Indoor Domain Model for Dialogue Systems

Porfírio Filipe[1,2,3] and Nuno Mamede[1,4]

[1] LF INESC-ID Lisboa– Spoken Language Systems Laboratory, Lisbon, Portugal
[2] GuIAA – Grupo de Investigação em Ambientes Autónomos, Lisbon, Portugal
[3] ISEL – Instituto Superior de Engenharia de Lisboa, Lisbon, Portugal
[4] IST – Instituto Superior Técnico, Lisbon, Portugal
{porfirio.filipe,nuno.mamede}@l2f.inesc-id.pt

Abstract. The design of spoken language applications that allow people to talk with machines/computers, in the same way that they talk with each other, is materialized as a Spoken Dialogue System (SDS). This paper presents a knowledge modeling approach to allow spontaneous configuration of SDS. Our approach focus on the representation and management of the domain knowledge that is aggregated at runtime and aims to update the dialogue management strategy. To do so, one developed an autonomous Environment Interaction Manager (EIM). When working on the indoor environment, the domain knowledge reflects the plan of the building and the SDS controllable resources. The building is modeled as a dynamic aggregation "part-whole" of controllable resources. Each resource owns and shares a semantic interface that makes available its task set manipulated by the SDS. These ideas have been applied with success in our lab, modeling the semantic interface aggregation under the semantic web vision.

Keywords: Human–Computer Interaction, Spoken Dialogue System, Natural Interaction, Smart Environments, Domain Model.

1 Introduction

Smart environments is a technological concept that, according to Mark Weiser is "a physical world that is richly and invisibly interwoven with sensors, actuators, displays, and computational elements, embedded seamlessly in the everyday objects of our lives, and connected through a continuous network" [3]. Consequently, the computational model of a smart environment can be viewed as a large collection of networked heterogeneous devices. For start, one have to refer the terminology typically used to designate two key concepts within the environment, which are Device and Service. Devices and services are the entities that participate in environment. "Devices" includes conventional computers, small handheld computers (PDAs), printers, and more specialized network devices, such as a thermometer or household appliances. "Services" includes any sort of network service that might be available. In fact, most devices are represented on the network by one or more services. Furthermore, a single network attached device may implement several services, e.g., a network printer may provide printing and fax (and who knows what else), all in a single device. In this context, devices and services are considered

© Springer-Verlag Berlin Heidelberg 2009

essentially equivalent and interchangeable. Together, these will be termed “Entities” or “Resources” on the network. Resources must interoperate with other resources minimizing pre-existing knowledge.

A smart environment can be characterized by the following basic elements: ubiquity, awareness, intelligence, and natural interaction. Ubiquity refers to a situation in which we are surrounded by a multitude of inter-connected embedded systems, which are invisible and moved into the background of our environment. Awareness refers to the ability of the system to locate and recognize objects and people, and their intentions. Intelligence refers to the fact that the digital surrounding is able to analyze the context, adapt itself to the people that live in it, learn from their behavior, and eventually to recognize as well as show emotion. Finally, natural interaction refers to advanced modalities like natural speech and gesture recognition, as well as speech synthesis, which will allow a much more human like communication with the digital environment than is possible today [5].

The design of natural language applications that allow people to talk with machines/computers, in the same way that they talk with each other, is materialized under the form of a Spoken Dialogue System (SDS), having constituted a natural interface, where the use of speech is privileged.

The research of SDS is commonly considered a branch of human-computer interaction, although its origins are generally rooted in the automatic speech recognition community. Current trends are putting more research emphasis on aspects of psychology and linguistics.

Speech-based human-computer interaction faces several challenges in order to be more widely accepted. One of these challenges is the domain portability. In order to face domain portability one assumed that practical dialogue and domain-independent hypothesis are true [1]. The reason is that all applications of human computer interaction involve dialogue focused on accomplishing some specific task. We consider the bulk of the complexity in the language interpretation and dialogue management is independent of the task being performed. In this context, a clear separation between linguistic dependent and domain dependent knowledge allows reducing the complexity of SDS typical components.

Summarizing, our contribution enables customization of SDS, within a smart environment scenario, to provide speech natural interaction. Section 2 gives an overview of the proposed approach. Section 3 gives an overview of the most relevant components of the knowledge model. Section 4 describes the Knowledge Aggregation Process (KAP).

2 Approach

An SDS can be used to access a domain database, being the final answers produced based on the external data source [9]. The traditional interaction cycle starts with the user’s request that is captured by a microphone, and provides the input for the Speech Recognition component. Next, the Language Understanding component receives the recognized words and builds the related speech acts. The Dialogue Manager (DM) processes the speech acts and then calls the Response Generation component to generate a message. Finally, the message is used by the Speech Output component to produce speech. The response of the SDS is final or is a request for clarification.

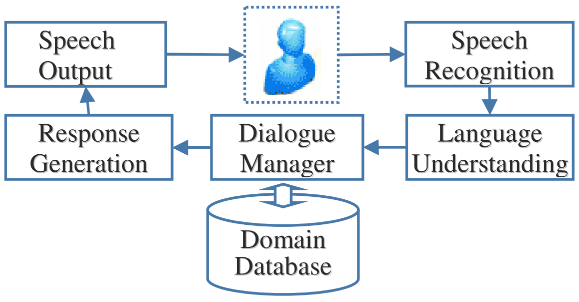

Fig. 1. Logical flow through SDS components

Figure 1 shows a typical logical flow through SDS components architecture to access a domain database.

Within a smart environment scenario, a SDS should be a computational entity that allows access to any resource by anyone, anywhere, at anytime, through any media or language, allowing its users to focus on the task, not on the tool [10].

Nevertheless, a traditional SDS cannot be directly used within a smart environment scenario because of is lack of dynamic portability, in view of the fact that SDSs are not ubiquitous yet [12]. Dynamic portability demands for spontaneous configuration. Within a ubiquitous domain, one does not know, at design time, all the resources that will be available. To address this problem we propose an approach for ubiquitous knowledge modeling, which was introduced in [6] and improved in [7], [8].

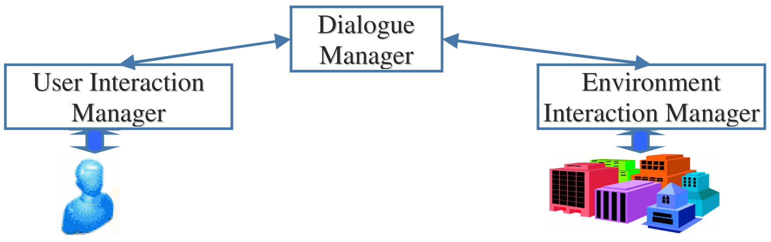

Fig. 2. SDS customization via EIM

The domain customization and spontaneous configuration of the SDS is achieved by the, proposed, Environment Interaction Manager (EIM), see Figure 2.

The main goal of the EIM is to support the communication interoperability between the SDS and a set of heterogeneous resources, performing knowledge management. For this, the EIM includes a knowledge model that represents all the aggregated resource semantic interfaces.

When working on the indoor environment, the knowledge reflects the plan or physical organization of the building and the SDS controllable resources. The building is modeled as a dynamic aggregation "part-whole" of controllable resources [4]. Each resource shares a semantic interface that makes available its capabilities set that makes possible its control by the SDS. The building itself is a controllable resource that aggregates resources such as floors, rooms, entrance halls, foyers, etc. Each one of these resources aggregates the resources that controls, such as, doors, windows, elevators, environment controls, multimedia controls, appliance controls and so on.

When a resource, designated by part, is activated, a discovery protocol determines the nearest resource, designated by whole. After that, KAP executes the aggregation merging the knowledge built in the semantic interface of the part. Finally, KAP propagates the changes to related building parts.

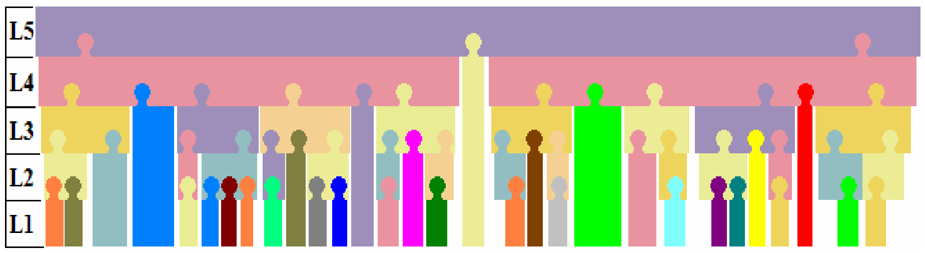

Fig. 3. Aggregation Levels

Figure 3 presents several aggregation levels (L1 to L5). A resource is distinguished by a different color an is aggregated to another resource existing in an upper level. The resource at L5 level aggregates all the existing resources representing, for instance, an entire building.

3 Semantic Interface

This section gives an overview of the most relevant components of the semantic interface knowledge model that includes four independent knowledge components: the discourse model, the task model, the world model, and the event model.

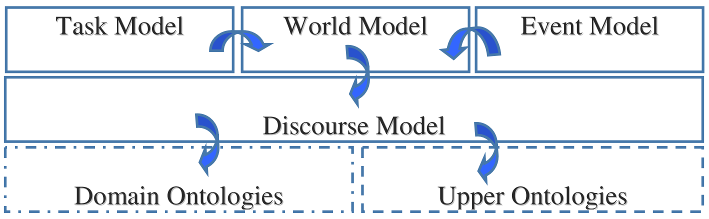

Fig. 4. Semantic Interface Knowledge Model

Additionally, we are also considered external ontological knowledge components to exemplify domain ontologies and upper ontologies, see Figure 4.

3.1 Discourse Model

The discourse model defines a conceptual support, grouping concept descriptions, used to express SDS controllable resources. The mining of a concept is previously accorded or is inferred comparing its linguistic knowledge or the resources pointed by a Uniform Resource Identifier (URI), for instance external ontology nodes.

A concept description is a knowledge atomic unit and maps linguist knowledge into domain knowledge. Essentially, a concept maps a set of URIs into a set of terms or more generically into a set of Multi-Word Unit (MWU) [10]. Concepts have linguistic and semantic parts. Concept descriptions are organized according to main types that are "task", "role", "event", "name", and "constant". Concepts with types "action" or "perception" hold task names. A perception task cannot modify the state of the environment, on the other hand an action task can. Concepts with types "collection" or "quantity" hold task roles (parameters or arguments). The type "collection" is used to define sets of constants (represented also by concepts such as white, black, red, …) to fill task roles (color, shape, texture, …). The type "quantity" is about numbers (integer, real, positive, …) and the "unit" type is for measures (time, power, …). The type "event" holds event names. The type "name" holds resources names. Figure 5 shows a mind map for a concept description.

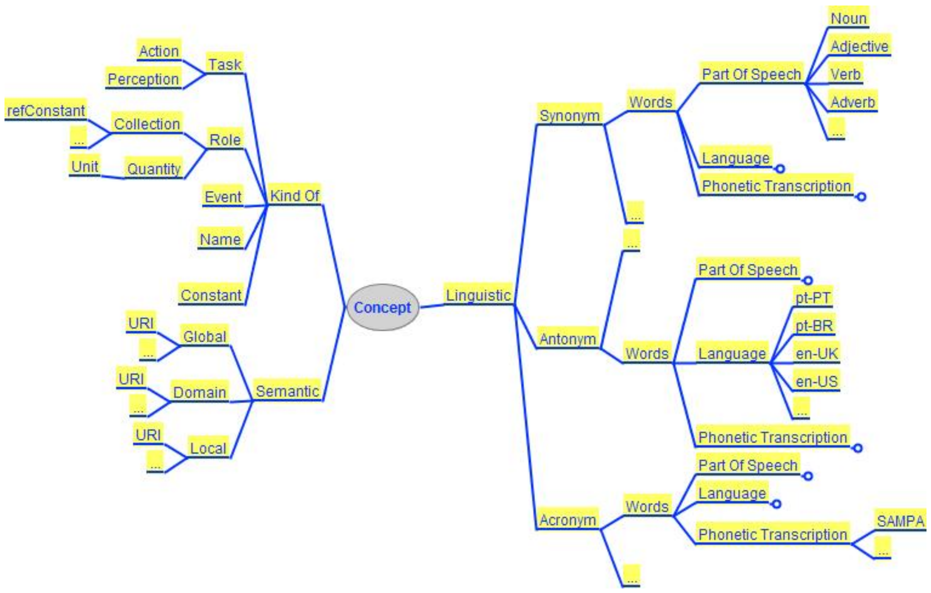

Fig. 5. Concept Description

In order to guarantee the availability of vocabulary to refer the domain's concepts, concept descriptions include linguistic properties. Each Word (or term), has a part of speech tag, such as noun, adjective, verb, adverb; a language tag, such as "pt-PT", "pt-BR", "en-UK" or "en-US"; and a optional phonetic transcription. The linguistic description holds a list of words, or more generically a MWU, referring linguistic variations associated with the concept, such as synonymous, antonymous or acronyms.

Concept descriptions have also semantic references characterized by Universal Resource Identifier (URI). The semantic description supports references to domain knowledge sources (domain hierarchy, domain ontologies) or global knowledge sources, (upper ontologies or a lexical database, such as WordNet). A concept description must include at least one URI for local reference.

3.2 Task Model

The task model contains one or more descriptions, based in concepts previously declared in the discourse model, which characterize resource capabilities or tasks. Figure 6 shows a mind map for a task description.

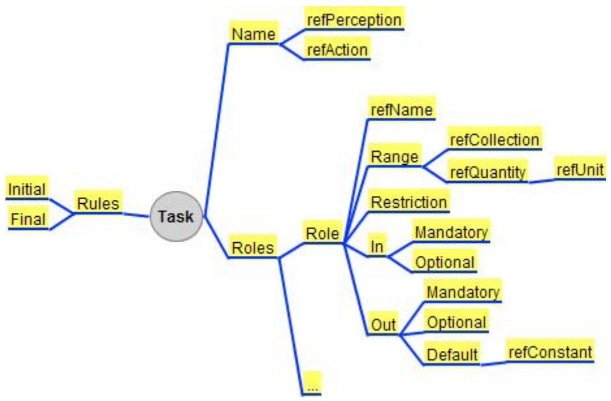

Fig. 6. Task Description

A task description is a semantic representation that has a task name and, optionally, a role input and/or output list. A role describes an input and/or output task argument or parameter. An input role has a name, a range, and a restriction. The role restriction is a rule that is implemented as regular expression and is optional. An output role is similar to an input role with an optional default constant. The initial and final rules perform environment state validation: the initial rule (to check the initial state of the world before a task execution) and the final rule (to check the final state of the word after a task execution). These rules, implemented also as regular expressions, can refer role names and constants returned by perception task calls.

3.3 World Model

The world model contains descriptions about one or more resources (the "whole" and its "parts") that refer concepts previously declared in the discourse model. These descriptions include name, serial number, and optionally physical properties (color, shape, …) that are known by the SDS user and are typically used to indentify and resource within a user request. Optionally, a resource description refers one or more classes symbolized in domain ontologies.

3.4 Event Model

The event model contains descriptions about events supported by concepts previously declared in the discourse model. These descriptions are similar to task descriptions only with name and input roles. An event is a notification about an expected or unexpected environment state modification. The event model supports the reactive behavior of the EIM notifying the SDS dialogue manager.

4 Knowledge Aggregation

The goal of the Knowledge Aggregation Process (KAP) is to update on-the-fly the knowledge model of a resource semantic interface "whole", merging the knowledge originated by one "part", that is also a resource. We assume that each resource has its own semantic interface built in or is virtualized by an external computational entity.

At its starting point, KAP puts side by side concepts and tasks descriptions using similarity criteria:

- (a) Two concepts are similar when its domain or global URIs is the same or its linguistic descriptors are literally equal. When the type of the concepts is collection, its constants must be also similar;
- (b) Two tasks are similar when its descriptions are literally equal;

In order to update the semantic interface of the "whole" KAP follows the next four steps:

- i) For each concept in "part", without a similar (a) in "whole", is added a new concept description to "whole" discourse model;
- ii) For each task in "part" without a similar (b) in "whole" is added a new concept description to "whole" task model;
- iii) A new resource description is added to "whole" world model;
- iv) The resource description is linked to the updated tasks descriptions.

5 Experimental Setup

The experimental setup is based on our simulator, originally developed for Portuguese users. This simulator incorporates a basic dialogue manager and several independent simulators, such as a microwave oven, a fryer, freezer, a lamp and a window.

Actually, we have about one thousand concepts and one hundred tasks. The simulator allows the debug of an invoked task analyzing the interaction with the target resource. We can execute KAP, execute tasks, and observe its effects on the environment. We can also consult and print several data about the each one of the semantic interface knowledge models.

The indoor environment is characterized by an arbitrary set of resources typically supported by augmented artifacts, such as appliances or furniture. The type hierarchy does not need to be complete because it can be improved, as new resources are dynamically added. Figure 7 shows part of the domain ontology (type hierarchy) for a kitchen environment.

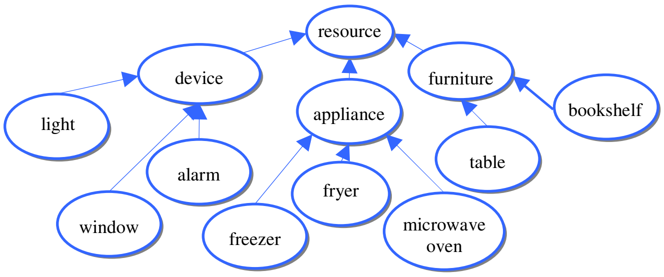

Fig. 7. Part of Kitchen Type Hierarchy

6 Concluding Remarks and Future Work

The growth in pervasive computing will require standards in device communication interoperability. These devices must be able to interact and share information with other existing devices and any future devices across the network. Current technologies require human interventions to solve environment reconfiguration problems. We believe that these technologies must be improved with more human like ways of interaction including spoken natural language support.

Our proposal tries do improved the flexibility of the SDSs architectures allowing the independent and collaborative design of semantic interfaces used to describe resource capabilities. We are proposing KAP (knowledge aggregation process) to deal at runtime with part-whole associations aggregating a completely new resources achieving “Strong PnP”.

The presented ideas have been applied with success in our lab to a set of resources that represents part-whole associations, within an indoor environment scenario, modeling the semantic interface aggregation under the semantic web vision [2].

Now, our work is based in the kitchen environment. However, we intend to generalize the use of the SDS natural interface to support inhabitants’ activities, for instance, to optimize climate and light controls, item tracking and automated ordering for food and general use items, automated alarm schedules to match inhabitants’ preferences, and control of media systems.

In the near future, we aim to study more deeply the behavior of the rate knowledge replication versus knowledge integration. We expect to prove for the upper aggregations levels (see Figure 3) an interesting knowledge integration rate because of the reuse of similar concepts and tasks descriptions.

References

1. Allen, J., Byron, D., Dzikovska, M., Ferguson, G., Galescu, L., Stent, A.: An Architecture for a Generic Dialogue Shell. Natural Language Engineering 6(3-4), 213–228 (2000)
2. Cardoso, J.: The Semantic Web Vision: Where are We? In: IEEE Intelligent Systems, September/October 2007, pp. 22–26 (2007)
3. Cook, D., Das, S.: Smart Environments: Technology, Protocols and Applications. Wiley Interscience, Hoboken (2004)
4. Daille, B., Gaussier, E., Lange, J.: Towards Automatic Extraction of Monolingual and Bilingual Terminology. In: COLING 1994, pp. 515–521 (1994)
5. Filipe, P., Barata, M., Mamede, N., Araújo, P.: Enhancing a Pervasive Computing Environment with Lexical Knowledge. In: 2nd International Conference on Knowledge Engineering and Decision Support, Lisboa, Portugal, pp. 307–312 (2006)
6. Filipe, P., Mamede, N.: Towards Ubiquitous Task Management. In: 8th International Conference on Spoken Language Processing, Jeju, Coreia, pp. 3085–3088 (2004)
7. Filipe, P.: Dynamic Integration of Artifacts in Dialogue Systems. Ph.D Thesis, Technical University of Lisbon UTL/IST (2007)
8. Filipe, P., Mamede, N.: Empirical Multi-Artifact Knowledge Modeling for Dialogue Systems. In: 11th International Conference on Enterprise Information Systems, Barcelona, Spain, pp. 286–292 (2008)
9. McTear, M.: Spoken Dialogue Technology. Springer, Heidelberg (2004)
10. Minker, W., López-Cózar, R., McTear, M.: The Role of Spoken Language Dialogue Interaction in Intelligent Environments. Journal of Ambient Intelligence and Smart Environments 1, 31–36 (2009)
11. Sohn, K., Lee, K., Kim, T., Lee, S., Kim, J.: Resource Sharing Using RDF in Ubiquitous Smart Space. In: International Conference on Convergence Information Technology (2007)
12. Weiser, M.: The Computer of the 21st Century. Scientific American 265(3), 66–75 (1991)