: the Subject is who performs the activity, Action is what is executed by the activity, and the Object is the thing that suffers the action. Subject and Object are related to ontological concepts (Classes), while actions are related to ontological relationships (Object Properties) between concepts. This approach provides more semantics to assist the design and the development of business process models.

Based on the methodology of Gassen et al. (2012), we also propose that each label activity should be composed of a triple <Subject, Action, Object>. However, in our case, every element of the triple must be related to an ontology concept, even Actions, which will have more specific semantics. For instance, in our approach, we are able to add ontological relationships and attributes to an action, and it allows us to perform associations between actions and elements that represent or execute in our scenario.

Figure 12 shows how the proposed approach can be used to model a BP to describe the behavior of agitation in a home care scenario, and it is based on the BP described by Gassen et al. (2012, p.5).

5.3 BP Instantiation

Each element of the activity label triple is here explained. The subject can be multi or single-subject. A multi-subject indicates that all corresponding devices will perform the same action, while single-subject indicates that only one will run it. The action can be classified into sensing and acting. The object can be divided into multi or single-objective. Multi-objective indicates that the action will be performed on a group of goals, and single-objective indicates that it will run on just one goal.

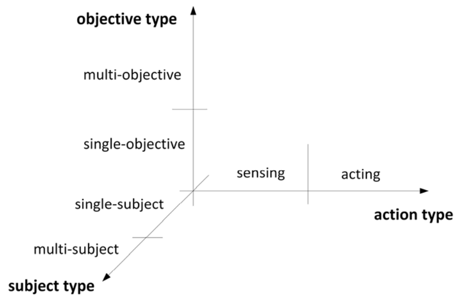

Figure 10: Categorization of the subject, action and objective.

As shown in Figure 10, the type of each element of the label is orthogonal, not interfering in each other's type. For example, the label <Controller, Close, Doors> means that the controller should close all the doors, while <Controller, Close, Door>, indicates that the controller should close only one specific door.

As stated before, we will focus only on Controller Actions of Acting type, i.e., the ones in which the subject labels are of single type, the action labels are of acting type and either single or multi-objective.

Moreover, the controller behavior changes depending to whom the action is concerning. Therefore, we will explain the functioning of the framework for the label cases listed below:

- <single-subject, acting-action, single-objective> regarding a Person;
- <single-subject, acting-action, single-objective> regarding a Device or an Organization;
- <single-subject, acting-action, multi-objective> regarding a Group of Person;
- <single-subject, acting-action, multi-objective> regarding a Group of Device or Organization;

5.3.1 Single-objective regarding a Person

The proposed framework for the instantiation of functions/operations of devices is presented in Figure. It takes into account the disabilities of the user and the location for a single object, regarding a Person. Notice that, in this text, we adopted the notation "DeviceFunc_*m.n*" to indicate the functionality *n* of the device *m*.

According to the framework (Figure 11), when the controller needs to perform a determined action by any of the functionalities available through the house devices, it has to execute the following steps:

a) Filters by person's location and active status: selects only the DeviceFunc that are in the same location or in an environment that is close to where the person is. The target device must be active.

b) Filters by functionality: filters the set returned by (a) by those that can attend the desired action (in this case, as Action is "Related to the Person", the functionalities that can execute the desired action are the "Interact with the People" type).

c) Filters person's disability: filters the set returned by (b) taking into account the disabilities of the target person. This is done attending the preconditions of functionality (e.g., to use the "Image Display" functionality, the precondition is that the level of visual disability of the target person cannot be high)

d) Extracts action as WS: a default abstract WS is defined from the business process and the ontology involved. This step creates one or more default abstract WS that correspond with the number of information that has the specified action. For instance, the "Calms Agitation" action has two types of information (Sound and Video), so it will define two WS, one for each information, i.e., each WS will use such information as input parameter.

e) Orders and filters of devices by relevance: using the default abstract WS defined by the BP, semantic and syntactic matching algorithms are executed (see below) with the DeviceFunc list returned by step (c), which returns a list of DeviceFunc by decreasing order of relevance based on the results of the execution of the matching algorithms and a threshold is made to discard the irrelevances (i.e., those who the controller cannot instantiate). If more than one abstract WS was defined in step (d), the matching algorithms will be executed for each abstract WS, and the result of all will be ordered in the same list. Syntactic matching may be performed by the algorithm proposed by Gao et al. (2002), because it considers datatypes' equivalence and subtype relationship, ensuring that a DeviceFunc can use and return the same datatypes. For the semantic matching, we propose using a modified version of the semantic algorithm proposed by Lin (1998), which, according to Sánchez et al. (2012), is the best for ontology concepts matching. In semantic matching of WS, the algorithm indicates

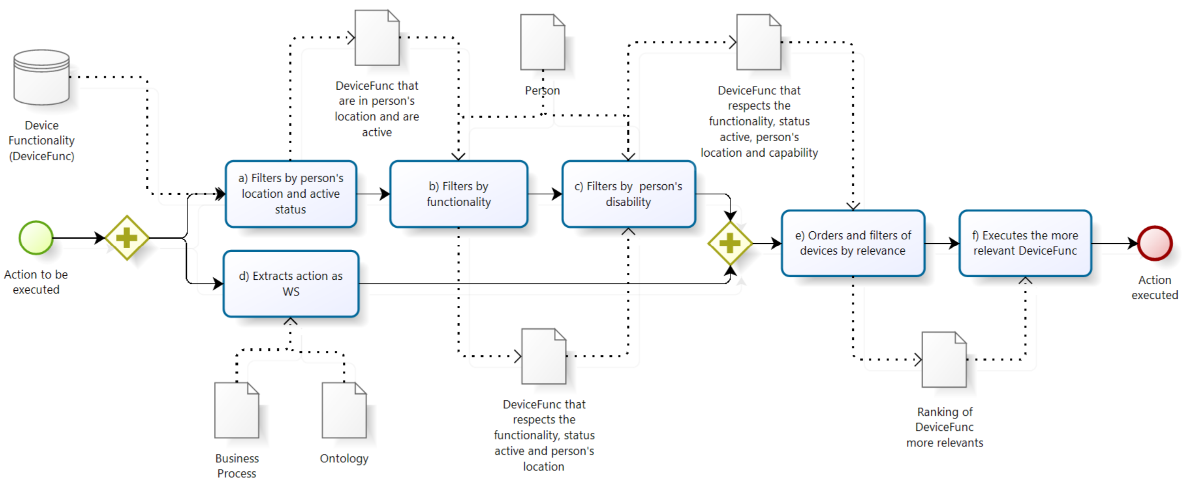

Figure 11: Proposed framework (in BPMN 2.0).

how related are the concepts present in the input and output of a WS operation.

f) Executes the more relevant DeviceFunc: the most relevant DeviceFunc from step (d) is selected and then executed.

Briefly, the framework flow begins when the execution of a given action is requested (*Action to be executed,* Figure 11). The controller then checks which DeviceFunc is available (i.e., has an active status) and are in the same location or close to the target person. After the selection of these DeviceFunc, the ones that can perform this particular action (as the Action is "Related to the Person", the functionality that can execute the action are those that of "Interact with the People" type, and these functionalities are filtered through the preconditions of use they carry) and attend the target user's disabilities are analyzed and then selected. From this list (which contains functionalities that are in the same location as the target person, which are active, can perform the desired action, and respect the disabilities of target person, but it is not known if the controller will be able to instantiate them from the information contained in the desired action), the matching with the default abstract WS is done, which is created from the business process and its ontology. Finally, the controller selects the most relevant DeviceFunc, according to the matching operations.

Thus, through the business process model considering the entities involved (Controller, Person, Device, Organization, Location), it is possible to select the functionality of the devices that are in a location close to a person and that are most suitable to his/her disability, and to define an abstract WS to be used in the matching algorithms (typically comparing how a particular WS is similar to a list of WS, and returning a list ordered by similarity value).

5.3.2 Remaining Label Cases

Other possible combinations are:

- Single-objective regarding a Device or an Organization: when the action regards a Device or Organization, step (c) does not need to execute, since the action is not regarding the Person, so the controller do not need to communicate anything to the Person.
- Multi-objective regarding a Group of Person: here the target objective is a group of persons, so the controller does the same thing as described in previous section, but for all people in the set.
- Multi-objective regarding a Group of Device or Organization: besides skipping step (c), in this case instead of running the DeviceFunc more relevant in step (f), it will run all DeviceFuncs present in the list.

6 APPLICATION SCENARIO

To illustrate our approach, we describe the application of our framework in two distinct fictitious scenarios regarding home care treatment: the first one describes the controller's behavior when the patient, who suffers from Alzheimer's disease, is in a state of agitation, and the second describes the controller's behavior when the patient, who suffers from senile dementia, forgets the stove on.

In these scenarios, we assume that the Patient is an 81 years old man who suffers from Alzheimer's disease and senile dementia, with severe visual disability, mild hearing, and is alone at home; the Caregiver is a woman of 53 years old, without disabilities, which is not at home in any of these scenarios. In Figure 12 and Figure 13 we described each activity following the format <Subject, Action, Object> proposed by Gassen et al. (2012). We also omitted the subject in the label of the BP activity in order to avoid repetition, so it is shown only in the BP pool name.

6.1 Behavior to Manage Patient's Agitation State

In Figure 12, the BP model for dealing with the patient's agitation state is shown. This model was designed using our approach, and it was adapted from (Gassen et al., 2012).

Although BPs usually model all participants involved in the process, we are interested only in the controller participant. In our BP model, the actions specified within the activities of the controller have the following types:

- Sensing: (b), (c) and (f) are sensing actions, since they only observe the status value of an entity, in this case, the patient's location and the patient's level of agitation.
- Acting: (a), (d), (e), (g), (h), (i), (j) are all acting actions because they will change the status value of an entity.

As our approach is focused on the actions of Acting type, we better describe such actions bellow:

- (a): first, it will select all the functionalities of the devices (DeviceFunc) that can run the "Notifies Agitation" action. As shown in Figure 9, this action has four types of aggregate information (Text, Image, Sound and Video). Because of this, only devices that have the following functionalities will be selected: "display text", "display image", "play sound" and "play video". As the caregiver is not in the patient's house, and the only device that has the same patient's location is the mobile phone (and it has all these functionalities), the controller can choose any of these mobile phone's features to notify the caregiver about the patient's state. It could, for example, send an auditory message (play sound), which would alter the status of the mobile phone to playing sound = on and running = true.

- (i) and (j): typically each kind of organization will have only a single mean of communication, what changes is the information.
- (d) and (g): the controller checks which functionalities can execute this action, then verifies the location of the target object, which are the Caregiver and Patient respectively, and selects the DeviceFunc that attend such requirements. In this example, the activity (d) will not be executed since the caregiver is not at home, and we assume that the patient is in a living room and there is a TV around (with functionalities: "display text" , "display image" and "play video") and a Stereo (with "play sound" functionality), but the latter is unplugged. Thus, the controller would choose the TV "play video" functionality, since the patient has severe visual disability, which prevents the execution of "Display Image" and "Display Text" (notice that a video has also sound besides the image shown). Once asked to perform the functionality of the TV "play video", the status will change to playing_video = on and executing = true.
- (e) and (h): to calm the patient, the controller will choose the functionality of the TV "play video" in order to play a relaxing sound to the patient. However it cannot be executed until the device's status changes to playing_video = off and executing = false. In this example, the activity (e) will not be executed since the caregiver is not at home.

6.2 Behavior to Manage the Oblivion of the Stove Turned on

The BP model to handle the oblivion of the stove turned on was taken from the work of Augusto et al. (2006). Again, we separate the activities by their types of actions into the following: Sensing: (a), (b), (c) and (d); and Acting: only (e).

In the model shown in Figure 13, the controller is concerned if the patient or anybody else forgot the stove on. This is presented in the BP activity (a), whose action is of the Sensing type and has People (a Group of Person) as a multi-target object, i.e., it

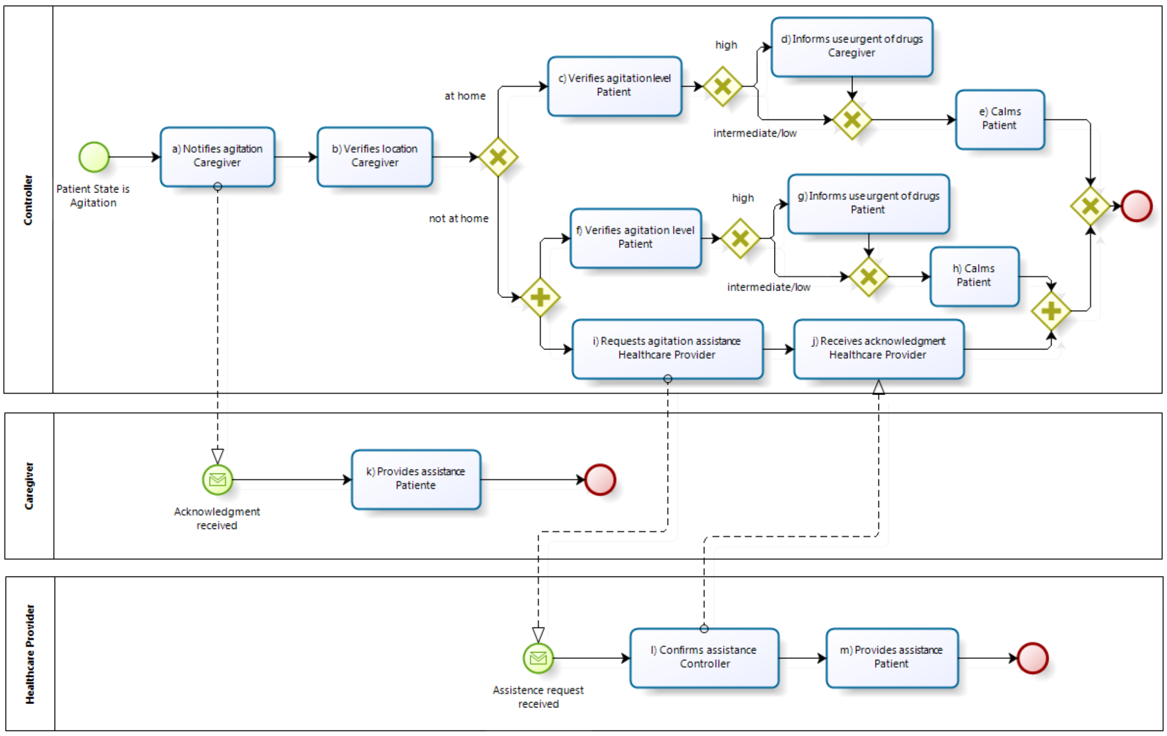

Figure 12: Business process model to take care the state of the patient's agitation (in BPMN 2.0).

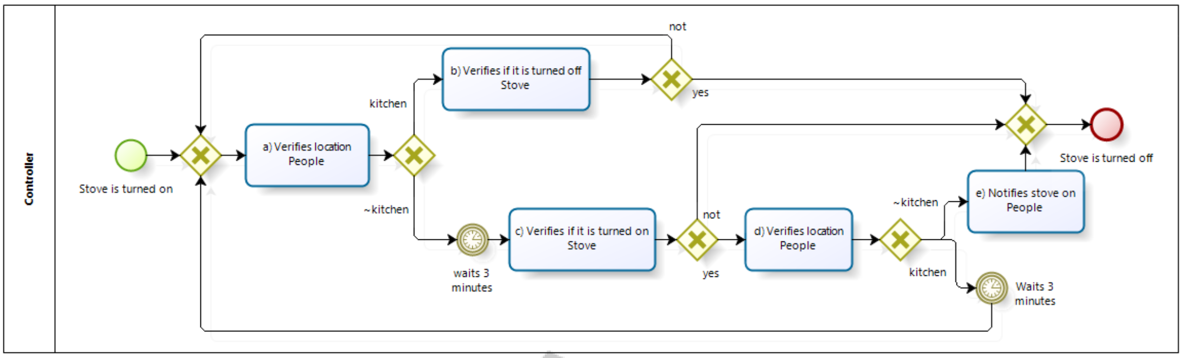

Figure 13: Business process model to notify that the stove was forgotten on (in BPMN 2.0).

will check the location of anyone inside the house. Thus, the action of acting is performed as follows:

- (e): here the target object is also multi; thereby what is expected is that every person in the house is informed that the stove is on. Thus, it is performed according to the following steps: the controller creates a list of DeviceFunc that could execute the "Notifies stove on" action for each person creates a set of DeviceFunc that is in a nearby location and meets its characteristics (disabilities). It uses the DeviceFunc information present in the controller model, and finally the notification is executed for each person.

7 CONCLUSIONS

In this paper we presented a novel approach to address the problem of home care services' instantiation modeled by a BP taking into account contextual aspects. To solve this problem, we proposed a base ontology to describe the central concepts of the domain and their relationship.

Therefore, when modelers are specifying a BP model at the operational level, they will select the corresponding concepts in the ontological model to fulfill the labels of the BP activities. After, when the BP is executing some acting action, the controller looks into the ontological model to verify which functionality can perform such particular action. Based on this, it also verifies if the device that possesses the functionality is closely to the user. Finally, the functionality that meets the user's disabilities is selected and the WS is executed.

As main contributions we can cite the base ontological models for home caring and the framework for providing dynamic instantiation of home devices to perform actions taking into account semantic, syntactic and contextual aspects. Such contributions allow BP models to keep at the operational level, allowing the modeler to be someone from the health care domain, without having to worry about implementation issues. In addition, our approach allows the system to be adaptive to the users, and the house can incorporate new devices and still manage them according to need.

A limitation of our work is that models need to be populated with all possible actions of the controller and device's functionalities in order to allow the modeler to establish the needed ontological concepts relationships with the BP.

As future work, we intend to simulate a patient at home and specify more use cases in the area of home care to validate our approach. We would like also to focus on actions of sensing type, so that the entire controller will be modeled and could be dynamically executed by home devices.

ACKNOWLEDGEMENTS

The authors would like to thank CAPES and CNPq, Brazil, for partially supporting this work.

REFERENCES

Ardissono, L., Di Leva, A., Petrone, G., Segnan, M., Sonnessa, M., 2006. Adaptive Medical Workflow Management for a Context-Dependent Home Healthcare Assistance Service. Electron. Notes Theor. Comput. Sci. 146, 59–68.

Augusto, J. C., Liu, J., Chen, L., 2006. Using ambient intelligence for disaster management. In: Knowledge-Based Intelligent Information and Engineering Systems, pp. 171–178.

Auvinen, A., Silen, R., Groop, J., Lillrank, P., 2011. Defining Service Elements in Home Care. In: Annual

SRII Global Conference. IEEE, pp. 378–383.
Bastide, R., Zefouni, S., Lamine, E., 2010. The homecare digital ecosystem: An information system support architecture. In: 4th IEEE International Conference on Digital Ecosystems and Technologies, pp. 475–480.
Bechhofer, S., Harmelen, F. van, Hendler, J., Horrocks, I., McGuinness, D.L., Patel-Schneider, P.F., Stein, L.A., 2004. OWL Web Ontology Language - Reference [WWW Document]. Online. URL http://www.w3.org/TR/owl-ref/
Bock, C., Fokoue, A., Haase, P., Hoekstra, R., Horrocks, I., Ruttenberg, A., Sattler, U., Smith, M., 2012. OWL 2 Web Ontology Language - Structural Specification and Functional-Style Syntax (Second Edition) [WWW Document]. Online. URL http://www.w3.org/TR/2012/REC-owl2-syntax-20121211/
Bonino, D., Corno, F., 2008. Dogont-ontology modeling for intelligent domotic environments. Semant. Web-ISWC 2008 790–803.
Chinosi, M., Trombetta, A., 2012. BPMN: An introduction to the standard. Comput. Stand. Interfaces 34, 124–134.
Crow, B.K.L., 2008. Four Types of Disabilities : Their Impact on Online Learning 51–55.
Current World Population [WWW Document], 2013. . WorldoMeters. URL http://www.worldometers.info/world-population/
Dey, A.K., 2001. Understanding and Using Context. Pers. Ubiquitous Comput. 5, 4–7.
Fensel, D., Lausen, H., Polleres, A., Bruijn, J. de, Stollberg, M., Roman, D., Domingue, J., 2007. Enabling Semantic Web Services. Springer Berlin Heidelberg.
Gao, X., Yang, J., Papazoglou, M.P., 2002. The capability matching of Web services. In: Fourth International Symposium on Multimedia Software Engineering. Proceedings. IEEE Comput. Soc, pp. 56–63.
Gassen, J.B., Machado, A., Thom, H., Oliveira, P.M. De, 2012. Ontology Support for Home Care Process Design. In: Proc. of the 14th International Conference on Enterprise Information Systems, pp. 84–89.
Gruber, T.R., 1993. A Translation Approach to Portable Ontology Specifications by A Translation Approach to Portable Ontology Specifications 5, 199–220.
Kaldeli, E., Warriach, E.U., Lazovik, A., Aiello, M., 2013. Coordinating the web of services for a smart home. ACM Trans. Web 7, 10:1–10:40.
Lin, D., 1998. An Information-Theoretic Definition of Similarity. In: Proc. of the Fifteenth International Conference on Machine Learning, pp. 296–304.
Machado, A., Pernas, A.M., Augustin, I., Thom, L.H., Krug, L., Palazzo, J., Oliveira, M. De, 2013. Situation-awareness as a Key for Proactive Actions in Ambient Assisted Living. In: Proc. of the 15th International Conference on Enterprise Information, pp. 418–426.
Martin, D., Burstein, M., Hobbs, J., Lassila, O., McDermott, D., McIlraith, S., Narayanan, S., Paolucci, M., Parsia, B., Payne, T., Sirin, E., Srinivasan, N., Sycara, K., 2004. OWL-S : Semantic Markup for Web Services 1 . Introduction : Services on the Semantic Web 2 . Some Motivating Tasks [WWW Document]. W3C. URL http://www.w3.org/Submission/OWL-S/
McGee-Lennon, M.R., 2008. Requirements engineering for home care technology. In: Proc. of the Twenty-Sixth Annual CHI Conference on Human Factors in Computing Systems - CHI'08, pp. 1439–1442.
Paganelli, F., Giuli, D., 2011. An ontology-based system for context-aware and configurable services to support home-based continuous care. IEEE Trans. Inf. Technol. Biomed. 15, 324–33.
Pung, H., Gu, T., Xue, W., Palmes, P., Zhu, J., Ng, W., Tang, C., Chung, N., 2009. Context-aware middleware for pervasive elderly homecare. IEEE J. Sel. Areas Commun. 27, 510–524.
Rong, W., Liu, K., 2010. A Survey of Context Aware Web Service Discovery: From User's Perspective. In: 2010 Fifth IEEE International Symposium on Service Oriented System Engineering. IEEE, pp. 15–22.
Sánchez, D., Batet, M., Isern, D., Valls, A., 2012. Ontology-based semantic similarity: A new feature-based approach. Expert Syst. Appl. 39, 7718–7728.
Stroulia, E., Wang, Y., 2003. Flexible interface matching for Web-service discovery. In: Proc. of the 7th International Conference on Properties and Applications of Dielectric Materials, IEEE Comput. Soc, pp. 147–156.
Su, Z., Wang, X., 2010. An improved approach to rapid discovery of semantic Web service. In: 5th International Conference on Pervasive Computing and Applications. IEEE, pp. 378–381.
Wang, F., Turner, K.J., 2008. Towards personalised home care systems. In: Proc. of the 1st ACM International Conference on PErvasive Technologies Related to Assistive Environments - PETRA'08, pp. 44:1–44:7.
Wang, H., Huang, J.Z., Qu, Y., Xie, J., 2004. Web services: problems and future directions. Web Semant. Sci. Serv. Agents World Wide Web 1, 309–320.
Weske, M., 2012. Business Process Management. Springer Berlin Heidelberg, Berlin, Heidelberg.
World Population Prospects: The 2012 Revision [WWW Document], 2013. . United Nations, Dep. Econ. Soc. Aff. URL http://esa.un.org/unpd/wpp/Documentation/pdf/WPP2012_Press_Release.pdf
Zhu, J., 2004. Passive Action and Causalism. Philos. Stud. 119, 295–314.