Towards European Portuguese Conversational Assistants for Smart Homes

Maksym Ketsmur
DETI/IEETA, Universidade de Aveiro, Aveiro, Portugal
mvk@ua.pt

António Teixeira
DETI/IEETA, Universidade de Aveiro, Aveiro, Portugal
ajst@ua.pt

Nuno Almeida
DETI/IEETA, Universidade de Aveiro, Aveiro, Portugal
nunoalmeida@ua.pt

Samuel Silva
DETI/IEETA, Universidade de Aveiro, Aveiro, Portugal
sss@ua.pt

Abstract

Nowadays, smart environments, such as Smart Homes, are becoming a reality, due to the access to a wide variety of smart devices at a low cost. These devices are connected to the home network and inhabitants can interact with them using smartphones, tablets and smart assistants, a feature with rising popularity. The diversity of devices, the user's expectations regarding Smart Homes, and assistants' requirements pose several challenges. In this context, a Smart Home Assistant capable of conversation and device integration can be a valuable help to the inhabitants, not only for smart device control, but also to obtain valuable information and have a broader picture of how the house and its devices behave. This paper presents the current stage of development of one such assistant, targeting European Portuguese, not only supporting the control of home devices, but also providing a potentially more natural way to access a variety of information regarding the home and its devices. The development has been made in the scope of Smart Green Homes (SGH) project.

2012 ACM Subject Classification Human-centered computing → Natural language interfaces; Human-centered computing → Interactive systems and tools; Human-centered computing → Sound-based input / output; Human-centered computing → Interaction techniques

Keywords and phrases Smart Homes, Conversational Assistants, Ontology

Digital Object Identifier 10.4230/OASIcs.SLATE.2019.5

Funding Research partially funded by project Smart Green Homes (POCI-01-0247-FEDER-007678), a co-promotion between Bosch Termotecnologia S.A. and the University of Aveiro, and IEETA Research Unit funding (UID/CEC/00127/2013.).
Samuel Silva: funded by Portugal 2020 under the Competitiveness and Internationalization Operational Program, and by the European Regional Development Fund through project SOCA-Smart Open Campus (CENTRO-01-0145-FEDER-000010).

1 Introduction

Home Assistants, such as Amazon Alexa [14], have gained popularity and there is also a consistent upward trend of integration of smart devices into our homes, from small sensors used to monitor home environment to large smart appliances.

Smart Home assistants aim to provide a more natural way to interact with our home and the smart devices that are becoming an intrinsic part of its environment. The most popular Smart Home assistants are Google Assistant, Amazon Alexa and a recent intelligent assistant

8th Symposium on Languages, Applications and Technologies (SLATE 2019).
Editors: Ricardo Rodrigues, Jan Janoušek, Luís Ferreira, Luísa Coheur, Fernando Batista, and Hugo Gonçalo Oliveira; Article No. 5; pp. 5:1–5:14
OpenAccess Series in Informatics
OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

contributed by Yandex, known under the name "Alice" [13]. These assistants are voice based and, in most cases, are integrated with small devices such as Google Home (Google Assistant) and Echo Dot (Amazon Alexa), but are mostly limited to supporting simple features, such as controlling the lights or checking the weather. Also, they do not support the European Portuguese language.

Despite all recent advancements, and the popularity of home voice assistants, the control and access to the information regarding Smart Homes needs improvement. Considering the diversity of smart devices (e.g., smart lights, smart plugs, smart appliances) and sensors (e.g., air quality, temperature, occupancy) available today, it is possible to obtain a wide range of potentially valuable information pertaining our home and its devices, which can be harnessed to provide the occupants with a broader picture of how home and its devices behave. This can, for instance, allow controlling appliances remotely, schedule their activities, and obtain detailed information regarding their consumption, also bringing forward a more effective use of device smartness towards economy and comfort.

These new capabilities need to be aligned with the users' needs. A recent inquiry, by the authors [10] considering 20 participants aged between 10 and 63 years old with different scientific backgrounds revealed a list of capabilities deemed important by users, such as: remote control, home state report, voice communication, appliances and light state querying and control, appliances activities scheduling, temperature control of appliances (e.g., water heater) and house divisions; and being informed about resource consumption.

Considering the limitations of already existing Smart Home assistants, the need for a consistent integration of new devices into our homes, and users expectations, the development of adequate Smart Home assistants poses several challenges to: (1) store all the relevant information (consumptions, interactions, inhabitant behavior) in organized way, creating a knowledge base; (2) provide the means for the assistant to use this information; (3) support users native language, in this case European Portuguese.

The remainder of this article is organized as follows: next section presents related work regarding Conversational Assistants and ontologies for smart environments. Afterwards, Section 3 describes the scenario selected as context to support the development and Section 4 presents the conversational assistant architecture and the knowledge base to support it. Then, Section 5 presents the main capabilities of the developed Assistant and the results obtained during its first evaluation. Finally, conclusions and future work complete the paper, in Section 6.

2 Background and Related work

This section presents background information in Assistants and ontologies, as well as some relevant related work, considered important to contextualize the presented work.

2.1 Conversational Assistants

Conversational assistants perform similar interactions to chatbots and allow speech as input and output [12].

Google Assistant, Apple's Siri, and Microsoft Cortana are examples of popular conversational interfaces. They are, in general, single-turn assistants only suitable for information seeking and simple execution of control commands. They do not adequately support multi-turn voice interactions (dialogue) and use of context.

To enable the interaction between the human and the machine, in a dialog format, a typical conversation system, as illutrated in Fig. 1, must integrate at least five modules [2, 22]:

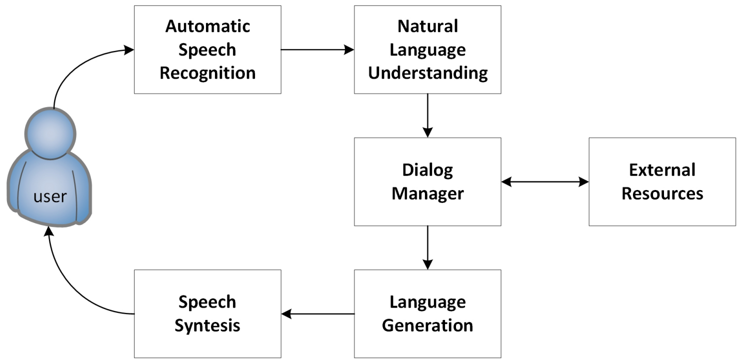

Figure 1 Typical system structure of a conversational assistant.

(1) Automatic Speech Recognition (ASR) converts recorder audio signal (speech) into text; (2) Natural Language Understanding (NLU) tries to understand the input sequences of words to identify important information such as intentions and entities; (3) Dialog Manager (DM) manages dialog and context information, considering intention, entities and previous conversations; (4) Natural Language Generation (NLG) generates sentences; (5) Speech Synthesis, uses Text-to-Speech (TTS) to produce synthetic speech.

2.2 Tools for Assistant development and NLU processing

Even though there is no option of a complete (and configurable) assistant to use as basis for the development, there are some tools that can be used to ensure the features for the different modules identified, above. Particularly relevant for the development of such assistant is the NLU module. Considering the specificity of the domain and the language requirements, there are some tools and resources available to support the developing of conversational assistants [11], as summarized in Table 1.

Table 1 Tools and APIs useful for Chatbots and Conversational Assistants development.

Name	Main capabilities	Access method	Portuguese
DialogFlow	NLU + DM	HTTP	Brazilian
Microsoft LUIS	NLU	HTTP/SDK	Brazilian
IBM Watson Assistant	NLU + DM	HTTP/SDK	Brazilian
Amazon LEX	NLU + DM	HTTP/SDK	Yes
WIT.AI	NLU	HTTP	Yes
Microsoft Speech	ASR + TTS	SDK	Yes

2.3 Ontologies for the Smart Home

Ontologies are commonly used to address data, knowledge, and applications heterogeneity, enabling the support to a service-oriented framework in Smart environments [19, 20, 5], such as Smart Homes that, unlike the traditional home, can be autonomous, consider the different habits of the occupants and adjust the setting accordingly in order to facilitate the household daily life. The use of an ontology for the Smart Home scenario allows to model and describe the different aspects of the smart "things" and residents by defining their semantic properties, the information they can supply or, even, the actions or controls they can perform [17, 4].

DomoML [18] was one of the first approaches that provided a full, modular ontology of household environments, divided into three main ontologies: (1) DomoML-env, defining all fixed elements inside the house; (2) DomoML-fun, describing the functionalities of each house device, in a technology independent manner; and (3) DomoML-core, providing support for the correlation of elements of DomoML-fun and DomoML-env, including the definition of physical quantities.

DogOnt [3] is another approach supporting device/network independent description of homes, classified as “controllable” and “uncontrollable” things, providing one of the few domain models to fit real world domotic systems capabilities and support interoperation between current and future solutions. Also, it uses inheritance mechanisms to automatically associate states and functionalities to the modeled elements.

Human activities, needs, and preferences must also be covered by the ontology and were addressed by Ye et al. [21], defining a generic approach to derive knowledge about context predicates and activities in a structured manner.

Beyond the home environment, and occupant activities and preferences, ontologies related to smart appliances and sensors [17], home energy management [16], and devices and appliances resource consumption [4] have been proposed, in the literature, which cover operational data of appliances and smart devices, the classification of home electrical appliances provided by various vendors and manufacturers, and handling of their consumption data.

3 Application scenario

To support the implementation of our ideas and to have a concrete scenario eliciting requirements, we designed a virtual Smart Home that contains virtual partitions mapped to rooms, in our lab, and many devices, in part real, in part virtual. The real part of the virtual home has the infrastructure to communicate and interact with devices; the virtual devices (appliances) are simulated as computer programs, which generate data and allow the interaction between the assistant and the virtual device.

Our virtual Smart Home has more than 10 virtual appliances generating, using probabilistic algorithms, resource consumption data that posteriorly is stored in the Smart Home semantic knowledge base. Also, new virtual appliances can be easily attached to the system using a simple XML file containing the appliance’s definition and characteristics.

4 The Conversational Assistant

This section presents our conversational assistant architecture and its modules, along with a brief description of the knowledge base and ontology supporting the whole system.

4.1 Architecture

Figure 2 presents the architecture adopted for the assistant, composed by three core modules (NLU, DM and NLG) and two attachable modules (ASR and TTS).

The three main modules can support interaction based in written text, making possible a chat version of the assistant. By coupling to the pipeline two additional modules (ASR and TTS), we get a full conversational assistant capable of accepting speech as input and producing speech as its output.

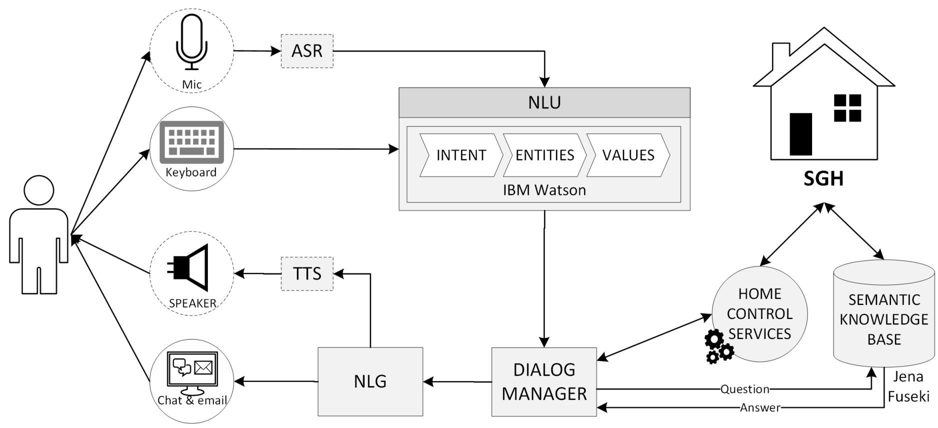

Figure 2 Overall architecture for the proposed Assistant.

4.2 Natural Language Understanding (NLU)

The natural understanding module aims to process the input information in text format and identify its intention and entities.

The NLU processing, in our system, is carried out by an external NLU service, IBM Watson Assistant, that identifies intents, entities and their values, essential to the correct understanding of the user's input, dialog management and output. IBM Watson was selected mainly due to simplicity of use and support for NLU and dialog management capabilities. Initial experiments proved that being available only for the Brazilian variant of Portuguese was not a major limitation.

Watson intents represent the purposes or goals expressed in an input text such as appliances control or consumption querying. While intents represent the purpose or goal, entities represent the context for that purpose. They are important nouns and named entities in the input text. For example, if the user's intention is control then the device and location entities are required.

In Watson, entities can be of two types: dictionary-based entities and contextual entities. The dictionary-based entities are those for which specific terms, synonyms, or patterns can be defined. At run time, the service finds entity mentions only when a term in the user input exactly matches (or closely matches if fuzzy matching is enabled) the value or one of its synonyms. Context-based entities are those for which occurrences of the entity, in sample sentences, are annotated to teach the service about the context in which the entity is typically used.

Each of the entities has a set of values that represent it, for example, the home entity can have room, garage, garden values. Each of the values may have synonyms or patterns that help to better identify them in the input text (e.g., the room value may have synonyms, such as kitchen, living room, and bathroom).

Taking into account the domain of the Assistant, intents and entities were defined to enable the identification of the user's purpose, what actions to take, on which appliances, and in which home partition. The main intents and entities added to Watson are presented in Table 2.

Adding these intents and entities to Watson is done through a training process, consisting essentially in providing sentence examples. Table 3 presents some of the sentences (examples) used to train each of the intents and the related entities. The first rows presents some of the examples used to train the identification of the intent `request` where some sentences are about

Table 2 Watson intents, entities and values.

Name	Type	Description
greet	intent	Identifies when user starts the conversation with the assistant.
command		Identifies when user tries to control any of the appliances or lights.
request		Identifies when user asks the assistant about appliances state or consumption.
action	entity	Represents the action to take, for example consumption querying, appliances state control or error identification.
building_thing		Identifies the appliances or lights that must be controlled in case of command intent or queried about its state or consumption in case of request intent.
error		Identifies when users tries to know about appliance issues, if it has any errors or not.
building_environment		Identifies the room (e.g. kitchen or living room) or home part (e.g. garden or garage) if present in the input sentence.
resource		Identifies resource (water, gas or energy) present in the input sentence.
sys_date		Identifies dates in the input sentences (this month, this week, today)

consumption and others about device state querying. Also, in general, all of the examples for the `request` intent contain already defined entities, such as: `action`, `resource`, `sys_date`, `building_environment`, and `building_thing`, which helps to get a better confidence level in intent recognition.

Table 3 Illustrative examples for training Watson intents.

Intent	Examples	English translation
request	Qual foi o [consumo]?	What was the [consumption]?
	Quanto [consumi]?	How much I've [consumed]
	Qual foi o [consumo] de [energia] da [TV]?	What was the [energy] [consumption] of [TV]?)
	Qual foi o [consumo] de [água] [esta semana]?	What was the [water] [consumption] [this week]?)
	Quanto [consumi] [esta semana] na [cozinha]?	How much I've [consumed] [this week] in the [kitchen]?)
	Como [estão] as [luzes] na [cozinha]?	How [are] the [lights] in the [kitchen]?)
	A [TV] na [sala] [está] [ligado]?	[TV] in the [living room] [is] [on]?)
command	Liga a televisão!	Turn [on] the [TV]!
	Desliga o GEOS !	Turn [off] the [GEOS]!
	Ligue as luzes na cozinha!	Turn the [lights] [on] in the [kitchen]!

To register a new appliance in the system, the information about its synonyms, location and capabilities, made available in a XML file, is used to produce examples and train Watson. As an example, to register a new TV and querying about its state, the implemented training process will automatically generate example sentences, like the following: What is the TV state? The state of television is on? What is the state of TV in the living room?

As an example of what Watson delivers, after being trained, Figure 3 represents Watson's response, in JSON format, for the input "What was the water consumption this week in the kitchen?". The JSON output contains valuable information for the identified intents and entities, location and the corresponding confidence. The location specifies the identified term position, in the input text, and may be helpful in the construction of complex responses. The confidence information is important to make dialogue decisions considering only intents or entities with confidence above a certain threshold (we use 80%).

4.3 Dialog Manager

The Dialog Manager module processes all the information from the NLU module in order to obtain the requested information or execute desired commands. This module consists in a set of rules that detect information changes and conduct the dialog accordingly.

```
{
"intents": [ { "intent": "request", "confidence": 0.9816005229949951 } ],
"entities":
[
    {   "entity": "action", "location": [ 11, 18 ],
        "value": "consumption_query", "confidence": 1 },
    {   "entity": "resource", "location": [ 22, 26 ],
        "value": "water", "confidence": 1 },
    {   "entity": "sys-date", "location": [ 27, 38 ],
        "value": "2019-03-03", "confidence": 1 },
    {   "entity": "sys-date", "location": [ 27, 38 ],
        "value": "2019-03-09", "confidence": 1 },
    {   "entity": "building_environment", "location": [ 42, 49 ],
        "value": "kitchen", "confidence": 1 }
],
"input": {
"text": "What was the water consumption this week in the kitchen?" }
}
```


Figure 3 Example in JSON of the structure and contents of the Watson response.

For each intent, the Dialog Manager has a predefined frame (set of slots) that need to be filled. For instance, when the intent is a `command` the values for an `action, device and home division` need to be provided. Figure 4 shows how the system processes the Watson output and identifies missing information in each step. If the verification of the information fails, a feedback message (question) is generated to the user. The user is queried in order to obtain the missing information and complete the request. The information regarding intent, entities and values are kept until user change the intent.

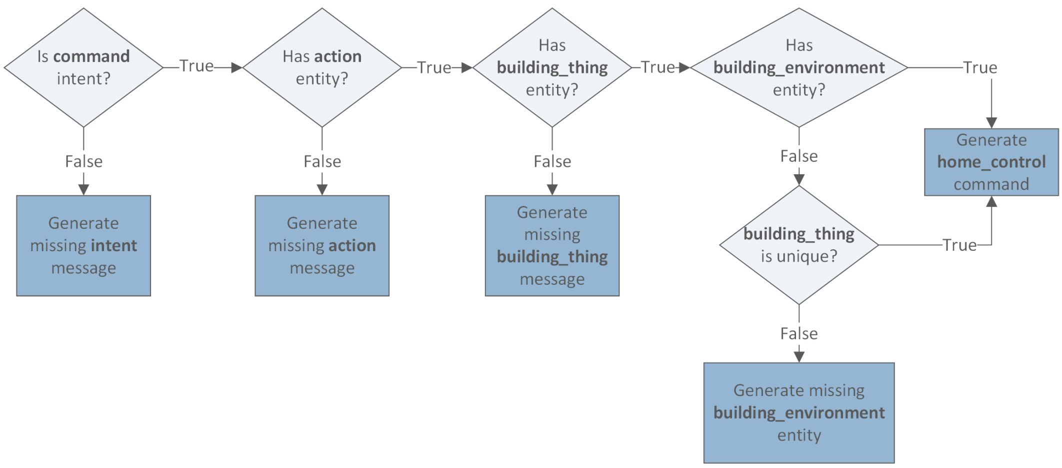

Figure 4 Command intent decision flow.

If the user does not state the `building_environment`, the Dialog Manager tries to infer that information by querying the knowledge base. If inference results in a single value the request is complete. Otherwise, the system generates a message, notifying the user that more than one device of that type is available and requests the specification of its location to the user.

In the case of `command` intents, using previously identified information, the Dialogue Manager builds the command to send to the Home Control Service, which tries to execute it, the result of command execution is sent to the DM, as reply, and feedback is provided to the user.

Besides the `command` intent, users can request information about the house. In this case, a query to the knowledge base is built using SPARQL [15]. For instance, the SPARQL query for the user request "What was the water consumption of GEOS in the kitchen, this month?", to obtain consumption information, is presented in Figure 5. The system builds this query in three steps: (1) select the `building_thing` by name and by `building_environment`; (2) select the consumption events generated by the `building_thing` that have water as consumed resource; (3) filter the information by date interval (this month).

```
SELECT (SUM (?amount) as ?totalAmount)
WHERE {
  ?building_thing :hasName "geos"^^xsd:string ;
                  :isLocatedIn :kitchen .
  ?event :hasType :ConsumptionEvent ;
         :isGeneratedBy ?building_thing ;
         :hasResource :water ;
         :hasDate ?date ;
         :hasAmount ?amount .
FILTER(?date >= "2019-03-01T00:00:00"^^xsd:dateTime &&
       ?date <= "2019-03-31T23:59:59"^^xsd:dateTime)
```

Figure 5 Example of a consumption request SPARQL query.

4.4 Natural Language Generation (NLG)

The NLG module provides mechanisms to create a response to the user. It uses predefined templates (Figure 6), which are completed with information obtained from the knowledge base. A template is chosen according to intent and entities. For example, if the user asks about water consumption for the water heater located in the kitchen, the system: (1) selects a synonym of the involved entities; (2) identifies the measurement unit (cubic meter) of the requested resource (water); (3) compiles the final response.

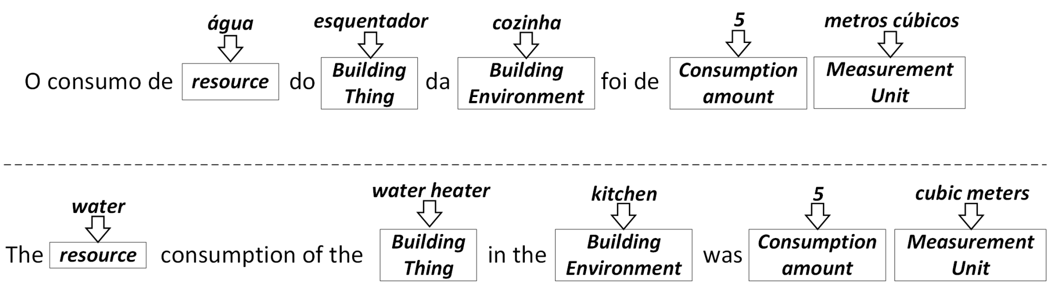

Figure 6 Example of the response template. English translation of actual Portuguese sentence is presented in the bottom of the figure.

4.5 Knowledge base

All the relevant information of the Smart Home and Users-Home interactions are stored using a semantic knowledge base, implemented using the Apache Jena Fuseki server. Information is structured according to a domain ontology developed by the authors integrating and extending existing ontologies.

The knowledge base ontology (Fig. 7) is based on 6 already defined ontologies (dogOnt [3], [1], FOAF [6], TIME [7], DUL [9] and OEMA [8]), and some classes specific to our scenario. The ontology covers all the existing Assistant functions as well as provision for future features.

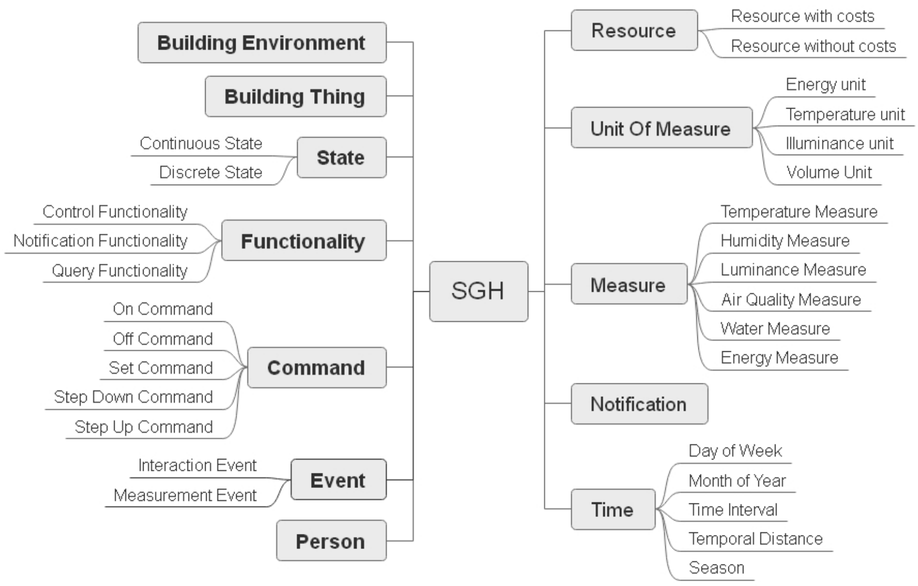

Figure 7 Ontology high level classes overview.

The `Building Environment` and `Building Thing` are the classes that define the Smart Home. The first describes the physical environment in which people live (e.g. living room, bathroom). The second class defines: (1) the equipment that is capable to produce, consume, store and measure data (`Energy Equipment`); (2) systems for detecting abnormal situations (`security system resource`).

The `Person` class represents the home occupants who have some common characteristics (name, surname, nickname, family member and age) and can interact with the home in two ways, physically or using the assistant. In the case of using the assistant, the ontology provides `Command` and `Functionality` classes. They are used by the assistant to identify if the target appliance supports the desired action.

The `Functionality` class is used to specify the appliances' capability. Functionalities are divided into three subclasses, `Control Functionality` describes the control capability of the `Building Things`; `Notification Functionality` specify which notification the `Building Things` can reply (e.g., door sensor notifies "open" or "close"); `Query Functionality` defines the capability to reply to an interrogation about its state (e.g., lights state, oven temperature).

The `Event` class defines the measurements and interaction events occurred in the home. The `Measurement Event` correspond to an appliance's consumption data or sensors data and the `Interaction Events` are physical interactions between an inhabitant and the home (e.g., turning on the oven, open the door, enter the living room).

The `State` class describes all the possible states of `Building Things`. State has two subclasses, a `Continuous State` defining an analog quantity (e.g., temperature, pressure, distance) and `Discrete State` defining a digital quantity from a set of possible values (e.g., on, off, closed, open).

The `Resource`, `Unit of Measure`, `Measure` , `Notification` and `Time` are the auxiliary classes to support store and query of consumption data and interaction notifications.

4.6 Dynamic handling of devices and appliances

One of the main characteristics of smart environments, including Smart Homes, is the dynamic nature of the devices and appliances they integrate. Therefore, a method to expand the capabilities of the assistant to integrate new devices is needed, which can recognize and handle them. Also, the process must be simple and, as much as possible, based in information that can be provided by device manufacturers.

To facilitate production of the needed information regarding the devices while making it readable by humans and machines, the XML format was adopted. Providing an XML file containing the device information (metadata) allows an easy and generic way to register the device. The file describes the available functionalities and how the assistant can interact with the new device. As an example, figure 8 illustrates the XML file for the registration of a new household equipment (water heater). The XML contains the definition of several elements to enable the recognition of devices and future interactions, such as: (1) `type`, specifying the new equipment's semantic type; (2) `room`, defining the new equipment's location in the home; (3) `synonyms`, enabling the assistant to recognize different names for the device, also helping the assistant to respond more naturally. For example, the user asks "Qual foi o consumo de água do esquentador esta semana?"(*What is the water consumption of water heater, this week?*) and the assistant responds "O consumo do esquentador esta semana, foi ..." (*The consumption of the water heater, this week, was ...*); (4) `functionalities`, enumerating the new device's capabilities. Considering the water heater, for instance, it is capable of on/off control, measure water and gas consumption and query state and errors.

The XML file structure contain an `errors` element that is optional, since this feature depends on each device and its capability to generate errors. When devices generate errors, it is necessary to provide a list of possible errors, their `identification`, `description`, possible `solution` and `severity`, usually described in user manuals. This kind of information is important since the user can ask the assistant about the error, what it means, receive suggestions of how to solve it and understand how critical it is, without consulting user manual.

```xml
<device>
  <name>geos</name>
  <type>waterHeater</type>
  <room>Bathroom</room>
  <synonyms>
    <synonym>geos</synonym>
    <synonym>esquentador</synonym>
  </synonyms>
  <errors>
    <error>
      <id>A0</id>
      <description>...</description>
      <solution>...</solution>
      <severity>4</severity>
    </error>
    ...
  </errors>
  <functionalities>
    <functionality>WaterConsumptionMeasurement</functionality>
    <functionality>GasConsumptionMeasurement</functionality>
    <functionality>StateQuerying</functionality>
    <functionality>ErrorQuerying</functionality>
    <functionality>OnOff</functionality>
  </functionalities>
</device>
```

Figure 8 XML structure to new appliance registering.

5 Results

The result section starts by presenting a proof-of-concept Conversational Assistant, describing two important examples of use of the assistant: chat messages and email. It is followed by a presentation of the results for the first evaluation with users.

5.1 Examples of use

An early user study conducted by the authors [10] revealed that, in most cases, users would like to control the home remotely. Our system enabled users to have a conversation with the Assistant from anywhere using Skype or email. Figure 9 shows a screen of a Skype conversation where the users tries to obtain the water consumption for the kitchen. Skype is intended to support more interactive dialogue while using the email is for simple command and control, such as "Desligue a TV na sala!" (*Turn off the TV in the living room!*) or "Desligue as luzes na cozinha!" (*Turn off the lights in the kitchen!*). This way, the user can select the best suited platform to remotely interact with the Conversational Assistant.

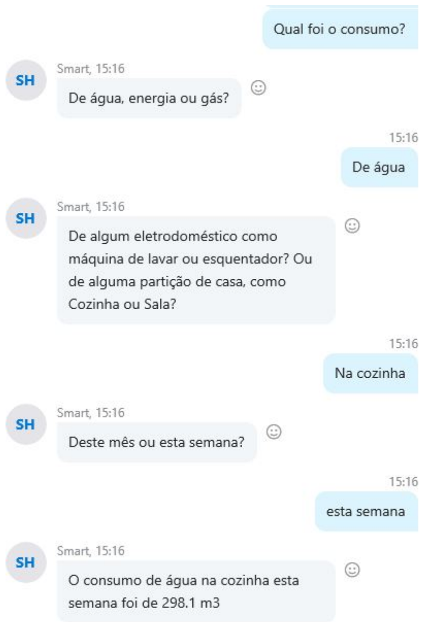

Figure 9 Example of a Skype chat with the conversational assistant: the user is querying the system about water consumption and the assistant is using context.

5.2 First Evaluation

5.2.1 Method

The main goal of the first evaluation with real users was to identify the strongest and weakest points of the first proof-of-concept version of the proposed Smart Home Conversational Assistant. The definition of the evaluation tasks, shown in Table 4, was one of the most challenging steps, since they should lead the participants to explore all the features of the assistant and to discover its dialog capabilities.

Table 4 List of tasks considered for the first evaluation of the proposed conversational assistant.

Nr.	Task
1	Turn on the lights in two home partitions of your choice
2	Find out in which home partition were spent more energy, this month
3	Find out in which home partition were spent more gas, this month
4	Ensure that all lights, in the home, are off.
5	Which TV consumed more energy, this month?
6	Try to find out if exists any issue with the water heater (partition of your choice)
7	If any issue exists, try to understand what is the origin of it and how to resolve it.
8	Compare the water consumption of the current and previous month for the washing machine located in the kitchen
9	Consult the state of any of the appliances.

The first evaluation was conducted with 6 users with diversity of academic backgrounds with ages between 10 and 30 years. In the first step of the evaluation, each participant attended to a brief presentation introducing the Conversational Assistant. The presentation explained its capabilities (obtaining resource consumption, appliance and lights control, and error management) without specifying how to perform these tasks. Additionally, the presentation also included a brief introduction to the virtual home shown in Figure 10, its partitions and virtual devices. Soon after the presentation, each participant performed the required tasks.

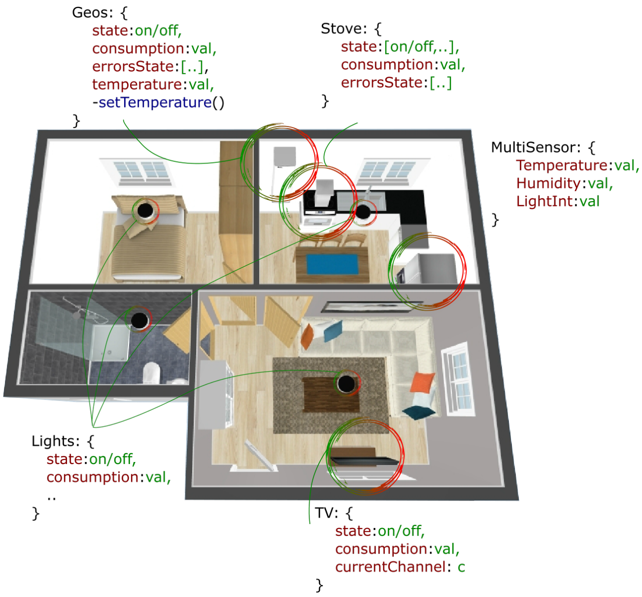

Figure 10 Virtual Smart Home overview.

5.2.2 Evaluation Results

The first evaluation was performed by 6 participants, who did not receive any external help regarding the tasks at hand. While executing each task, the Assistant onboarding mechanisms gave some insights of what the user could ask to obtain certain results.

The results allowed the identification of a few weak points in some modules of the system, namely in the onboarding logic and in the dialogue manager.

Explaining the task to the user without giving him the solution was one of the biggest issues of the evaluation. In some cases, in the first attempt, participants just tried to copy the task description to ask the assistant. The evaluation also showed that most participants gave up easily when they did not get the expected result, while a smaller group tried more than one way to successfully complete the task.

The onboarding system proved to be a valuable help, despite that sentence generation is still rather static. Participants did not explore the full conversational capabilities of the assistant. They often repeated the complete question rather than just completing the missing information. This was more notable in the second and third task, which required more information to make the queries.

6 Conclusion and Future work

This paper presents a first proof-of-concept of a Conversational Assistant for Smart Homes and its first evaluation with users to obtain insights regarding the current strengths and weaknesses and how users explore and use the system capabilities. Despite its stage of development, the Assistant presents unique capabilities and, to best of authors' knowledge, is the first assistant for Smart Homes in (European) Portuguese including dialog capabilities.

Users can interact with the conversational assistant using written or spoken conversation in European Portuguese. The assistant processes the dialog to interpret the commands and requests made by the user. When the user issues a command, the assistant invokes the control service and when the user makes a request it look up in the knowledge base. The proposed ontology provides a structure for the information and enables easy access to it. The ontology is a key component of the system, providing simple methods for the assistant to query the information requested by the user. Using this structure, the information supporting the current implementation of the conversational assistant is stored in a knowledge base, and the ontology was designed to support future additions, such as inhabitants preferences and interactions with the home.

Resulting from the first evaluation, we observed that participants were able to use the system to interact with the house. During the evaluation, some difficulties were perceptible in taking full advantage of the existing features.

Future improvements and new features to the assistant can help users overcome those difficulties, for instance, improving the onboarding system, implementing initiative of the assistant to start a conversation, and adding notifications on critical situations.

References

1 Marjan Alirezaie, Jennifer Renoux, Uwe Köckemann, Annica Kristoffersson, Lars Karlsson, Eva Blomqvist, Nicolas Tsiftes, Thiemo Voigt, and Amy Loutfi. An ontology-based context-aware system for smart homes: E-care@ home. *Sensors*, 17(7):1586, 2017.

2 Suket Arora, Kamaljeet Batra, and Sarabjit Singh. Dialogue System: A Brief Review. *arXiv preprint arXiv:1306.4134*, 2013.

3 Dario Bonino and Fulvio Corno. Dogont-ontology modeling for intelligent domotic environments. In *International Semantic Web Conference*, pages 790–803. Springer, 2008.

4 Dario Bonino, Fulvio Corno, and Luigi De Russis. Poweront: An ontology-based approach for power consumption estimation in smart homes. In *International Internet of Things Summit*, pages 3–8. Springer, 2014.

5 Dario Bonino and Giuseppe Procaccianti. Exploiting semantic technologies in smart environments and grids: Emerging roles and case studies. *Science of Computer Programming*, 95:112–134, 2014.

6 Dan Brickley and Libby Miller. FOAF vocabulary, 2014. URL: http://xmlns.com/foaf/spec/.

7 Simon Cox and Chris Little. Time Ontology in OWL, 2017. URL: https://www.w3.org/TR/owl-time/.

8 Javier Cuenca. Ontology for Energy Management Applications, 2018. URL: https://innoweb.mondragon.edu/ontologies/oema/ontologynetwork/1.1/index-en.html.

9 Aldo Gangemi. DUL, DOLCE+DnS Ultralite ontology. http://www.ontologydesignpatterns.org/ont/dul/DUL.owl. Accessed: 2019-03-05.

10 Maksym Ketsmur, Nuno Almeida, António Teixeira, and Samuel Silva. Contribute for an Ontology for Smart Homes and their Conversational Assistants. *Submitted and under review*, 2019.

11 Jangho Lee, Gyuwan Kim, Jaeyoon Yoo, Changwoo Jung, Minseok Kim, and Sungroh Yoon. Training IBM Watson using Automatically Generated Question-Answer Pairs. *arXiv preprint arXiv:1611.03932*, 2016.

12 Cathy Pearl. *Designing Voice User Interfaces: Principles of Conversational Experiences*. O'Reilly, 2016.

13 E. V. Polyakov, M. S. Mazhanov, A. Y. Rolich, L. S. Voskov, M. V. Kachalova, and S. V. Polyakov. Investigation and development of the intelligent voice assistant for the Internet of Things using machine learning. In *2018 Moscow Workshop on Electronic and Networking Technologies (MWENT)*, pages 1–5. IEEE, 2018.

14 Martin Porcheron, Joel E Fischer, Stuart Reeves, and Sarah Sharples. Voice interfaces in everyday life. In *proceedings of the 2018 CHI conference on human factors in computing systems*, page 640. ACM, 2018.

15 Eric Prud'hommeaux and Andy Seaborne. SPARQL Query Language for RDF, 2008. URL: https://www.w3.org/TR/rdf-sparql-query/.

16 Nazaraf Shah, Kuo-Ming Chao, Tomasz Zlamaniec, and Adriana Matei. Ontology for Home Energy Management Domain. In *Digital Information and Communication Technology and Its Applications: International Conference, DICTAP*, volume 167, pages 337–347, Dijon, France, June 2011. doi:10.1007/978-3-642-22027-2_28.

17 Alexander Smirnov, Alexey Kashevnik, and Andrew Ponomarev. Multi-level self-organization in cyber-physical-social systems: smart home cleaning scenario. *Procedia Cirp*, 30:329–334, 2015.

18 Lorenzo Sommaruga, Antonio Perri, and Francesco Furfari. DomoML-env: an ontology for Human Home Interaction. In *Swap*, volume 166. Citeseer, 2005.

19 Ming Tao, Kaoru Ota, and Mianxiong Dong. Ontology-based data semantic management and application in IoT-and cloud-enabled smart homes. *Future generation computer systems*, 76:528–539, 2017.

20 Ming Tao, Jinglong Zuo, Zhusong Liu, Aniello Castiglione, and Francesco Palmieri. Multi-layer cloud architectural model and ontology-based security service framework for IoT-based smart homes. *Future Generation Computer Systems*, 78:1040–1051, 2018.

21 Juan Ye, Graeme Stevenson, and Simon Dobson. A top-level ontology for smart environments. *Pervasive and mobile computing*, 7(3):359–378, 2011.

22 Steve Young, Milica Gašić, Blaise Thomson, and Jason D Williams. POMDP-based statistical spoken dialog systems: A review. *Proceedings of the IEEE*, 101(5):1160–1179, 2013.