Towards Ubiquitous Task Management

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

[1] L²F INESC-ID - Spoken Languages Systems Laboratory, Lisbon, Portugal
{porfirio.filipe, nuno.mamede}@l2f.inesc-id.pt
[2] ISEL - Instituto Superior de Engenharia de Lisboa, Lisbon, Portugal
[3] IST - Instituto Superior Técnico, Lisbon, Portugal

Abstract

In the near future people will be surrounded by intelligent devices embedded in everyday objects where the knowledge and understanding of device attributes and capabilities will be a key enabler.
This paper describes the current state of our research in designing distributed knowledge based devices as a solution to adapt spoken dialogue systems within ambient intelligence. In this context a spoken dialogue system is a computational entity that allows universal access to ambient intelligence for anyone, anywhere, at anytime to use any device through any media. Our aim is to build knowledge-based devices to enable dynamic adaptation of the components integrated in the dialogue system architecture. An example focusing household appliances is depicted.

1. Introduction

Ambient Intelligence (AmI) is a vision where a pervasive and unobtrusive intelligence in the surrounding environment supports the activities and interactions of the users. This vision has the potential to fundamentally change our world combining three key technologies: ubiquitous (or pervasive) computing [1] [2], ubiquitous communication and intelligent user interfaces. Ubiquitous communication enables generic objects to communicate with each other and with the user by means of ad-hoc and wireless networking. Actually, the environment supports a network of sensors, computing devices and information appliances. Furthermore, this means integration of microprocessors into everyday objects like household appliances, furniture, clothing, toys and even paint. Networked computing devices will proliferate in this landscape, and users will no longer be tethered to a single computing device. People on the move will become networks on the move as the devices they carry network together and connect with the different networks around them [3].
The nature of devices will change to form augmented environments in which the physical world is sensed and controlled in such a way that it becomes merged with the virtual world [4]. This massive use of such devices will allow filling the gap between cyberspace and the real world.
Design Spoken Dialogue System (SDS) working on networks of heterogeneous devices with a continuously changing landscape has a major difficulty in dealing with interoperability and integration problems. In this context a static definition of application domain does not make; sense we might only say: “the domain is a set of heterogeneous devices, with dynamic cardinality, spread in AmI”.

2. Background

Speech and Language Processing includes several other related research fields known as Natural Language Processing, Computation Linguistics, and Speech Recognition and Synthesis [5]. The origins of SDSs can be traced back to Artificial Intelligence (AI) research in the 1950s concerned with developing conversational interfaces. However, it is only within the last decade or so, with major advances in speech technology, that large-scale working systems have been developed and, in some cases, introduced into commercial environments [6]. Since the first SDSs, developed under American and European research projects, issues on portability, extensibility, scalability and reusability still remaining as active research. These issues are addressed typically through architectures that allow the integration of reusable components. Nevertheless, there is still significant work required to build a SDS that is expressed in practical terms, by large difficulties in the integration and reutilization of resources or components even with similar application domain. The integration of components into a working system is still a key issue of actual research [6]. The use of SDS into AmI can only worsen the former problems transforming them into a major challenge. Recent progresses can be seen in [7] [8] [9] [10] [11].

3. Spoken Dialogue Systems at Home

In this paper a SDS is a computational entity that allows universal access to AmI for anyone, anywhere, at anytime to use any device through any media.
Under the major topic that is AmI we are mostly interested on home environments as a particular example of other spaces such as the office, the car or public spaces. The devices throughout the house can be in constant contact with each other, making the AmI home responsive to all its inhabitants' needs. These devices must be easily installed and personalized according to the users' wishes. The AmI home is also energy-conscious, able to intelligently manage its use of heat, light and other resources depending on the occupants’ requirements.
The exponential drop in microprocessor cost over time has enabled appliance manufacturers to pack increasingly complex features into appliances such as video recorders, refrigerators, washing machines and air conditioner. As household appliances grow in complexity and sophistication, they become harder and harder to use, particularly because of theirs tiny display screens and limited keyboards. In fact, this can be seen in the growing amount of information on manuals and inscriptions or symbols on the appliance itself. The SDSs have here an opportunity to show what they can do for its users. They can manage this amount of rising technical information

and help users to directly invoke tasks as a way to solve a growing interface problem in a effortless style.
According to [12], is demonstrated convincingly that interactions with computers and new technologies are identical to real social relationships and to the navigation of real physical spaces. In this context it is reasonable to disclose, for instance, that people will talk naturally with its microwave oven.
Still, at the moment, it is unrealistic to consider for each device the existence of an autonomous SDS because it is much more difficult to deal with coordination and collaborations problems and due to device hardware limitations.
The use of a SDS admits non-technical users with no a priori knowledge of the environment. But, if the dialogue is conducted straight to the device the user is more oriented.
It is frequent in multimodal SDSs the use of simulated characters [13] [14]. Nevertheless, we believe that in an increasing number of cases, these characters may be replaced by real devices with intelligent behavior.
We consider that the major challenge is to deal with tasks involving collaboration and selection between several devices. Dealing with isolated devices is a first important step. Within a sophisticated home environment it is not an important feature turning on the light through a voice command, since the light will be turned on just by the user's presence. However, it will be very useful to negotiate the room's luminosity, when the order "*more light*" is given, as the SDS (automatically at day) will change the transparency of the window, built with electrochromic materials [15], instead of acting over an artificial source of light. The SDS might take the initiative asking the user if they want to leave a light on as they leave the house.
We think that the mixed-initiative dialogue will emerge not only from isolated devices but also fundamentally from its collaboration. For instance, a sensor in a window sends an broken alarm "*window open*", at the same time to the security system, to the sound system and to the control environment system (to turn off the air conditioner in the room). In this context the SDS must be aware of any alarm, measure (i.e. temperature, wind) or command that may determine the behavior of any device (or peripheral system) in order to be ready to suggest actions (adding not predicted content to the discourse) or just to provide answers when asked.

4. Knowledge base approach

From the theoretical point of view, some of the most important contributions in the evolution of knowledge representation are presented in [16] [17], using a layered representation. A new level, the ontological level, has been identified in [18] which aim is to constrain knowledge primitives and thus build more understandable and consistent ontologies.
Knowledge acquisition is known as a serious bottleneck in building knowledge-based systems, since it is difficult to elicit expertise from domain experts. However, successful natural language understanding requires knowledge about the reasoning domain. Within a ubiquitous domain we do not know, at design time, all the available devices and which tasks they execute. We decided, as a basic strategy to work within a, ubiquitous domain, to split the domain knowledge in global and local knowledge under a device-centered perspective.
The global knowledge is built-in in the SDS domain model (that contains the task model) and includes the common sense knowledge (strictly needed), all the domain specific knowledge and the knowledge about the most common classes of devices in the domain.
The local knowledge is distributed into all available devices and includes knowledge about the related classes of devices and knowledge about the class device itself: knowledge about particular attributes (color, shape, dynamic position, etc), user profile (for security and privacy policy) and other linguistic and phonetic knowledge. This strategy is presented as an open specification, which can contain any other unclassified knowledge, if it serves the special needs of a particular component integrated in any architecture.
These two kinds of knowledge (global and local) will be integrated on the fly. This approach is inspired on ONIONS [19] that is a methodology for integrating ontologically heterogeneous taxonomic knowledge. In this paper we focus particularly on local knowledge acquisition process to support ubiquitous task management. A recent work that uses ontologies in SDS design can be seen in [20] [21].

5. Knowledge Based Devices

Our aim is to build devices that support the required knowledge to enable dynamic adaptation of the components integrated in the SDS architecture. We are truly convicted that to achieve a really natural dialogue, it is fundamental to prepare the SDS with knowledge about the devices and supported tasks and with dialogue capabilities as well. The specific device dependent knowledge (local domain knowledge) was separated and maintained attached to the device itself, because we cannot predict at design time all the system's needs since we do not know all the available devices.
The use of a simple representation of a generic device, supported for instance in XML, is a common way to enable the adaptation of a device into the SDS architecture [22]. Nowadays, as you can see in [23], in demonstrations is commonly used three kinds of devices: switchable, dimmable and sensors. Switchable devices are binary state devices that can be set or queried. Dimmable devices have a state varying on a single scalar dimension, which can be set, changed, or queried. Sensors are similar but can only be queried. However, if these devices were more sophisticated the gap between the execution API (in XML) and the needed knowledge about the device is clearly exposed. When one is dealing with simple devices, the approach seems to be enough, but if one tries to adapt a small household appliance, like a microwave oven, into a SDS architecture one is able to see that the interaction will become far from being natural.
At this point, we have to stop and think about the main capabilities of such devices and about a really useful way to address them within a SDS. Generally, we have to find the right answer for several questions like: "What is achieved if one adapts a SDS to this device?", "What kind of manipulation is needed?", "What kind of feedback is needed?", "How can the system accomplish it?" and "What does it means in a dialogue style?", "How natural is the dialogue?" and "Does the result match the user's expectations?".

6. Integrating Linguistic Parts into Device Task Model

A widespread method to adapt a dialogue system to a new domain is to map the words in the lexicon straight to concepts within the domain model [24]. We argue that each device should represent generic concepts complemented with linguistic parts as a way to allow task management in AmI. A concept is mapped to a regular expression at the highest abstract level. The regular expressions are used to match and validate multi-word units (one or more words) aligned with the domain task model that defines a semantic interface. Each concept is linked to a multi-word unit list. This list contains linguistic variations related with the concept such as synonymous, acronyms as well as other multilingual equivalent. A task descriptor is expressed by generic concepts that are: task name, task arguments and return concept. Usually, a task name is related with a verb and its arguments with nouns and adjectives. We defined a set of specific tasks with the name started by "get" followed by an attribute name to access particular attributes of the device, like color, shape and so on. The task return concept may be used (as semantic representation) by a language generator component to produce feedback to the user. Each task has a generic pre-condition that allows or disallows its execution. This depends on the execution state of the device and on the user profile represented as a list of concepts that should be satisfied. The task arguments that cannot be enumerated, like quantities and dates, have a special management using validation rules to check their range and format.

At runtime, the local taxonomic knowledge is integrated with the global taxonomic knowledge (using knowledge about the related concepts) and the local tasks (with names W and U where U appears twice) are added to the global task list, see Fig. 1. The task name is the first concept in the task descriptor, the task return concept is the last and in the middle are optionally represented the classes of each argument.

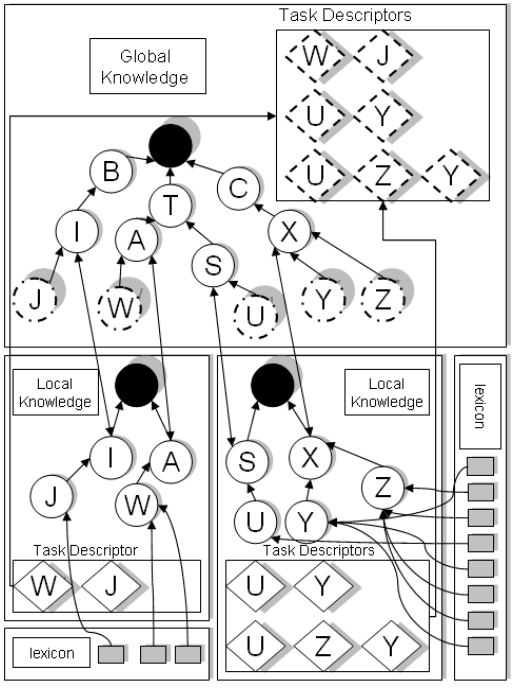

Figure 1: Schematic view of the local knowledge integration

With this task model we can determine the device tasks available and the set of values that can fill any particular argument or its validation rule, allowing at the same time a loose coupling between task and dialogue management.

7. Talking With my Microwave Oven

Microwave ovens are used to heat up and defrost food. But they are also used for cooking vegetables, fruit, fish and poultry. After an analysis of several models we concluded that it is necessary to establish values for two input selectors: time and power. The user interacts with this device not only with the hands but with other senses as well: vision, hearing and smelling. So, the SDS should know when the door is open or closed, when the oven is switched on and when a local user attempts to manipulate the time and power selectors. According to this scenario it was defined the execution interface (primitive task set) of a standard microwave oven, which is presented at Table 1.

Table 1: Primitive task set

Task	Argument	Return
Starting	-	Boolean
Stopping	-	Boolean
Selecting	Time	Time
Selecting	Power	Power
isClosed	-	Boolean
isSwitchedOn	-	Boolean
isInUse	-	Boolean

In a first step we have to adapt (build a wrapper) the microwave oven (moulinex – micro-chef 900) so that it presents the minimal execution interface. Furthermore we define the range of Time [0, 30] min and Power [100, 900] Watt arguments of the task "Selecting".

In a second step we have to extend the basic knowledge about power by relating it with cooking notions. Table 2 presents the relation between power and some cooking processes. Depending on the type of food and its amount the values present in the Table 2 should be adjusted.

Table 2: Power and task

Task	Power
Cooking	900 W
Reheating	900 W
Defrosting	500 W
Keeping	100 W

In a last step, we continue to extend the local knowledge about cooking using a microwave oven introducing relations of time, types of food, task and a reference amount (we have to be careful with unit's conversions). Some of these relations are illustrated in Table 3.

Table 3: Time, food, task and amount

Time	Food	Task	Amount
8 min	Cod Steaks	Defrosting	2*400g
4 min	Shelled prawns	Defrosting	200g
9 min	Roast beef	Cooking	1Kg
8 min	Carrots	Cooking	400g
30 sec	Rice	Reheating	150g
30 sec	Coffee	Reheating	3.5 fl oz

This process might continue adding all the knowledge about the tasks we need in order to achieve a more natural

dialogue based on a well-done design task model. This knowledge might also be completed with safety recommendations like "never operate the oven when empty". At last we present some lines in Table 4 that illustrate the use of primitive tasks, presented at Table 1, combined with the acquired knowledge expressed in Table 2 and Table 3.

Table 4: The local task descriptors

Task	Composition
Cooking	Selecting((Power) Table2('Cooking'))
Reheating	Selecting((Power) Table2('Reheating'))
Defrosting(food)	Selecting((Power) Table2('Defrosting')) Selecting((Time)Table3(food, 'Defrosting'))
IsOpened	Not isClosed

With this task model we can give directly the order "cook cod steaks" and then the time selection is of 8 min and the power selection is of 900 W. A confirmation of the amount of food may be required.

8. Discussion

Without any standards to represent task knowledge one can only hope that progresses in semantic web research will help to define the required standards to deal with pervasive computing environments. A contribution in that direction can be seen in [25]. We could use the FIPA device ontology [26] to represent memory type, connection, hardware description, software description and so on. However, generally, this kind of information is not relevant for task management because the user will not be particularly interested in asking about that. On the other hand, this ontology may be useful to define the physical device specification as a standard way.

9. Conclusions

In this paper, we have devised a strategy to adapt, on the fly, the components of SDS architecture trough the design of knowledge-based devices. We have presented this subject focusing on task management within a future computing vision. We have implemented a simulator of a common household appliance, which is the microwave oven. Under this simulator we have tested with success the proposed task model that defines a semantic interface.

10. References

[1] Weiser, M. "The Computer of the 21st Century", Scientific American, vol. 265, no. 3, 1991, pp. 66-75.

[2] Weiser, M. "The world is not a desktop", ACM Interactions, 1, 1, 7-8 1994.

[3] Ref: Europe's Information Society Thematic Portal, http://europa.eu.int/information_society/policy/ambientI/index_en.htm. 2004.

[4] Henricksen, K., Indulska, J., Rakotonirainy, A. "Infrastructure for Pervasive Computing: Challenges", Workshop on Pervasive Computing Informatik 01, Viena, September 2001.

[5] Jurafsky, D., Martin, J. "Speech and Language Processing: An Introduction to Natural Language Processing", Prentice-Hall, 2000.

[6] McTear, M. "Spoken Dialogue Technology: Enabling the Conversational Interface", ACM Computing Surveys, 34,1, March 2002:90-169, 2002

[7] Turunen, M., Hakulinen, J. "JASPIS^2 – An Architecture for Supporting Distributed Spoken Dialogues", Eurospeech 2003: 1913-1916, 2003.

[8] Bohus, D., Rudnicky, A. "RavenClaw: Dialog Management Using Hierarchical Task Decomposition and an Expectation Agenda", Eurospeech 2003:597-600, Genéve, Switzerland, 2003.

[9] Polifroni, J., Chung, G. "Promoting Portability in Dialogue Management", ICSLP 2002:2721-2724, 2002.

[10] Pakucs, B. "Towards Dynamic Multi-Domain Dialogue Processing", Eurospeech 2003:741-744, 2003.

[11] O'Neill, I., Hanna, P., Liu, X., McTear, M. "The Queen's Communicator: An Object-Oriented Dialogue Manager", Eurospeech 2003:593-596, Genéve, Switzerland, 2003.

[12] Reeves, B., Nass, C. "The Media Equation: How People Treat Computers, Television and New Media Like Real People and Places", Cambridge, Mass: Cambridge University Press. 1996.

[13] Gustafson, J., Lindberg, N., Lundeberg, M. "The August Spoken Dialogue System", Eurospeech 1999, 1999

[14] Gustafson, J., Bell, L., Beskow, J., Boye, J., Carlson, R., Edlund, J., Granström, B., House, D., Wirén, M. "AdApt - A Multimodal Conversational Dialogue System", ICSLP 2000, (2) 134-137, Beijing, China, 2000.

[15] Wigginton, M. "Glass in Architecture", Phaidon Press Ltd, London, 1996.

[16] Brachman, RJ. "On the Epistemological Status of Semantic Networks", NV Findler, Ed., Associative Networks: Representation and Use of Know/edge by Computers, Academic Press, New York, pp. 3-50, 1979.

[17] Newell, A. "The Knowledge Level", Artificial Intelligence, 18, pp. 87-127, 1982.

[18] Guarino, N. "The Ontological Level", Wittgenstein Symposium, 4, Kirchberg, Austria, 1993.

[19] Gangemi, A., Steve, G., Giancomelli, F. "ONIONS: An Ontological Methodology for Taxonomic Knowledge Integration", ECAI-96 Workshop on Ontological Engineering, Budapest, August 13, 1996.

[20] Milward, D., Beveridge, M., "Ontology-based dialogue systems". IJCAI-2003, Acapulco, Mexico, August 2003.

[21] Dzikovska, M., Allen, J., Swift, M., "Integrating Linguistic and Domain knowledge for Spoken Dialogue Systems in Multiple Domains". IJCAI-2003, Acapulco, Mexico, August 2003.

[22] Neto, J., Mamede, N., Cassaca, R., Oliveira, L. "The Development of a Multi-purpose Spoken Dialogue System". Eurospeech 2003, Genéve, Switzerland, Sept. 2003.

[23] Rayner, M., Lewin, I., Gorrell, G., Boye, J. "Plug and play speech understanding", 2nd SIGdialWorkshop on Discourse and Dialogue, Aalborg, Denmark, Sept. 2001.

[24] Seneff, S. "TINA: A Natural Language System for Spoken Language Applications", Computational Linguistics, 18(1):61–86, March 1992.

[25] Masuoka, R., Parsia, B., Labrou, Y. "Task Computing - The Semantic Web Meets Pervasive Computing", International Semantic Web Conference, 866-881, 2003.

[26] Ref: FIPA Consortium. FIPA Device Ontology Specification, http://www.fipa.org/specs/fipa00091/. 2004.