Towards a Specification Language for Spatial User Interaction

Khadidja Chaoui, Sabrina Bouzidi-Hassini, Yacine Bellik

Towards a Specification Language for Spatial User Interaction

KHADIDJA CHAOUI
Laboratoire des Méthodes de Conception des Systèmes (LMCS), ESI ex INI, Algiers, Algeria

Sabrina Bouzidi-Hassini
Laboratoire des Méthodes de Conception des Systèmes (LMCS), ESI ex INI, Algiers, Algeria

Yacine Bellik
LIMSI, CNRS, Université Paris-Saclay, Orsay, France

ABSTRACT

Spatial interactions have a great potential in ubiquitous environments. Physical objects, endowed with interconnected sensors, cooperate in a transparent manner to help users in their daily tasks. In our context, we qualify an interaction as spatial if it results from considering spatial attributes (location, orientation, speed…) of the user's body or of a given object used by her/him. According to our literature review, we found that despite their benefits (simplicity, concision, naturalness…), spatial interactions are not as widespread as other interaction models such as graphical or tactile ones. We think that this fact is due to the lack of software tools and frameworks that can make the design and development of spatial interaction easy and fast. In this paper, we propose a spatial interaction modeling language named SUIL (Spatial User Interaction Language) which represents the first step towards the development of such tools.

CCS CONCEPTS

• **Human centered computing**; • **Human Computer Interaction**; • **Ubiquitous and mobile computing.**;

KEYWORDS

Spatial user interactions (SUI), Smart environments, Ambient intelligence, Ubiquitous computing

ACM Reference Format:

KHADIDJA CHAOUI, Sabrina Bouzidi-Hassini, and Yacine Bellik. 2020. Towards a Specification Language for Spatial User Interaction. In *Symposium on Spatial User Interaction (SUI '20), October 31–November 01, 2020, Virtual Event, Canada.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3385959.3424278

1 INTRODUCTION

Spatial interactions represent a great opportunity for users in ambient and smart environments. In such environments, physical space itself can be used as an interaction modality. We consider as spatial interaction any interaction that results from considering spatial attributes (location, orientation, etc.) of a Spatial Interactor (Spin) which refers to user's body or a given object used by her/him, as well as considering spatial relations (distance for example) between Spins. For example, Proximo [1] is location-sensitive mobile application. It helps users to keep track of different objects using Bluetooth technology. When users move away from their objects, the application raises warning alarm this application, the Spin corresponds to the user's body because the system tracks one of its spatial attributes which is location. The reversing radar is another example of spatial interactive applications used in the automotive industry to facilitate parking even when rear visibility is zero. The electronic control calculates the distance between the vehicle and the obstacle. A sound feedback is produced to inform the user about the distance of the obstacle. If the distance value achieves the minimal one, the system raises an alarm. In this application, the vehicle represents the Spin. In fact, the system reacts after calculating distance that separates the Spin from the obstacle. Note that a spatial interactive system may use more than one Spin according to the number of tracked users or objects.

Unfortunately, spatial interaction is not as widespread as other interaction models (graphical, tactile, gestural, speech…). Furthermore, most of existing applications exploit only one spatial attribute among location [2], distance [3] and orientation [4] while spatial interactions could be much richer. In our opinion, the lack of tools that may facilitate the designers/developers work is the main cause of this fact. We think that the first step to be able to build such tools is to define a specification language for spatial interactions on which future software tools can rely on. Such language should be as expressive as possible to allow the coverage of all kind of spatial interactions.

2 SUIL: SPATIAL USER INTERACTION LANGUAGE

To define the SUIL, we first need to identify the main concepts that the language should cover to allow the most possible exhaustive description of spatial interactions. We propose to base the language on the following elements.

2.1 Coordinate system

The coordinate system defines the absolute reference in the given physical space where user interaction takes place

2.2 Entity

Entity represents any object and/or user that may be involved in spatial interactions that need to be modeled in the system. We distinguish two types of entities.

2.2.1 Fixed entity. This kind of entities allows to describe entities that are supposed keeping the same absolute location and absolute orientation in the physical space. It can be, for instance, a piece of furniture (such as a wardrobe), but it can also refer to specific space areas that the designer wants to assign specific functions.

SUI '20, October 31–November 01, 2020, Virtual Event, Canada

ACM ISBN 978-1-4503-7943-4/20/10.
https://doi.org/10.1145/3385959.3424278

2.2.2 Mobile entity. These entities can move by their own in the physical space (such as users and robots) or can be moved by someone else (such as manipulated tangible objects). The term **Spin** (Spatial Interactor) refers to a mobile entity.

Whether it is mobile or fixed, an entity is characterized by its *absolute location* and *absolute orientation* that represent respectively the 3D coordinates and orientation of an object with respect to the considered coordinate system. Besides the absolute location and absolute orientation, mobile entities can be characterized by four additional spatial attributes: *absolute displacement speed, absolute rotation speed, absolute displacement acceleration* and *absolute rotation acceleration.*

2.3 Spatial relation

A spatial relation provides spatial information about an entity A, not with respect to the considered coordinate system, but with respect to another entity B. We define the following relations.

2.3.1 Relative location: determines the location of an entity with respect to another entity (the reference entity).

2.3.2 Relative distance: represents the distance between two entities. Notate that relative distance can be inferred from relative location but not the opposite. Some spatial interactions are based only on distance while others require precise location.

2.3.3 Relative orientation: It consists of indicating how an entity is placed with respect to another reference entity.

Note that using logical operators it is possible to describe spatial relations between more than 2 entities. For instance, if we want to express a condition on 3 entities A, B and C that should be close within a given perimeter, we can specify this condition by using the logical operator AND such as $RD(A,B)<d$ AND $RD(A,C)<d$ AND $RD(B,C)<d$ where *RD* refers to *Relative Distance.*

2.4 Event

Events represent actions (usually issued by users) that trigger system functions. We propose the following events (Table 1), but this list is not yet exhaustive, and more events may be added in future work.

Table 1: List of events used in the language

Event	Condition	Used attribute/ relation
Rotation	The Spin changes its orientation without changing its location.	Absolute orientation.
Move	The Spin changes its location without changing its orientation.	Absolute location.
Enter	The distance between the Spin and a fixed entity becomes lower than a given value.	Relative distance.
Leave	The distance between the Spin and a fixed entity becomes greater than a given value.	Relative distance.
Meet	The distance between 2 Spins becomes lower than a given value.	Relative distance.
Separate	The distance between 2 Spins becomes greater than a given value.	Relative distance.

2.5 System function

System functions represent the system's triggered services. Each event constructed from considering spatial attributes/relations must invoke a system function that meets a user need.

3 FIRST IMPLEMENTATION

We have developed an environment for modeling and automatically generating spatial interactions based on SUIL. We used Obeo Designer for the complete specification of our language SUIL and for the development of the modeling environment and code generation. More precisely we used the EMF framework for the creation of the DSL SUIL (abstract syntax) then we used Siriuis for the creation of the environment for modeling spatial interactions. Finally, for the generation of spatial interaction models we used the Acceleo generator. Figure 1 shows a screenshot of the modeling environment that was used to model a first prototype allowing to control the functions of the VLC video player using spatial interactions.

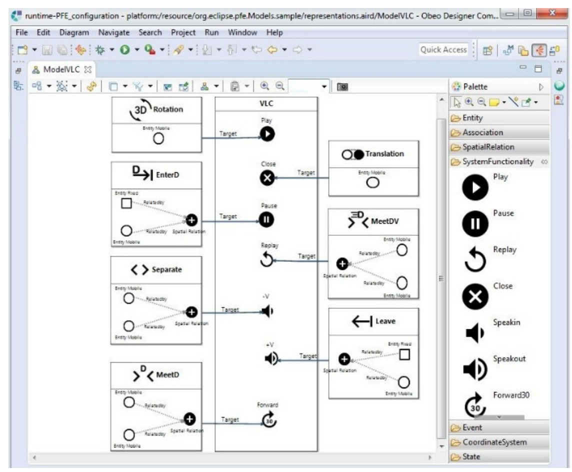

Figure 1: The main window of the modeling environment

4 CONCLUSION

In this work, we have proposed the first foundations of a specification language for spatial user interactions. We have described the key elements of the language that were implemented in a first prototype. In future work, we will continue to enrich the language and check its expressiveness by using it to develop more applications that use spatial interactions.

REFERENCES

[1] Eoghan Parle and Aaron Quigley. 2006. Proximo, location-aware collaborative recommender. School of Computer Science and Informatics, University College Dublin Ireland, 1251-1253.
[2] Ryan D. Bruner. 2012. Exploring spatial interactions. PhD thesis, The University of Texas at Austin.
[3] Thorsten Prante, Carsten Röcker, Norbert Streitz, Richard Stenzel, Carsten Magerkurth, Daniel van Alphen and Daniela Plewe. 2003. Hello.Wall - Beyond Ambient Displays. In Proceedings of Ubicomp'03, pages 277-278.
[4] Frank Chun Yat Li, David Dearman and Khai N. Truong. 2009. Virtual shelves: interactions with orientation aware devices. In Proceedings of the 22nd annual ACM symposium on User interface software and technology (pp. 125-128).