

# Semantic Virtual Environments with Adaptive Multimodal Interfaces

Mario Gutiérrez, Daniel Thalmann, Frédéric Vexo

Virtual Reality Lab (VRlab)  
Swiss Federal Institute of Technology Lausanne (EPFL)  
Lausanne Switzerland CH-1015  
{Mario.Gutierrez, Daniel.Thalmann, Frederic.Vexo}@epfl.ch

## Abstract

*We present a system for real-time configuration of multimodal interfaces to Virtual Environments (VE). The flexibility of our tool is supported by a semantics-based representation of VEs. Semantic descriptors are used to define interaction devices and virtual entities under control. We use portable (XML) descriptors to define the I/O channels of a variety of interaction devices. Semantic description of virtual objects turns them into reactive entities with whom the user can communicate in multiple ways. This article gives details on the semantics-based representation and presents some examples of multimodal interfaces created with our system, including gestures-based and PDA-based interfaces, amongst others.*

**Keywords:** Multimodal Interfaces, Visual programming, Ontology-driven systems, Semantics, Virtual Environments.

## 1. The need for adaptive multimodal interfaces

This article presents research related to the field of interactive virtual environments. We focus on developing multimodal interfaces. Detailed reviews of the state of the art can be found in [3], [15].

Oviatt [14] identifies three main types of multimodal interfaces that have reached a certain level of maturity after several years of research: speech/pen, speech/lip movement and multibiometric input. We have proposed a variation of the speech/pen interface, replacing the pen input by a basic posture recognition of a magnetic tracked wand: the "Magic Wand" [5]. Indeed, the interface proved to be robust enough when tested by many users in a public event [1]. Nevertheless, despite the maturity level of some multimodal technologies, the issue of interface adaptation is still

to be solved. In fact, multimodal interfaces (MMI) are usually implemented as ad-hoc systems.

Even if Multimodal interfaces are designed with a focus on flexibility, few of them are capable of adapting to different user preferences, tasks, or contexts [23].

Changing the application or the context in which an MMI is used often requires costly modifications in terms of time and effort. This is usually a matter of doing changes in the application's source code. MMI should be able to adapt dynamically to the needs and abilities of different users, as well as to different contexts of use [16].

MMI adaptation requires being able to manage in real-time the mapping between multiple user inputs and application functionalities.

Different alternatives have been proposed for adaptive man-machine interfaces that can be used in a wide variety of tasks and contexts within virtual environments. Research includes adaptive interfaces for 3D worlds (e.g. [2]) but also adaptation of multimedia content (e.g. [12]). Content adaptation implies accessing the same information in different scenarios/contexts, through different interfaces. Efforts aimed at unifying management, delivery, consumption and adaptation of content led to the creation of multimedia frameworks such as MPEG-21 [4], [18]. Content adaptation requires standard representations of content features, functionalities (manipulation/interfaces information). In MPEG-21 such information is represented -declared- in the form of "Digital Items" which are defined by XML-based descriptors.

XML-based descriptors are frequently used for handling multimedia content (MPEG-7, MPEG-21) but they can be useful for representing multimodal interaction models as well. For instance, in [17] the authors present an adaptive system for applications using multimodal interfaces. They avoid implementing special (ad-hoc) solutions for special problems. All functionalities are treated coherently using a knowledge based approach. For all multimodal inputs and outputs (speech, gestures, pen/keyboard inputs; PDA, TV

screens) they use a common representation approach and generic interaction models. Interaction processing is based on an ontology-driven system. Everything the user and the system can talk about is encoded in the ontology, using an XML-based representation.

Systems as the one cited before solve the problem of accessing -multimedia- content through multimodal interfaces without the need of ad-hoc applications. The coherent content representation allows for implementing a variety of interaction and visualization modalities with minimum effort. However, dynamic input adaptability is not so easily achieved.

Input adaptability can be defined as the ability of an interactive application to exploit alternative input devices effectively and offer users a way of adapting input interaction to suit their needs [7]. Dragicevic [6] has proposed the "Input Configurator Toolkit" which provides a Visual Programming interface for modifying the mapping between input devices and functionalities of an application. The system enables interactive applications to adapt to special interaction devices as well as user preferences and needs. Inputs can be mapped to different application controls, creating customized interaction techniques. For instance, speech input can be connected to a scroll-bar control. One of the advantages is the ease of use and interactivity of the visual representation. The user manipulates block diagrams representing the interaction and application devices and the connections between their respective I/O slots. The system has been used to customize mainly desktop-based applications. Devices and interface configurations are defined through a dedicated script language (ICScript). Using a non-standard language could prevent from porting/extending the system to other programming languages/contexts.

Systems like the ones presented in [17] and [7] show the need and benefits of adaptive multimodal interfaces. In this paper we define the foundations for a real-time adaptive multimodal interface system. We use a visual programming interface as a front-end for dynamic configuration and input adaptation. The system is based on an ontology-driven system using a standard XML representation which allows for extensibility and portability.

The rest of the article is organized as follows. In the next section we describe in detail the foundations of our system: an ontology for interactive Virtual Environments (VE). We define the main entities and their relationships required to build multimodal interfaces to a wide variety of VEs, including 3D worlds and other types of multimedia content. The article continues by describing the Visual Programming interface which serves as front-end for dynamic input configuration. Finally we present some sample multimodal interfaces developed with our system and conclude the paper.

## 2. An Ontology for interactive Virtual Environments

We work mainly with inhabited Virtual Environments (3D worlds). However, the term Virtual Environment (VE) as used in this paper can be applied to any multimedia application. For us, a VE can be considered as a collection of entities, each of them defined by a set of functionalities, with a particular type of associated information and semantics. Entities can be represented in a variety of ways, e.g. as 3D/2D animated shapes, text, images, video, etc.; depending on the context and application. Thus a virtual environment can be a 3D world or a multimedia document containing text, images, audio, etc. Both of them are sets of entities with clearly defined functions and information that can be represented in different ways.

In [10] we defined an object representation based on the semantics and functionality of interactive digital items - virtual objects- within a Virtual Environment (VE). Every object participating in a VE application is a dynamic entity with multiple visual representations and functionalities. This allows for dynamically scaling and adapting the object's geometry and functions to different scenarios.

Based on the semantic representation of the VE, we focus now on defining a way to formalize the dynamic adaptation of the multiple interaction techniques that can be used to communicate within it. The objective was to let the user access the available interaction devices and customize in real-time the way of controlling the VE's functionalities, personalizing the interaction technique.

The research reviewed in the introduction has shown the benefits of using a standard representation -through XML- and a visual programming interface as front-end for an adaptive system. We decided to complement the semantic model presented in [10] with an ontology of objects that allowed for expressing the relationships between interaction devices and virtual entities in a VE.

According to Gruber [8], an ontology is a formal specification of a shared conceptualization. The systems we target are composed of two main parts: interaction devices (multimodal interface) and virtual environment (world under control). The conceptualization shared by both sides consists on the abstraction of two main types of entities: interaction devices and virtual entities (3D animated shapes, images, text, etc.). The formal aspect of the specification refers to the fact that this model shall be both human and machine readable -this is achieved by means of an XML-based representation.

Handling semantic descriptors defined in XML has several advantages. First of them is portability, XML parsers are available for a wide range of HW and SW platforms. Moreover, XML has become the standard format for data representation. Standards for content annotation and description, such as MPEG-7 use XML [19]. Virtually ev-

ery language and specification for semantics annotation and retrieval of digital items (multimedia content, 3D models, etc.) is based on XML.

Ontological principles are well recognized as effective designing rules for information systems [9], [20]. This has led to the notion of "Ontology-driven information systems" which covers both the structural and temporal dimensions [9]. Our adaptive multimodal interface is supported by such a system. The structural dimension concerns a database containing the information describing both interaction devices and virtual entities (semantic descriptors). The temporal dimension is related to the interface (visual programming) that gives access to such information at run-time.

The central point of our formal representation is the conception of VEs as a set of entities with a semantic meaning. Entities that can be represented and affected in a variety of ways, either through user interaction or autonomous processes. Virtual entities have a meaning -semantics- a role to play in the environment. The way they are visualized and controlled -user interaction- depends on: the application context, the interaction devices available and the user preferences and needs. Thus, we must provide a flexible system that allows for adapting the interfaces to the semantics -function- of the content. The functionality of a virtual entity can be accessed in a variety of ways (multiple modalities) the user should be able to choose and configure the interaction technique that best adapts to her needs.

Choosing and configuring an interaction technique translates into mapping the output of an interaction device to a functionality on a particular virtual entity. We designed an ontology expressing this basic principle. Figure 1 shows the diagram of the ontology for interactive Virtual Environments.

On the one hand we have a range of **Interaction Devices** that let the user express its intentions through multiple modalities. It can be by means of a classical mouse-keyboard or through more sophisticated multimodal interfaces such as a PDA, speech, hand gestures or a combination of them. The essential attribute of an interaction device is the data it delivers (output ports). It can be a 2D vector, a token indicating a particular gesture or word being recognized, etc.

On the other hand there are the **Virtual Entities** to be controlled. They can be 3D animated shapes such as virtual characters or multimedia documents, a video, an so on. From the interaction point of view the most important attribute are the user modifiable functionalities. The input ports that let us communicate with them. Virtual entities can be fully manipulable by the user -e.g. the reproduction control of a video, while others could display some behavior as reaction to user input -for example, an autonomous virtual character.

Data coming from interaction devices may require some

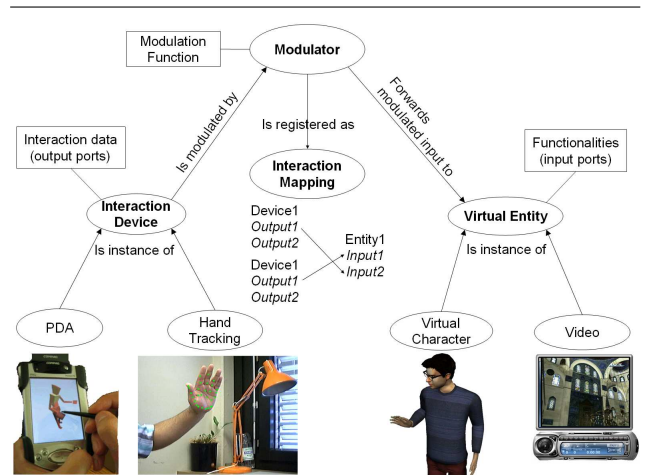


Figure 1. Ontology for interactive VEs: elements involved in a multimodal interface.

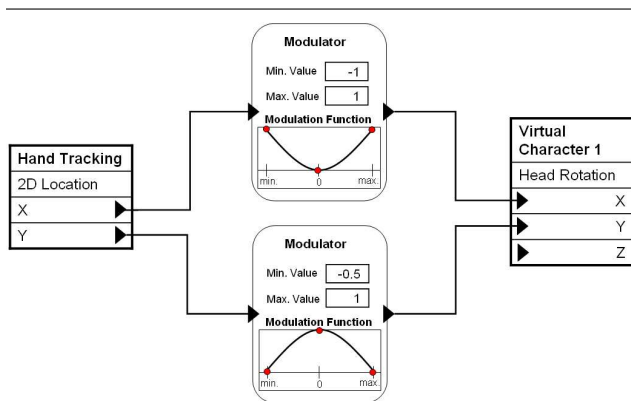
additional post-processing before reaching the controlled entity. We incorporate a mechanism to further process interaction data in the form of **Modulators**. They are containers for modulation functions. We consider interaction devices as black boxes whose output has been already normalized according to some criteria. Nevertheless, modulators are included in the ontology to maximize its flexibility. For instance, in the case some adjustments are needed at execution time, when there is no immediate access to the inner processing mechanisms of interaction devices.

Modulators are also used as the register unit for storing the mapping between an interaction device output and the input of a virtual entity. A multimodal interface is constituted by a set of **Interaction Mappings**. They can be stored and reused.

Now that we have explained the principles of the formal representation, we describe in the next section the meta-interface, the visual programming front-end for creating interfaces.

### 3. Building multimodal interfaces through Visual Programming

The elements of the ontology presented in the previous section translate into XML descriptors such as the ones used in standards like MPEG-7. Descriptors can be created manually, since they are XML files (readable by both humans and machines). However in our system we implemented a visual programming interface (VPI). A meta-interface that eases the task of handling the interface building blocks: interaction devices, modulators and virtual entities; and the links (mappings) between them.



**Figure 2. Visual Programming meta-interface: modulating interaction data and mapping to virtual entity's functionalities.**

When developing this meta-interface we drew inspiration from well-known programming interfaces like the ones implemented in commercial software such as Virtools [21] and LabView [13]. The visual programming paradigm has several advantages when it comes to specify relationships between entities in the form of links between output and input ports [11], [22].

The interface was developed using MS-VisualBasic, this allows for fast implementation of a visual programming interface, from the graphics point of view. Interaction devices, modulators and virtual entities are represented as boxes containing the corresponding attributes. Mapping between interaction data and virtual entities' functionalities is done by connecting I/O ports through modulators, see Figure 2. Interaction data can be of two types: tokens or numeric -normalized- values. Tokens are generally the output of speech recognition algorithms or high-level gesture analysis tools. In the case of numeric values, modulators can treat the input data by means of some user-defined function. In the current version, user can specify the output interval (min., max. values) and modulate the output with a polynomial function. Figure 2 shows two modulation functions with three control points. Up to four control points can be used to define a modulation function.

In the example, the orientation of a virtual character's head is controlled by the user's hand (optical tracking). The hand tracker outputs the normalized position of the head: a 2D vector (0,0) means the hand is on the left upper corner of the camera's view window, while (1,1) indicates the hand is on the down corner to the right. Modulation results into faster movements movements as the user's hand approaches the right or left borders of the view window. Moving the hand up and down directs the characters gaze in the

same direction but the motion speed is faster when the hand is on the center of the view window. In this configuration, there is no way to control the character's head rotation on the Z-axis. The whole configuration is stored as an interaction mapping register which can be retrieved and further modified.

The main elements of the adaptive multimodal interface system are illustrated in Figure 3. Interaction mapping is done in a central component acting as repository and interaction handler. It exchanges data between interaction devices and the virtual environment application. Interaction devices are usually constituted by the device used to capture user input (microphone, PDA, camera,...) and an interaction controller system that process the raw input and normalizes -recognizes- it. Interaction controllers are responsible of communicating with the central interaction handler. This is done by sending the corresponding device -semantic- descriptor through a network connection. Once the central handler is aware of an interaction device, it can display the graphical representation of the semantic descriptor in the VPI. An analogous process occurs in the case of the VE application. Once the user loads a previously defined interaction mapping descriptor or creates a new one, the interaction handler starts processing the interaction data. The central interaction handler modulates data and forwards it to the corresponding input port on the Virtual Environment application.

All communications are done through TCP sockets, allowing the implementation of a distributed system. The interface repository and handler is a Windows application programmed in C++ using the QT development framework. XML processing is done with the Xerces (<http://xml.apache.org/xerces-c/index.html>) and Pathan (<http://software.decisionsoft.com/pathanIntro.html>) libraries. They allow for parsing and evaluating XPath expressions for XML node selection. This way we implemented basic database functionalities (queries, updates, ...) for the semantic descriptors repository.

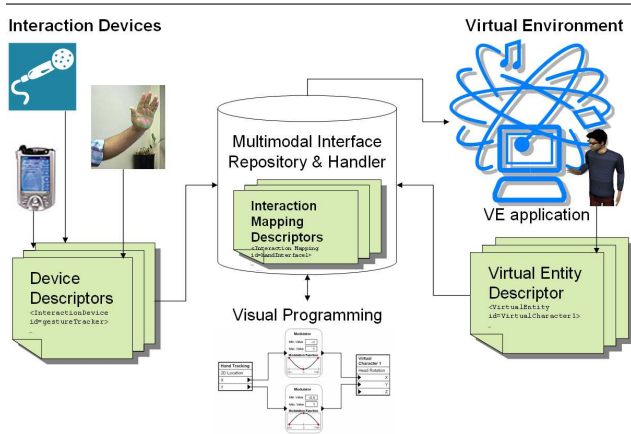
In the next section we describe some examples of adaptive multimodal interfaces implemented with our system.

## 4. Adaptive multimodal interfaces in action

This section shows the feasibility of our system and the benefits of using a semantics-based representation of Virtual Environments. The examples are based on 3D virtual worlds, but the principles are applicable to any multimedia environment.

### 4.1. Gestures based interface

We use optical tracking of facial gestures to animate a virtual character. The application is based on the MPEG-4



**Figure 3. Architecture of the multimodal interface system: data exchange and storage is done through semantic descriptors.**

body animation engine developed in the framework of the IST-INTERFACE project. The stand alone demo of body emotional gestures is transformed into an interactive application using a gesture-based interface. The virtual character displays the user emotions recognized by the features tracker.

#### 4.2. Pen-based interface: the mobile animator

Based on the "Mobile animator", a PDA-based interface presented in. We generalize the use of a handheld as a direct manipulation tool for interaction within 3D virtual worlds.

#### 4.3. Speech-Gestures interface

The "Magic Wand" is revisited and re-implemented according to our new semantics-based representation of Virtual Environments and interaction devices. Our systems gains in flexibility and adaptability.

### 5. Conclusions

#### References

[1] T. Abaci, R. de Bondeli, J. Ciger, M. Clavien, F. Erol, M. Gutierrez, S. Noverraz, O. Renault, F. Vexo, and D. Thalmann. The enigma of the sphinx. In *Proceedings of the 2003 International Conference on Cyberworlds*, pages 106–113. IEEE Computer Society, 2003.

[2] S. B. Banks, M. R. Stytz, and E. Santos. Towards an adaptive man-machine interface for virtual environments. In *Proceedings of Intelligent Information Systems, IIS '97*, pages 90–94, 1997.

[3] C. Benoit, J.-C. Martin, C. Pelachaud, L. Schomaker, and B. Suhm. *Handbook of Multimodal and Spoken Dialogue Systems: Resources, Terminology and Product Evaluation*, chapter AudioVisual and Multimodal Speech-Based Systems, pages 102–203. Kluwer, 2000.

[4] I. Burnett, R. Van De Walle, K. Hill, J. Bormans, and F. Pereira. Mpeg-21: goals and achievements. *IEEE Multimedia*, 10(4):60–70, 2003.

[5] J. Ciger, M. Gutierrez, F. Vexo, and D. Thalmann. The magic wand. In *Proceedings of Spring Conference on Computer Graphics 2003*, pages 132–138, Budmerice, Slovak Republic, 2003.

[6] P. Dragicovic. *Un modèle d'interaction en entrée pour des systèmes interactifs multi-dispositifs hautement configurables*. PhD thesis, Université de Nantes, March 2004.

[7] P. Dragicovic and J.-D. Fekete. The input configurator toolkit: towards high input adaptability in interactive applications. In *Proceedings of the working conference on Advanced visual interfaces*, pages 244–247. ACM Press, 2004.

[8] T. Gruber. The role of a common ontology in achieving sharable, reusable knowledge bases. In *Proceedings of the Second International Conference on Principles of Knowledge Representation and Reasoning*, pages 601–602, 1991.

[9] N. Guarino. Formal ontology and information systems. In *Proceedings of FOIS 98, (Trento, Italy, June, 1998)*. IOS Press, pages 3–15, 1998.

[10] M. Gutiérrez, F. Vexo, and D. Thalmann. Semantics-based representation of virtual environments. In *International Journal of Computer Applications in Technology (IJCAT) Special issue - "Models and methods for representing and processing shape semantics" (to appear)*, 2004.

[11] T. Ichikawa and M. Hirakawa. Visual programming, toward realization of user friendly programming environments. In *Proceedings of the 1987 Fall Joint Computer Conference on Exploring technology: today and tomorrow*, pages 129–137. IEEE Computer Society Press, 1987.

[12] M. Metso, A. Koivisto, and J. Sauvola. Multimedia adaptation for dynamic environments. In *IEEE Second Workshop on Multimedia Signal Processing*, pages 203–208, 1998.

[13] National Instruments Corporation. LabVIEW: Graphical development environment for signal acquisition, measurement analysis, and data presentation. <http://www.ni.com/labview/>.

[14] S. Oviatt. Advances in robust multimodal interface design. *IEEE Computer Graphics and Applications*, 23(5):62–68, 2003.

[15] S. Oviatt, P. Cohen, L. Wu, J. Vergo, L. Duncan, B. S. J. Bers, T. Holzman, T. Winograd, J. Landay, J. Larson, and D. Ferro. Designing the user interface for multimodal speech and gesture applications: State-of-the-art systems and research directions. *Human Computer Interaction*, 15(4):263–322, 2000.

[16] L. M. Reeves, J. Lai, J. A. Larson, S. Oviatt, T. S. Balaji, S. Buisine, P. Collings, P. Cohen, B. Kraal, J.-C. Martin, M. McTear, T. Raman, K. M. Stanney, H. Su, and Q. Y. Wang. Guidelines for multimodal user interface design. *Commun. ACM*, 47(1):57–59, 2004.

- [17] N. Reithinger, J. Alexandersson, T. Becker, A. Blocher, R. Engel, M. Löckelt, J. Müller, N. Pfleger, P. Poller, M. Streit, and V. Tschernomas. Smartkom: adaptive and flexible multimodal access to multiple applications. In *Proceedings of the 5th international conference on Multimodal interfaces*, pages 101–108. ACM Press, 2003.
- [18] L. Rong and I. Burnett. Dynamic multimedia adaptation and updating of media streams with mpeg-21. In *IEEE First Consumer Communications and Networking Conference (CCNC 2004)*, pages 436–441, 2004.
- [19] P. Salembier. Overview of the mpeg-7 standard and of future challenges for visual information analysis. *EURASIP Journal on Applied Signal Processing*, 4:1–11, 2002.
- [20] M. Uschold and M. Gruninger. Ontologies: Principles, methods and applications. *Knowledge Engineering Review*, 11(2):93155, February 1996.
- [21] Virtools SA. Interactive 3d content development tools: <http://www.virttools.com>.
- [22] D. Vodislav. A visual programming model for user interface animation. In *Proceedings of IEEE Symposium on Visual Languages, 23-26 Sept.*, pages 344 – 351, 1997.
- [23] B. Xiao, R. Lunsford, R. Coulston, M. Wesson, and S. Oviatt. Modeling multimodal integration patterns and performance in seniors: toward adaptive processing of individual differences. In *Proceedings of the 5th international conference on Multimodal interfaces*, pages 265–272. ACM Press, 2003.