Non-Intrusive Context-Aware System for Ambient Assisted Living in Smart Home

Willy Allegre, Thomas Burger, Jean-Yves Antoine, Pascal Berruet, Jean-Paul Departe

A Non-Intrusive Context-Aware System for Ambient Assisted Living in Smart Home

Willy Allègre[1 2], Thomas Burger[1], Jean-Yves Antoine[1],
Pascal Berruet[1], and Jean-Paul Departe[2]

[1] Lab-STICC (CNRS), European University of Brittany, Université de Bretagne-Sud
Centre de Recherche, BP 92116, F-56231 Lorient Cedex, France
E-mail: first.last@univ-ubs.fr

[2] Electronic Lab of Kerpape Functional Reeducation and Rehabilitation Center
E-mail: wallegre@kerpape.mutualite56.fr [*]

Abstract. Automation of smart home for ambient assisted living is currently based on a widespread use of sensors. As efficient as it seems to be, this solution can sometimes be problematic when one focus on user acceptability intimately related to cost and intrusivity. In this paper, we propose a context-aware system based on the semantic analysis of each user request at runtime. Our goal is to infer user data usually sensored by using advanced semantic web tools to adapt home automation services to people with special needs. To take up this challenge, an ontology, automatically derived from a model-driven process, firstly defines user-system interactions. Then, the use of rules allows an inference engine to deduce user location and intention leading to adapted service delivery.

Keywords: Smart Home, Ambient Assisted Living, Ontologies, Model-Driven Engineering, Home Automation Services, Context-Awareness

1 Introduction

Care for dependents is becoming a major social and economic issue for the next few years. In 2050, 30% of people from the European countries will be at least 65 years old [1], thus increasing the number of dependent people with chronic diseases. The number of potential caregivers can not evolve accordingly, so that home automation for ageing and disabled people is considered as an alternative solution to fill the gap, allowing one to facilitate or automate devices activation and to provide services tailored to the user needs: we talk about *Smart Home*. Due to the miniaturization of electronic devices and their dissemination in the environment, smart home is no longer confined to the simple control of household appliances. This field of application is evolving more and more toward *Pervasive Systems* [2], where context-awareness is an important challenge. Indeed, knowing `who` is doing `what` and `where` is essential to adapt a "smart behaviour", especially in the area of *Ambient Assisted Living*. To adapt a house functioning to

[*] This work is financed by the Brittany Region and is the result of a cooperation supported by Kerpape MFRRC (`http://www.kerpape.mutualite56.fr/`).

the user needs, this kind of systems must be able to model and process contextual information (e.g. location, preferences, time markers). Sensors is the most commonly used solution to collect data for home automation [3–6], from which context information and knowledge can be inferred. If this solution is efficient, it can sometimes be problematic when one focus on user acceptability. The use of cameras can be seen as intrusive, whereas the massive deployment of sensors in the environment mainly implies a significant financial cost.

We strongly believe that user data in home automation is still underexploited as the trend in this field is clearly to add extra sensors instead of optimizing the home automation system in place. In the framework of a fully equipped home automation environment, as it is sometimes needed for some wheelchair users or elderly people in loss of autonomy, we also believe that the analysis of control logs (instead of sensor logs) can provide relevant information to adapt system to user needs. Control log is data directly coming from the user remote control whereas sensor log is coming from sensors connected to the home automation bus. This is why analysing the first type of data instead of continuously adding new sensors to gather user data can be seen as a non-intrusive solution (no added devices in the environment). We finally believe that control log analysis provides a new point of view to tackle key challenges in home automation such as real-world deployment, sensor optimisation, but also strategies for dealing with more common issues such as multiple occupancy, sensor failure or interleaved activity.

To take up this challenge, an ontology is defined to support the semantic analysis of each control log. Our goal is to provide the user adapted services without any training steps and using as few sensors as possible. Figure 1 illustrates the basic use of the ontology on a sequence diagram. When the user asks for a service, the control log is processed by the home automation server, but also translated in the context ontology as an assertion. An inference engine apply defined rules on this ontology to enrich the knowledge base by i) higher level information that a third party (occupational therapists, family members, etc.) can analyse to propose new services/rules and ii) services proposition on the user interface based on information previously inferred.

As ontologies and related tools are of great interest for context-awareness in ambient assisted living, mainly due to their interoperability and logic reasoning capabilities, there is a lot of works in this domain[1]. However none of them focused on the analysis of control logs to provide context information about the user. Morevover, very few works provide a design process to ease the ontology deployment which is not a trivial task. In this paper, we describe our non-intrusive context-aware system introduced above as an alternative solution to assist people at home. To do so, we first propose to model user-system interactions and their semantics by means of an ontology (section 2). Then, the use of rules applied to ontology concepts and properties allows an inference engine to deduce user location and intention, leading to service delivery (section 3). In the context of a large scale deployment, a model-driven process is applied for populating the ontology and a software architecture is defined for runtime (section 4). A case

[1] Related works are discussed in section 6 to clarify our contribution.

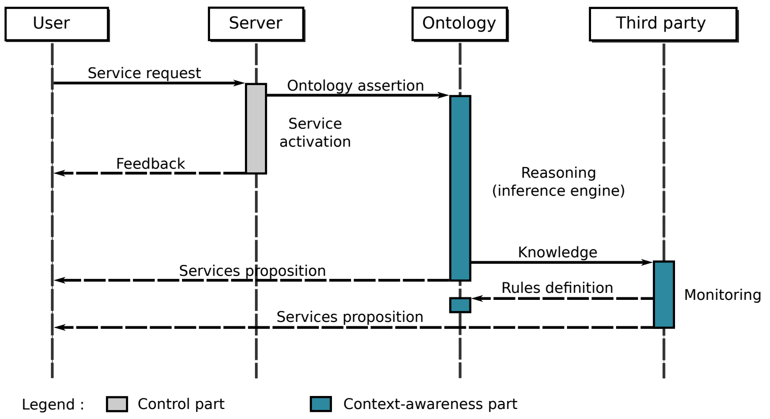

Fig. 1. Basic use of the ontology for the non-intrusive context-aware system

study (section 5) based on a real dataset shows the benefits of the entire work. Finally, related works (section 6) are presented before concluding.

2 An ontology for user-system interaction modeling

The ontology described below models the semantics underlying interactions between one user and his environment. As it is not possible to distinguish who activates a switch, we assume that the user stands alone. Obviously, our work should fit the need of a multi-user application, provided each log can be associated with an individual user id (e.g. services are requested from individual interfaces).

The ontology is made of generic concepts (describing context at a high level of abstraction) which can be applied on pervasive systems (*Upper Class*). Other concepts are responsible for the description of domain specific concepts for home automation (*Specific Class*). As shown in figure 2, the defined ontology is made of six fundamental concepts:

- `User`: someone who interacts with his environment using spread devices
- `Device`: device activated by user i) remotely or ii) from the same room
- `Basic Service`: directly provided by both local and remote devices
- `System`: interface to collect interactions composed of devices and services
- `Location`: part of the environment that can contain devices and users
- `Effect`: effect of services activation on the environment (e.g. luminosity)

These ontology concepts are a useful support for our non-intrusive context-aware system. Indeed, from the way the `User` interacts with his `System`, we seek to infer his `Location` and his intention related to the `Effect` produced

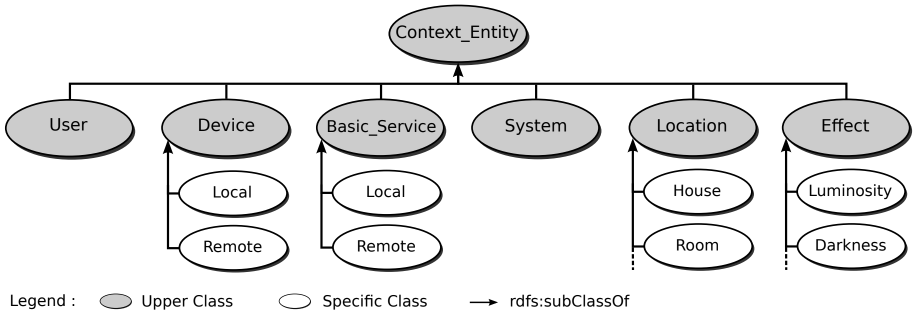

Fig. 2. Excerpt of the ontology taxonomy (*T-Box* with concepts only)

by `Device` activation when he asks for a `Basic Service`. Knowing where the user is and what is his intention, it is then possible to lead to service delivery. This is made possible by the formal semantics of description logics underlying ontologies, but also by rule-based reasoning.

3 From a non-intrusive monitoring to service delivery

3.1 User location

Location is a context information from which user activities can be inferred. In this work, we propose to infer such information, usually deduced from sensors data (e.g. motion detectors, switches contacts), from services request. In addition to ontology concepts described above, we add semantic relationships to link them. Figure 3 (left part) shows an excerpt of the ontology taxonomy (*TBox* with concepts and object properties) for user location. Local activation devices `Local Device` and remote activation devices `Remote Device` (concepts mutually exclusives) are defined to infer the property `isLocatedIn` related to the user. Rule 1 from table 1 allows an inference engine to deduce that a user *?u* who is using a local activation device *?d* which stands in room *?r*, is himself in this room. `Door` concept is defined with exactly two links `isBetween` towards two concepts `Room`. Given this logic representation, rules are expressed to infer properties such as `isEntering` and `isLeaving` from a door activation (see rules 2 and 3 from table 1). A user *?u* located in room *?r1* who is asking for the door *?d* (between *?r1* and *?r2*) opening, is actually entering *?r2* and leaving *?r1*.

3.2 User intention

When the user asks for a service, he knows that this activation will have an effect on his environment. In other words, this latter asks for a home automation service for its effect. In this work, we seek to infer the desired effect from a

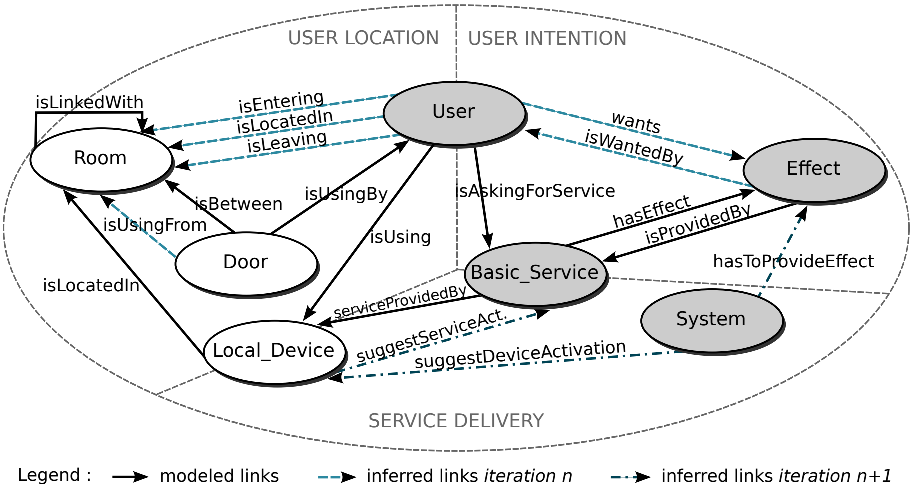

Fig. 3. Excerpt of the ontology for user location, user intention and service delivery

N°	Property	SWRL expression *(Semantic Web Rule Language)*
1	isLocatedIn	LocalDevice(?d), Room(?r), User(?u), isLocatedIn(?d, ?r), isUsing(?u, ?d) → isLocatedIn(?u, ?r)
2	isLeaving	Door(?d), Room(?r), User(?u), isUsing(?u, ?d), isUsingFrom(?d, ?r) → isLeaving(?u, ?r)
3	isEntering	Door(?d), Room(?r1), Room(?r2), User(?u), isBetween(?d, ?r1), isBetween(?d, ?r2), isLeaving(?u, ?r1), isLinkedWith(?r1, ?r2) → isEntering(?u, ?r2)
4	wants	BasicService(?b), Effect(?e), User(?u), hasEffect(?b, ?e), isAskingForService(?u, ?b) → wants(?u, ?e)
5	suggestDevice-Activation	BasicService(?b), Device(?d), Effect(?e), Room(?r), System(?s), hasToProvide(?s, ?e), isLocatedIn(?d, ?r), isProvidedBy(?e, ?b), isWantedIn(?e, ?r), serviceProvidedBy(?b, ?d) → suggestDeviceActivation(?s, ?d)
6	suggestService-Activation	BasicService(?b), Device(?d), Effect(?e), Room(?r), System(?s), hasToProvide(?s, ?e), isLocatedIn(?d, ?r), isProvidedBy(?e, ?b), isWantedIn(?e, ?r), serviceProvidedBy(?b, ?d) → suggestServiceActivation(?d, ?b)

Table 1. Excerpt of rules for location/intention (1,2,3/4) and service delivery (5,6)

semantic analysis of control logs. Figure 3 (right part) shows an excerpt of the ontology taxonomy for user intention, based on the `Effect` concept. `Luminosity`, `Darkness`, `Heat`, `Quietness` are sub-classes of effect which can be linked with `Basic Service` through the `hasEffect` property. The user intention can thus be deduced by an inference engine applying rule 4 from table 1: a user *?u* who is asking for a basic service *?b*, wants its associated effect *?e*.

3.3 Service delivery

From context information on user location and intention previously inferred, the inference engine can infer services proposition. Figure 3 (bottom part) shows an excerpt of the ontology taxonomy for service delivery. The `System` concept, which acts as an interface for user-system interactions, is added to concepts and properties described above. Service delivery is based on `suggestDeviceActivation` and `SuggestServiceActivation` properties, applied respectively on `System` and `Device` (see rules 5 and 6 from table 1). For example, a user who wants luminosity (e.g. `wants(User1, Luminosity)`), and who enters a room (e.g. `isEntering(User1, Bedroom)`) will be offered services allowing to increase luminosity in the incoming room.

The design and deployment of such an ontology has to be flexible to adapt to user needs and evolving environments. On the prospect of a large scale service deployment, a model-driven process developed in previous works [10] is applied for populating the ontology and a software architecture is defined for runtime.

4 Implementation: from design to deployment

4.1 A model-driven architecture for ontology design

In the context of ambient assisted living, we strongly believe that, in addition to the home automation expert, a non-expert - but someone who is able to properly consider the context of living with a disability (e.g. an occupational therapist, a family member) - must be included in the design process. We previously developed in [10] a model-driven flow for the design of assistive home automation systems. Platform independent modeling (*PIM*) are separated from platform specific modeling (*PSM*)[2], so that non-expert designers can focus on specification rather than on implementation. Using a domain specific language, they can model both *environment* and *interactions* to define respectively the context of use and the user needs, before automatically generating the control code[3].

In this paper, we propose to automatically generate ontology individuals with the same domain specific language. Figure 4 shows the context-awareness part which consists of several steps integrated in the existing design flow. In step 1', relevant model concepts defined by this latter at *PIM* level are transformed into ontological concepts (*PSM* level). They have some basic information such as the layout of housing, devices parameters, but also services defined to consider user needs. Step 2' is about generating individuals conformed to our ontology defined with *OWL* (*Web Ontology Language*), the most commonly used ontology language. The joint use of a model-driven architecture and ontology[4] makes

[2] In accordance to the *Model-Driven Architecture* (MDA) specification [7].

[3] Experiments have been conducted with students and occupational therapists with developed tools (see demo here: `http://www.youtube.com/watch?v=B4FJcdDBa6Q`).

[4] The *Ontology Definition Metamodel* [8] is a specification to unify MDA and Ontology, making OWL conformed with the three modeling levels of the MDA specification.

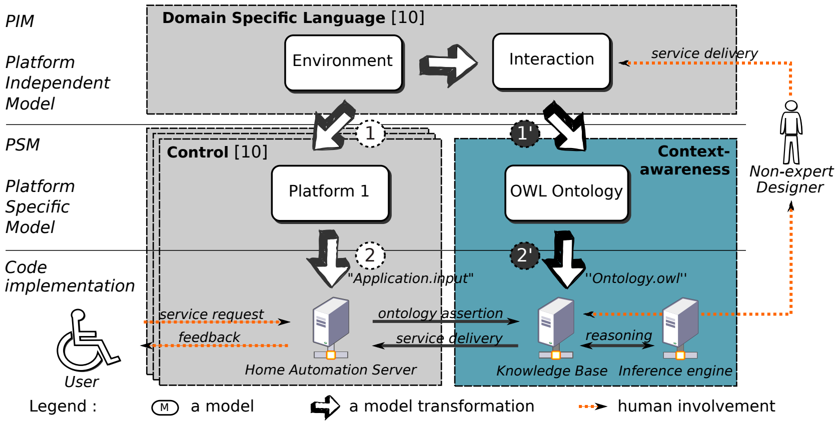

Fig. 4. Model-driven flow for control and context-awareness parts

sense to benefit from their respective advantages, thus leading to the definition of a complete control and context-awareness system for smart home : MDA and languages derived from this paradigm provide tools, concepts, and languages to create and transform models, but offer few means to formally interpret the semantics of models at runtime. On the contrary, ontologies make knowledge machine-understandable, by formalizing and by making explicit the semantic relations between concepts.

4.2 A software architecture for ontology deployment

The ontology deployment has to deal with scalability at runtime. From models defined with the domain specific language developed in previous works (see previous section), individuals are generated to populate the ontology. Associated with *SWRL* rules, it composes the knowledge base (*T-Box*, *A-Box*, and rules) representing user-system interaction.

Each time the user asks for a service, the application have to process the control log, to update the ontology, to write logs in files and to send back services proposition on user interface. Figure 5 shows the proposed software architecture for the context-aware application. Written in Java using the OWL API [11] to interact with the knowledge base and the *HermiT* [12] inference engine, this application is based on six units:

1. `OWL Parser` takes as input control logs coming from the home automation server and translates it in *OWL* format
2. `Supervision Unit` loads the ontology and adds assertions related to logs (e.g. the log "*User1 Switch_ON.TV1*" is added as assertions "*User1 isAskingForService Switch_ON*" and "*User1 isUsing TV1*")
3. `Inference Unit` calls the inference engine for applying SWRL rules on ontology with newly added assertions and sends inferences to the filtering unit

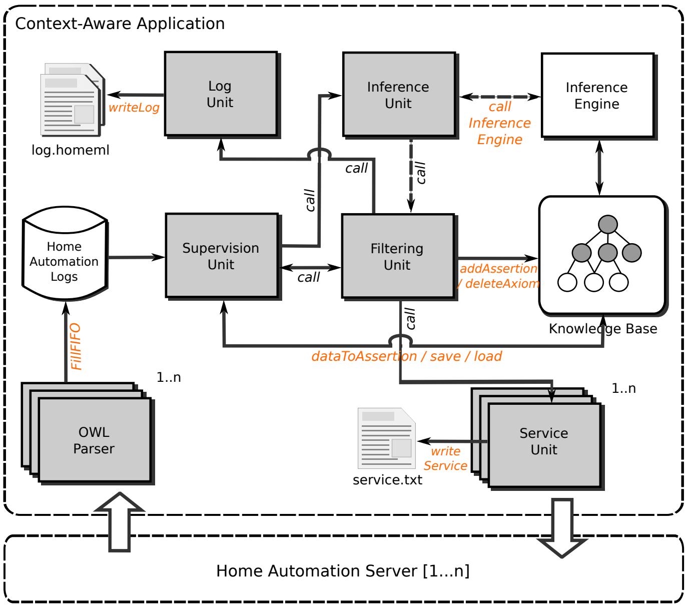

Fig. 5. Software architecture with application and modeling part

4. `Filtering Unit` updates the ontology (e.g. update location consists of removing old assertions and filtering inferences to add the more accurate location)
5. `Log Unit` saves the inferred information on user location and intention in a text file and saves input logs conforming to the *HomeML*[5] syntax [13]
6. `Service Unit` finally saves services proposition in the same manner, before sending back the list of services to the home automation server

5 Case study : the Domus Smart Home

5.1 A person tracking application

To validate our approach, a person tracking application is first presented. This case study is based on a real monitoring dataset recorded in the Domus smart home [15, 16] from the University of Sherbrooke (see figure 6). This dataset is the result of two series of experiments with six adults to evaluate the early morning routines (wake up, toileting, preparing and having breakfast) during 10 days for series 1 and 5 days for series 2. Each experiment lasted about 45 minutes. Table 2 shows the excerpt of activities performed in day 2 by user 1 [17] chosen for this person tracking application. Since our aim is to analyse control logs and

[5] *HomeML* is a format to store dataset openly accessible from a web repository [14].

the dataset is based on sensors, we have converted some of these sensors logs by user requests on devices when such devices can be controlled. For example, light sensor logs from *Light5101*, *Light5105* and *Light5106* can be replaced by control logs considering a light as a controllable device (represented by * symbol in table 2). Otherwise, the flow meter *Fl01* is mounted on the cold water tap of the kitchen sink and sends an event when the tap is opened. As we consider this latter cannot be remotely controlled, we keep logs 8 and 9 as sensors logs (represented by ° symbol in table 2).

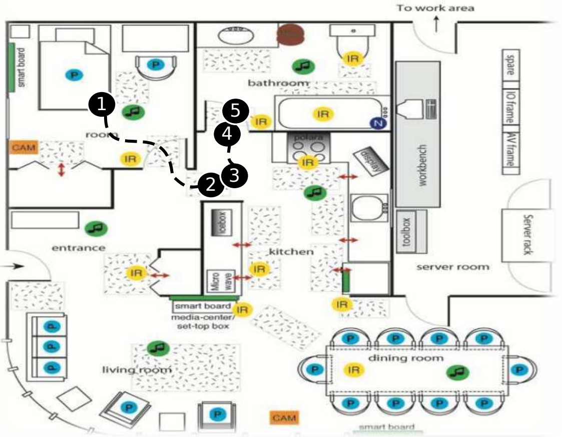

Fig. 6. A person tracking scenario in the Domus smart home, from [17].

Step 1 `The user` (*user1*) `switches on the bedroom light`: "*[Log 1]: Light5105.SwitchON*" (log 1 from table 2 and step 1 from figure 6). This log is introduced in the ontology through the following assertions[6]: "*[Assertion 1]: User1 isAskingForService SwitchON*"and "*[Assertion 2]: User1 isUsing Light5105*". Knowing that the ontology defines *Light5105* as a local activation device located in *Bedroom*, and that this latter has a *SwitchON* basic service with *Luminosity* effect, rules 1 and 4 from table 1 can be applied. The inference engine execution allows then to lead to the following inferences: *[Inference 1]: User1 isLocatedIn Bedroom* and *[Inference 2]: User1 wants Luminosity.*

Step 2 The user now wants to go to the bathroom: after opening the bedroom door, `he switches on the kitchen light`: "*[Log 3]: Light5101.SwitchON*". Since this light is considered as a local activation device, the following inferences are realized: *[Inference 3]: User1 isLocatedIn Kitchen* and *[Inference 4]: User1 wants Luminosity.*

[6] Each home automation log is divided into *isAskingForService* and *isUsing* properties.

N°		**Time**	**ID**	**Name**	**Location**	**Service/State**
1	*	8:56:17	5105	Light	Bedroom	SwitchON
2	*	8:56:26	0215	Door	Bedroom	Open
3	*	8:56:31	5101	Light	Kitchen	SwitchON
4	*	*8:56:35*	*5105*	*Light*	*Bedroom*	*SwitchOFF*
5	*	8:56:39	0215	Door	Bedroom	Close
6	*	8:56:44	0216	Door	Bathroom	Open
7	*	*8:56:46*	*5106*	*Light*	*Bathroom*	*SwitchON*
8	°	8:56:51	Fl01	T.ColdWater	Bathroom	Open state
9	°	8:56:54	Fl01	T.ColdWater	Bathroom	Close state
10	°	8:57:09	IR01	Infrared	KitchenSink	Close state
11	°	8:57:12	IR02	Infrared	KitchenOven	Close state
12	*	*8:57:12*	*5106*	*Light*	*Bathroom*	*SwitchOFF*

*: logs coming from user requests / °: logs coming from sensors

Table 2. Dataset (excerpt) of series 1 / user 1 / day 2 considering control logs

Step 3 `The system proposes him to switch off the bedroom light.` This proposition is based on the inverse effect property (e.g. *Luminosity* isInverseOf *Darkness*). The user being located in the kitchen, the system proposes to switch off lights in other rooms. In this scenario, this service is accepted by the user: "*[Log 4]: Light5105.SwitchOFF*".

Step 4 To go into the bathroom, `he asks for the door opening`: "*[Log 6]: Door0216.Open*". Considering this door linked with exactly two rooms with the *isBetween* property, rules 2 and 3 of table 1 can be applied. The user-door interaction allows the inference engine to extract the following information: *[Inference 5]: User1 isEntering Bathroom* and *[Inference 6]: User1 isLeaving Kitchen.*

Step 5 `The system proposes him to switch on the bathroom light` (log 7), from user intention and location previously inferred. These inferences are made possible trough rules 5 and 6 from table 1. Later, with the location information from sensor IR01, the system proposes him to switch off the bathroom light (as in step 3). This proposition is accepted by the user: "*[Log 12]: Light5106.SwitchOFF*".

5.2 Results on the dataset

Results on the whole dataset, presented in table 3, can be interpreted in different ways according to the users needs. If the goal is to reduce the cost related to sensors installation and maintenance, it is possible to limit their use by replacing sensors data by a semantic analysis of control logs. If the goal is to increase dependability, it is possible to use our context-aware approach keeping sensors already installed to have information redundancy (i.e. a couple {*sensor*, *actuator*} when the user interacts with his environment).

	Series 1		**Series 2**	
	Limitation	Redundancy	Limitation	Redundancy
User 1	79%	13%	82%	6%
User 2	68%	10%	79%	9%
User 3	75%	6%	90%	2%
User 4	70%	9%	79%	6%
User 5	59%	48%	81%	27%
User 6	73%	12%	76%	6%
Average	71%	16%	81%	9%

Limitation : sensors use / Redundancy : information (sensor ↔ actuator)

Table 3. Results of the context-aware system on the entire Domus dataset

Limitation of sensors use The Domus Smart Home is equipped with infrared motion detectors, pressure sensors, switches contacts, flow meters, etc. As an illustration, the person tracking application based on the dataset excerpt presented in table 2 shows that our approach can limit the use of sensors to 37.5%. This means that only 3 sensors over 8 can potentially be used to ensure this application (see logs from table 2) : the flow meter *Fl01* and the two infrared motion detectors *IR01* and *IR02*) which cannot directly be associated with an actuator[7]. More generally, considering the whole dataset, actuators can replace electromagnetic door sensors and light switches. Thus, it should be possible to reduce the use of sensors in the Domus Smart Home to 71% for series 1 and 81% for series 2 (see table 3). The results vary from one user to another according to his/her mode of interaction with home automation devices. For example, the use of sensors for user 5 (in series 1) can be limited to 59%: this means that only 10 sensors over 17 can potentially be used (the 7 others can be replaced by actuators which logs can replace sensor logs). Sensors which cannot have associated actuators, like infrared motion detectors or pressure sensors for user location, can still be replaced by logic inference with a lower quality of services (*QoS*)[8].

Redundancy information In a context of severe disabilities without economic constraint, quality of services may be preferred. If we decide to keep all sensors in the Domus Smart Home, our approach can improve the dependability by information redundancy. In this work, the information "*The user interacts with his*

[7] If *IR01* and *IR02* give precise user location in one part of the kitchen, our context-aware approach can infer it at a room scale when a local activation device is used.

[8] *QoS* is related to data acquisition frequency and reliability of inferred knowledge.

home automation system" may come from two different sources of information: sensors data (i.e. the device states), but also actuator data (i.e. user request sent). As an illustration, if we consider the dataset excerpt presented in table 2, one can see that 8/12 sensors logs can be associated with control logs (see logs with *). More generally, we observe on the whole data set that 16% of sensors logs (series 1) can be associated with either one control log or the result of logic reasoning to deduce user location (see table 3) [9]. Information redundancy depends directly on user-system interactions: for example, user 5 often interacts with his home automation devices what leads to a maximum redundancy of 48% on series 1. However, when the user walks through rooms without sending controls, no logic inference is made related to his location. The context-aware approach is described as proactive: data acquisition frequency depends on the home automation system use. This proactiveness also explains differences between results from series 1 and series 2. In the second series, user was asked to perform the early morning routine of series 1, and perform in addition an activity which does not require any controls (i.e. learning a tea recipe). The use of actuators decreases from 29% to 19% between the two series, leading to decrease the information redundancy (9% on average for series 2).

6 Related Works

The use of ontologies and related tools are of great interest in the domain of context-awareness for ambient assisted living, mainly due to their interoperability and logic reasoning capabilities. In [18], an ontology is defined to model home automation environment on a large scale (*building automation system*), allowing to face to the heterogeneity of home automation platforms (e.g. *KNX*, *LonWorks*, *BACnet* technologies). Instead of using gateways between these different platforms (configurable with specific tools), the ontology is an abstract view of the heterogeneous network. DogOnt (*Domotic OSGI Gateway*) [19, 20] also support the hardware and telecommunication independent description of home automation environment. Using SWRL rules, logic reasoning is applied to automatically recognize devices classes for defining their technical parameters.

Most of the work on activities of daily living recognition based on ontologies [6, 25] relies on the use of triples {*subject, predicate, object*} to model context. *CoBra* [21] and *Gaia* [22, 23] are two of the best known frameworks, based on a multi-agents architecture in charge of collecting and processing contextual information. For example, in [22], when a powerpoint application is used and the number of people in a room is more than three, system can infer that the room activity is a presentation: *People(Room2401, >=, 3) and Application(PowerPoint, Running)* ⇒ *RoomActivity(2401, Presentation)*. In [24], the *TempCRM* ontology extends the RDF triple commonly used with time. Depending on the temporal usage profile of home automation devices, potential emergencies can be detected.

[9] Results are based on the original dataset without cleaning data from duplicates.

For example, *Device(gas_switch#1, is, on, 18:38:41)* describes the opening of the gas which have to be closed after a certain amount of time.

The deployment and implementation of context-aware applications is currently not trivial due to the potentially large size of the ontology. In *Conon* [26], the context is separated in two distinct parts: the first one is related to concepts describing the system at a high level of abstraction, whereas the second one is responsible for the description of domain specific concepts. For example, the generic concept "*IndoorSpace*" is translated in "*Building*", "*Room*", and "*Corridor*" in the home automation domain. This hierarchical decomposition is found in [27] which propose to separate micro/macro context to facilitate the implementation of such an application. The proposed approach deals with the relative complexity of context information processing in distributing reasoning process on *Devices Nodes* which have knowledge of micro context without loosing the benefits of macro information processing with *Coordinator Nodes*.

These works are based on sensors data, from which context information and knowledge can be inferred via logic reasoning. Our context-aware approach seeks to infer information usually sensored from the analysis of control logs. From this point of view, the ontology and rules presented in this paper differs noticeably from existing works.

Previous works from [28] highlight the interest of monitoring activities based on control logs to limit the use of sensors. The proposed monitoring system is based on a frequency analysis in which a service is considered as a repeating pattern over time. Based on the hypothesis of very regular activities, the goal is to automatically extract and to propose daily living scenarios to wheelchair-bound users with limited moving capabilities. However, this hypothesis is not always verified in practice [29]. For example, the user needs to switch on/off lights in rooms leading to the bathroom at night. This kind of interaction can be requested from time to time, in a non-regular manner: at any hour of the night, any day of the week, etc. In this paper, instead of a frequency analysis based on the assumption of regular activities, we proposed a context-aware system based on a semantic analysis of each control log. From the ontology and rules defined to model user-system interaction, logic reasoning provides the user adapted services without any training steps and using as few sensors as possible.

7 Conclusion

The considerable increase of dependent people brings out important needs of monitoring to ensure a safe and secure independent living and adapted services to compensate for their potential disability. In this context, this paper has presented a non-intrusive context-aware system based on ontology for ambient assisted living in smart home. The context-awareness of such systems is not based on sensors use, but on the semantic analysis of logs recorded each time the user asks for a service. An ontology with associated rules allows the inference engine to extract information usually sensored. As the design and the deployment of ontology is still not trivial, a model-driven process is applied for populating the ontology and a software architecture is defined for runtime. Finally, a case study based on the Domus Smart Home dataset validates our approach.

The person tracking application and results on the entire dataset are encouraging: depending on the context of use, it is possible to i) limit the use of sensors (low QoS) or ii) increase the information redundancy (high QoS). To do so, our approach considers control logs and logic inferences i) instead or ii) in addition to switches contacts and motion detectors logs. However, to achieve a complete monitoring system over time, this approach should consider both user feedback and sensors whose information can not be inferred from home automation logs: information coming from biomedical sensors (e.g. blood pressure, core body temperature, etc.) or accurate information meters (e.g. temperature, luminosity, etc.). These kind of information could also be translated in assertions in our context-aware system.

Service delivery is also an interesting contribution to allow users faster and easier access to services from their control interfaces. Some people with disabilities including motion impairment (e.g. tetraplegia) can only control their home automation system by means of an automatic scrolling interface. In such restrictive situations, the selection of the desired service is very time consuming. Benefiting from the model-driven process presented in this paper, a full-scale deployment is planned in Kerpape Living Lab to evaluate services acceptability and how it can assist patients in their everyday life[10]. These apartments have been built to evaluate the autonomy of patients before leaving the center. As an experimental validation is essential to evaluate service delivery, we are currently collecting home automation logs with motion-impaired users.

References

1. United Nations, Department of Economic and Social Affairs, Population Division: World Population Ageing (2009)
2. Weiser, M.: Some computer science issues in ubiquitous computing. Communications of the ACM. 36, 75-84 (1993)
3. Yu, C.R., Wu, C.L., Lu, C.H., Fu, L.C.: Human localization via multi-cameras and floor sensors in smart home. In: IEEE SMC (2006)
4. Nourizadeh, S., Deroussent, C., Song, Y.Q., and al.: Medical and home automation sensor networks for senior citizens telehomecare. In: IEEE ICC Workshops (2009)
5. Long, S., Holder, L.: Using graphs to improve activity prediction in smart environments based on motion sensor data. In: ICOST (2011)
6. Chen, L., Nugent, C.D., Cook, D.J, Zhiwen, Yu: Sensor-Based Activity Recognition. In: IEEE Transactions on Systems, Man, and Cybernetics - Part C. 42-6, 790-808 (2012)
7. Miller, J., Mukerji, J., and al.: Model Driven Architecture (MDA). Object Management Group, Draft Specification ormsc/2001-07-01 (2001)

[10] see a home automation application with an automatic scrolling and a brain-computer interface used in one apartment of Kerpape Living Lab here: `http://www.youtube.com/watch?v=0grL24U5duM`

8. Consulting, D.F. and Deere, J.: Ontology Definition Metamodel (2005)
9. Gruber, T.R., and al.: Toward principles for the design of ontologies used for knowledge sharing. International Journal of Human Computer Studies. 43, 907-928 (1995)
10. Allègre, W., Burger, T., Berruet, P.: Model-Driven Flow for Assistive Home Automation System Design. In: 18th IFAC World Congress (2011)
11. Horridge, M. and Bechhofer, S.: The owl api: A java api for owl ontologies. Semantic Web Journal. 2-1, 11-21 (2011)
12. Shearer, R. and Motik, B. and Horrocks, I.: HermiT: A highly-efficient OWL reasoner. In: OWLED (2008)
13. Nugent, C. and Finlay, D. and Davies, R. and Wang, H. and Zheng, H. and Hallberg, J. and Synnes, K. and Mulvenna, M.: homeML–an open standard for the exchange of data within smart environments. Journal of Pervasive Computing for Quality of Life Enhancement. 121–129 (2007)
14. McDonald, HA and Nugent, CD and Moore, G. and Finlay, DD. and Hallberg, J.: A web based tool for storing and visualising data generated within a smart home. In: IEEE EMBC (2011)
15. Kadouche, R., Pigot, H., Abdulrazak, B., Giroux, S.: Support Vector Machines for Inhabitant Identification in Smart Houses. Ubiquitous Intelligence and Computing. 83-95 (2010)
16. Chikhaoui, B., Wang, S., Pigot, H.: A New Algorithm Based on Sequential Pattern Mining for Person Identification in Ubiquitous Environments. In: KDD Workshop on Knowledge Discovery from Sensor Data. 19-28 (2010)
17. Domus Monitoring Dataset , http://domus.usherbrooke.ca/jeux-de-donnees/url
18. Reinisch, C. and Granzer, W. and Praus, F. and Kastner, W.: Integration of heterogeneous building automation systems using ontologies. In: IEEE IECON (2008)
19. Bonino, D. and Corno, F.: Dogont-ontology modeling for intelligent domotic environments. Semantic Web Journal. 790-803 (2008)
20. Bonino, D. and Castellina, E. and Corno, F.: Automatic domotic device interoperation. IEEE Transactions on Consumer Electronics. 55-2, 499-506 (2009)
21. Chen, H. and Finin, T. and Joshi, A.: The SOUPA ontology for pervasive computing. Ontologies for agents: Theory and experiences. 233-258 (2005)
22. Román, M. and Hess, C. and Cerqueira, R. and Ranganathan, A. and Campbell, R.H. and Nahrstedt, K.: A middleware infrastructure for active spaces. IEEE Transactions on Pervasive Computing. 1-4, 74-83 (2002)
23. Bittencourt, I.I. and Bispo, P. and Costa, E. and Pedro, J. and Véras, D. and Dermeval, D. and Pacca, H.: Modeling jade agents from gaia methodology under the perspective of semantic web. Enterprise Information Systems. 780-789 (2009)
24. Liao, H.C. and Tu, C.C.: A RDF and owl-based temporal context reasoning model for smart home. Information Technology Journal. 6-8, 1130-1138 (2007)
25. Chen, L., Hui, Wang: A Knowledge-Driven Approach to Activity Recognition in Smart Homes. IEEE Transactions on Knowledge and Data Engineering. 24-6, 961-974 (2012)
26. Wang, X.H. and Zhang, D.Q. and Gu, T. and Pung, H.K.: Ontology based context modeling and reasoning using OWL. In: IEEE Pervasive Computing and Communications Workshops. 18-22 (2004)
27. Gouin-Vallerand, C. and Roy, P. and Abdulrazak, B. and Giroux, S. and Dey, A.: A Macro and Micro Context Awareness Model for the Provision of Services in Smart Spaces. Impact Analysis of Solutions for Chronic Disease Prevention and Management. 173-181 (2012)
28. Truong, T.B.T., de Lamotte, F.F., Diguet, J.P.: Proactive remote healthcare based on multimedia and home automation services. In: IEEE CASE (2009)

29. De Lamotte, F. and Departe, JP. and Le Saout, F. and Diguet, J.P. and Philippe, JL.: Quatra: Final report (Technical report, in french). See demo here: http://www.youtube.com/watch?v=T6GCFnkLTc0 (2008)