MODELING ENVIRONMENTS FOR USE IN A LOCOMOTION ASSISTANCE DEVICE FOR THE BLIND

Christophe Jacquet
Supelec – Plateau de Moulon – 91192 Gif-sur-Yvette CEDEX – France
LIMSI-CNRS – 91403 Orsay CEDEX – France
Christophe.Jacquet@supelec.fr

Abstract The study presented in this paper aims at designing a locomotion assistance device that can deliver semantic information about its surrounding environment at any time. The device will be based on the *Teletact*, a laser telemeter that currently helps blind people feel obstacles at distance. To this end, we first introduce an original model suited for the description of building structure. Then, we explain how it is possible to link semantics to structure. Finally, we expose some research directions for user positioning and interface design.

Keywords: Mobility, environment modeling, human interaction, robotics

Introduction

Over the past few years, the LIMSI (Laboratoire d'Informatique pour la Mecanique et les Sciences de l'Ingenieur) and the LAC (Laboratoire Aime Cotton) have been developing the *Teletact* [Farcy and Bellik, 2002] (see fig. 1), a locomotion assistance device for the blind. The system uses a laser telemeter to measure the distances to obstacles and transforms them into tactile vibrations or musical notes (the higher the tone, the closer the obstacle). Thus, users are aware of the presence of obstacles *in advance*, so they can anticipate movements, and have more fluent trajectories.

To improve the system, we want to give it the ability to provide *symbolic information* about pointed objects. This paper discusses our preliminary results and future directions on this topic. The basic needs for our project are:

- a model and an associated formalism to describe and annotate architectural environments,
- algorithms to determine what relevant information to provide users with,
- a means to compute the 3D positions of the device and of the user.

In consequence, my PhD thesis aims first at defining a theoretical framework for the tracking of users within a modeled environment, based upon both symbolic and analog information. The project will be carried out in collaboration by Supelec and the LIMSI. Ultimately, the design of a next-generation locomotion assistance device will be a practical application of this theoretical research work.

Figure 1. Photo of the current *Teletact*.

In this paper, we focus on this application and its requirements, and hence we introduce some more theoretical aspects. After giving a short overview of the system, we present a model for the description of buildings. Next, we introduce semantic annotations, and we show how to link them to building structure descriptions. This represents the work achieved so far. The last part of the paper presents directions for user positioning and human-computer interface design.

1. System Overview

The system will try to determine its position thanks to a GPS (Global Positioning System) receiver where GPS reception is possible, and otherwise thanks to an inertial unit. The position calculated from these devices will then be matched against structural and semantic information embedded in the environment description retrieved through the network, so as to increase precision and compensate for positioning errors (see section 4).

Context-awareness will be enhanced thanks to telemeter data (as in existing prototypes of the *Teletact*) and to light sensors that can provide additional information about light sources (sunlight, artificial light).

When the position of users has been determined, the system will give them context-related semantic information. And when they point at some specific object or location, the system will provide them with information *about* this object or location.

In addition, the system will act as a path planning assistant, enabling visually impaired people to find their way to some target.

2. Environment Modeling

Up to now, we have worked on building descriptions only. Of course, our model will eventually cover the full range of environment descriptions.

Existing Description Formats

Currently, formats like VRML (Virtual Reality Modeling Language, [Web 3D Consortium, 1997]) or X3D (Extensible 3D, [Web 3D Consortium, 2003]) are available to describe 3D scenes. However, they focus at describing the mere visual *appearance* of environments, whereas we need to embrace both their *structure* and *semantics*.

The **structure** of architectural environments determines how architectural elements are organized to compose buildings. Some elements of this structure may not be visible: we can imagine that a room be divided into two zones, a smoking one and a non smoking one. From a semantic point of view, a frontier exists between the zones. Although this is not a *physical* frontier, we need to model it. We call this a *virtual wall*.

Symbolic data associate semantics to the underlying structure. For instance, these data may contain information about the owners of rooms in a building, access restriction schemes, fire instructions, etc.

To put it short, virtual reality models target *sighted people* and try to describe scenes with many accurate visual details in order to be visually as close to the reality as possible. Conversely, we target *blind people* and thus we need to model the *structure* and *semantics* of environments.

For these reasons, most existing current 3D description languages do not suit our particular needs.

Geographical databases [Hadzilacos and Tryfona, 1997] are useful and practical when dealing with geographic data, but they do not allow the explicit representation of structure, so we will not use them for building description.

Thus we define our own model to describe environments, but we aim at being able to perform conversions from existing descriptions (see section 4).

Modeling Building Structure

A Three-Tier Model. We introduce a three-tier approach to describe building architectures:

- **first tier**: we call *lexical elements* the simple (elementary) architectural elements, such as walls, doors, flights of stairs, and so on. They are defined by their geometric coordinates,

- **second tier**: the so-called *syntactic elements* are complex (composed) architectural elements, constituted by putting together several lexical elements. For instance, a room is defined by its walls, a stairway is defined by several flights of stairs and landings, and so on,

- **third tier**: syntactic elements are further aggregated in what we call *aggregation elements*. For instance, several offices can be gathered in a *cluster* called, say, “sales department”.

We can draw a parallel between our terminology and the structure of natural-language texts. At a low level, texts are simply made up with words: this is the lexical level. These words are then put together in sentences with respect to a defined syntax. In turn, sentences are aggregated in various units such as paragraphs, bulleted lists and so forth.

Algorithms exist, that use this model to determine which piece of information is most relevant to the user, as shown in [Jacquet et al., 2004].

Concept hierarchy. The concepts used to build descriptions follow an object model, and thus take place in a class hierarchy. On grounds of genericity, all classes derive from an abstract common ancestor called `Element`. This class has got three abstract subclasses, corresponding to the three tiers of our model: respectively `LexicalElement`, `SyntacticElement` and `AggregationElement`.

Concrete classes are then derived from one of these abstract classes, depending on what tier they belong to.

Building a Description. As stated above, a description is composed of three tiers of objects, bound together as shown on fig. 2.

Elements are linked by two kinds of edges:

- **inclusion links** (solid lines on fig. 2) represent inclusion between elements within a given tier,

- **composition links** (dotted lines with arrows on fig. 2) enable objects of tier n to be composed of elements of tier $n-1$.

For instance, a room is on the one hand *composed* of several walls, and on the other hand *included* in a floor.

Metamodel. So far, we have defined a certain number of concepts. The set of concepts available in our model (i.e. the class hierarchy) may evolve in the future, as we will add, move, rename, or delete some concepts.

However, several characteristics of our model will persist: for instance, the hierarchical organization of concepts or the tier-based structure of descriptions.

This has led us to define a metamodel: the stable characteristics of the model are defined by the metamodel, whereas the concept hierarchy is declared using a *model description language* associated with the metamodel.

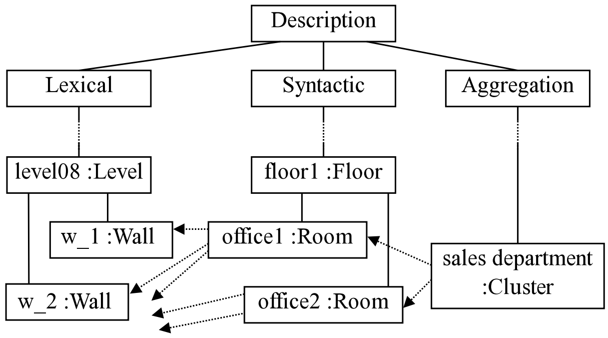

Figure 2. Excerpt of an example description. The *sales department* is composed of two offices, in turn defined by some walls. Intermediate layers of objects have not been represented for the sake of clarity, and have been replaced with dotted portions of vertical lines.

Choosing a Formalism. Our building descriptions correspond to structured data. Thus, it is quite natural to use XML (eXtensible Markup Language) as a formalism, since it is possible to represent structure *explicitly* in XML.

The XML vocabulary used in our descriptions is specified using the XML Schema language [Fallside, 2001].

Schema generation. The schema corresponds to the expression of the model in the XML Schema language. Therefore we need to convert the description of the model from the specific model description language into XML Schema. This can be done quite easily thanks to an XSLT transformation [Clark, 1999] if the model description language is itself an XML dialect.

So we focus at writing the model description (i.e. the class hierarchy) with respect to the metamodel, the XML schema being automatically generated by an XSLT stylesheet written once and for all.

3. Beyond Structure: Semantics

Motivation

What are we able to do now? When the user points at an architectural element, the system is able to find it in its cartography. For instance, if the user points at a door, the system *knows* that it is a door, and that there is, say, an office behind.

However, our ultimate goal is to provide the user with *semantic* information. In the above example, the system would not only state that the user is pointing at a door leading to an office: it would also return the office owners name, the office function, and so on.

To do this, we associate semantic information to the structure description. More generally, such information can be used:

- to add *normative* information to the structure, for example in order to tag restricted areas in a building,
- to identify objects, rooms, and zones,
- to represent connexity information,
- to add specific information to certain kinds of objects, for instance information about painters could be associated with paintings in a museum.

Technical Solutions

Information has been modeled using the Resource Description Framework (RDF) [Manola et al., 2003], an emerging W3C standard quite close to the theory of conceptual graphs [Sowa, 1976].

It allows the representation of semantic information as graphs, where vertices represent objects (called *resources*), and edges represent relations between objects (called *properties*).

In our system, we impose a strong typing of objects used in RDF graphs. It means that every object has got a well defined data type. What is more, we use an object model: all data types take place into a type hierarchy. It follows that every object in a graph is actually an *instance* of an object.

Therefore, how do we formally define the class hierarchy? We have to resort to an *ontology language* that is compatible with RDF. Over the last few years, several such languages have been defined, but the most advanced one is the Web Ontology Language (OWL) [McGuinness and van Harmelen, 2003] that is being fostered by the W3C.

So far, we have defined a set of OWL classes that allow the construction of descriptions involving people and places within a given organization.

Linking Semantics to Structure

Until now, we have defined two worlds: on the one hand, a world of structure descriptions, and on the other hand, a world of semantic annotations.

Actually, semantic annotations are *anchored* in structure descriptions, through architectural elements (see fig. 2). Indeed, architectural elements belong to both worlds. For example, a room takes place in the structure description, as part of the building. But it is also likely to appear in the semantic annotations, for instance in an access restriction scheme.

In conclusion, some architectural elements will appear twice: once in the structure description, and once in the semantic annotations, as instances of a special class called `Place`, thus building a bridge between both worlds.

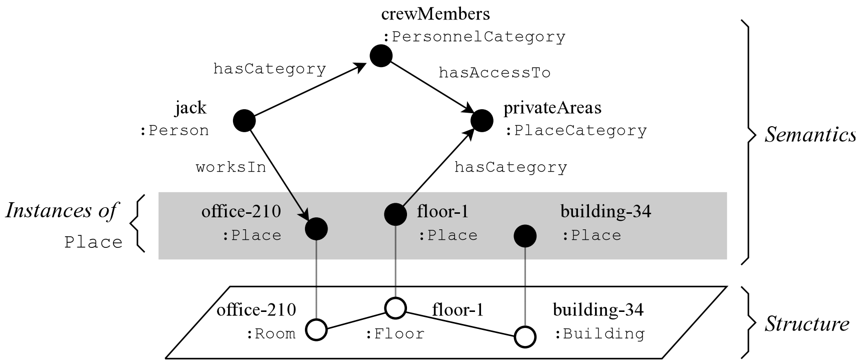

Figure 3. Example of semantic description anchored in the "structural" world. Note that instances of `Place` appear twice, thus acting as bridges between both worlds.

Current Ontology for Structure Description

Currently, we have defined a basic ontology for structure description. The classes of this ontology take place in three categories:

- **Locations:** in addition to the basic class `Place` already described, we define two classes, `Function` that associates a function to a location, and `PlaceCategory` that allows categories of places to be defined (with respect to an access restriction scheme),
- **People:** likewise, a class `Person`, borrowed from the FOAF (Friend Of A Friend) ontology [Dumbill, 2002], represents human beings, `Role` represents roles played by people within an organization, and `PersonnelCategory` allows to specify categories of people (again with respect to an access restriction scheme),
- **Schedules:** the classes `Schedule` and `Event` are used to associate events to locations.

The ontology also defines a whole set of RDF properties, used to specify relations between class instances. For example, `hasCategory` is used to associate a `PersonnelCategory` to a `Person` or a `PlaceCategory` to a `Place`; `hasAccessTo` states that a `Person` or a `PersonnelCategory` have access to a `Place` or `PlaceCategory`.

4. Future Work

Tracking User Position: Semantic Map Matching

The whole system depends on its ability to track the position of the user. In the open, it should be quite easy thanks to the use of a GPS receiver. Indeed, GPS positioning has become increasingly accurate over the past, especially since Selective Availability has been turned off.

However, there are many places where GPS reception is not possible, for example inside buildings, or in dense urban areas, due to *canyon effect* caused by the presence of high building, as described in [Chao et al., 2001]. Unfortunately, these are precisely the places where our system would prove the most useful. Therefore we need a means to compute the position of users, even when GPS signal is unavailable.

The basic idea is to embed an inertial unit in the device, and then determine successive positions by means of dead reckoning : at every moment we try to determine our new position by estimating how far we have moved since the last computed position (this is done thanks to gyroscopes, magnetic compasses and accelerometers embedded in an inertial unit). We call this *relative positioning*. Conversely, when GPS reception is possible, we can perform *absolute positioning* because the GPS receiver can compute absolute positions from satellite signals.

Unfortunately, dead reckoning has the drawback of being very much error-prone [Fusiello and Caprile, 1997]: computed positions are likely to slowly deviate from real positions (cumulative errors). But if we restrict the movements of people, we can overcome this shortcoming by performing *map-matching* [Kitazawa et al., 2000], which consists in computing the most probable position of the user *along well-defined paths* and not in every possible direction.

Such constraints are tolerable for vehicles that follow roads, but not for pedestrians. However, our device has got sensors that give much information about its environment, so by combining this rich information with structural and semantic descriptions, it may be possible to restrict the probability of presence of users in some well-defined areas. For instance, from telemeter data we can know if the user is following a wall (and how far the wall is); from light sensors we can know if we are inside a building our outside, etc. This idea, dubbed *semantic map-matching*, seems to be an interesting research topic and will be further investigated in the future.

Acquisition of Descriptions

In this whole paper, we have assumed that we had environment descriptions at our disposal. Actually, these descriptions need to be constructed. We have listed three ways of obtaining environment descriptions:

- to write them from scratch, either by editing an XML file by hand, or by using a graphical editor,
- to perform a conversion from an existing description, either automatically or semi-automatically. As architects are currently defining their own languages for building description [van Rees et al., 2002], it could

be useful to be able to reuse their building description data at some point in the future. Similarly, we could use existing Geographical Information Systems to obtain geographical information,

- to scan environments with the *Teletact*, and label objects on the fly.

The last method seems the most promising – and the most challenging, too. It would allow blind people to use their locomotion assistance devices even in places where there is no available description. We can imagine that they would tag the environment when first visiting a new place accompanied by some sighted person (as blind people usually do). From these data, the system would compute a partial model (*map learning*, [Fusiello and Caprile, 1997]), that could be re-used next time. Each time the system would return to the same place, it would refine its model based on new information acquired.

It can even be imagined that blind people visiting a new place would be allowed to publish their partial model for others to use it and improve it in turn.

Presenting Semantic Data

In section 3 we have defined a way to associate semantic annotations to environment descriptions. However, our semantic graphs have to comply with a given ontology. So far, we have defined a test ontology that covers the fields of basic organizational relations. It is possible to extend this ontology, so as to cover wider fields. Unfortunately, it will never be possible to design an ontology wide enough to cover every situation.

Thus, description designers will have to define their own ontologies, or ontology fragments. As a result, the system will have to be able to deal with new (i.e. unknown) ontologies. In particular, it will have to know how to present users with information conforming to any given ontology. We could impose that every ontology has to be accompanied by a presentation sheet that would explain how to present instance data, within the framework of a multimodal system: for instance, a particular type of relations could be represented by speech synthesis or using a Braille display.

Conclusion

Over the past years, a useful locomotion assistance device for the blind has been developed by our laboratories, but currently it is merely capable of indicating distances to obstacles. To achieve our goal of being able to name objects and give additional information, we have proposed solutions for two critical issues in this paper:

- modeling the structure of visited buildings,

- representing semantic information associated with the structure. With user position tracking, the system will then be able to determine candidate interesting information.

User position tracking, description acquisition, semantic data presentation and user interface design will be among the topics of our future research work.

References

[Chao et al., 2001] Chao, J., qi Chen, Y., Wu Chen, X. D., Li, Z., Wong, N., and Yu, M. (2001). An Experimental Investigation into the Performance of GPS-based Vehicle Positioning in Very Dense Urban Areas. *Journal of Geospatial Engineering*, 3(1):59–66.

[Clark, 1999] Clark, J. (1999). XSL transformations (XSLT) version 1.0. Technical report, W3C World Wide Web Consortium.

[Dumbill, 2002] Dumbill, E. (2002). Finding friends with XML and RDF. `http://www-106.ibm.com/developerworks/xml/library/x-foaf.html`.

[Fallside, 2001] Fallside, D. C. (2001). XML schema part 0: Primer. Technical report, W3C World Wide Web Consortium.

[Farcy and Bellik, 2002] Farcy, R. and Bellik, Y. (2002). Locomotion assistance for the blind. In Keates, S., Langdom, P., Clarkson, P., and Robinson, P., editors, *Universal Access and Assistive Technology*, pages 277–284. Springer.

[Fusiello and Caprile, 1997] Fusiello, A. and Caprile, B. (1997). Synthesis of Indoor Maps in Presence of Uncertainty. *Robotics and Autonomous Systems*, 22(2):103–114.

[Hadzilacos and Tryfona, 1997] Hadzilacos, T. and Tryfona, N. (1997). An extended entity-relationship model for geographic applications. *SIGMOD Record*, 26(3):24–29.

[Jacquet et al., 2004] Jacquet, C., Bourda, Y., Bellik, Y. (2004). A Context-Aware Locomotion Assistance Device for the Blind. In *HCI 2004 – Design for Life*, Leeds Metropolitan University, UK. Springer. *To appear*.

[Kitazawa et al., 2000] Kitazawa, K., Konishi, Y., and Shibasaki, R. (2000). A Method of Map Matching For Personal Positioning Systems. In *Proceedings of the Asian Conference on Remote Sensing ACRS 2000*.

[Manola et al., 2003] Manola, F., Miller, E., and McBride, B. (2003). RDF primer. Technical report, W3C World Wide Web Consortium.

[McGuinness and van Harmelen, 2003] McGuinness, D. L. and van Harmelen, F. (2003). OWL web ontology language overview. Technical report, W3C World Wide Web Consortium.

[Sowa, 1976] Sowa, J. F. (1976). Conceptual graphs for a database interface. *IBM Journal of Research and Development*, 20(4):336–357.

[van Rees et al., 2002] van Rees, R., Tolman, F., and Beheshti, R. (2002). How bcXML handles construction semantics. In *CIB W98 Workshop*, Denmark, Arhus.

[Web 3D Consortium, 1997] Web 3D Consortium (1997). The Virtual Reality Modeling Language International Standard ISO/IEC 14772-1:1997. Technical report. Available at `http://www.web3d.org/technicalinfo/specifications/vrml97/`.

[Web 3D Consortium, 2003] Web 3D Consortium (2003). Extensible 3D (X3D) International Standard ISO/IEC FCD 19775:200x Final Committee Draft. Technical report. Available at `http://www.web3d.org/fs_specifications.htm`.