A Uniform Specification of Mixed Reality Interface Components

Pablo Figueroa∗
Universidad de los Andes.
Colombia

Raimund Dachselt†
Dresden University of Technology.
Germany

Irma Lindt‡
Fraunhofer FIT, St. Augustin.
Germany

ABSTRACT

This is a short presentation of a uniform approach for specifying mixed reality user interfaces, including 3D interaction techniques and 3D widgets. Our main goal is to facilitate the process of reusing previous work, so more complex applications can be built and documented in a formal and uniform way. We describe here the conceptual model of user interface components, which allows us to generalize user interface components and to port them to different hardware settings and application contexts. An XML-based specification language implements the conceptual model and allows for automatic processing and tool support.

CR Categories: I.3.6 [Computer Graphics]: Methodology and Techniques—Languages D.2.1 [Software Engineering]: Requirements/Specifications—Languages;

Keywords: 3D User Interfaces, 3D Interaction Techniques, Mixed Reality, VR, AR, Desktop VR, 3D Widgets, Interface Description Language

1 INTRODUCTION

There are still big difficulties in the creation of complex, portable mixed reality (MR) applications. The lack of a standard description language makes difficult the processes of analyzing, comparing, and reusing interaction techniques. It is also difficult to abstract ideas from previous experiences and use them in new applications, which may be running in different hardware setups. Finally, the lack of a common description language also precludes the creation of supporting tools, guidelines, and methodologies, which are important in the professional work of 3D application designers.

This work attempts to provide a unifying method for the description of 3D user interfaces (3DUIs). We pursue with this work to ease accessibility of 3DUIs, lay the foundations for further and more complex developments, and push the development of 3DUI guidelines, tools, and standards.

2 RELATED WORK

In the field of VR numerous 3D interaction techniques have been developed by various researchers during the past decade. A recent overview is given in the book on 3D user interfaces by Bowman et al. [4], where techniques are classified in terms of task decomposition using the following categories: selection, manipulation, travel, wayfinding, system control, and symbolic input. Figueroa et al. presented InTml [7], an XML-based specification of interaction techniques that was used to create a set of reusable components and influenced this work.

3D interaction in the field of Desktop VR is usually done with the help of 3D widgets, since they allow the subdivision of higher-dimensional interaction tasks into sub-tasks being suitable for lower-dimensional input devices. There have been several taxonomies [10, 5]. In particular, the last one can be found online [13] and contains more than 70 basic widget types from the literature.

Various interaction techniques were also developed in the field of Augmented Reality (AR) in the past few years. The work of Azuma et al. presents a number of advanced solutions [3]. To our knowledge there are no detailed taxonomies nor generalized specification languages for AR, partly because of the novelty of this research area. However, some of the objectives pursued by MR authoring systems such as DART [11] and APRIL [9] are similar to those presented here.

It can also be noticed that XML-based user interface description languages [12] are gaining attention in the field of GUIs, especially for mobile devices. The advantages of using XML-based languages for flexible and parameterized interface descriptions have become obvious with languages such as the User Interface Markup Language (UIML) [2]. For the specification of widgets an XML-based language was also proposed within the Contigra research project [6, 13]. Although it is mainly suitable for Desktop VR, tailored to Web3D applications and not sufficiently general, it influenced the language proposed here.

3 A CONCEPTUAL MODEL FOR 3D INTERFACE COMPONENTS

This section presents our model for the formal specification of interface components. 3DUIs are described as a set of typed components. Some operations defined over the type system allow us to generalize and port components to different hardware platforms, so we can reuse definitions at an abstract level.

In the design of our specification language, we studied the space of interface components and the possible uses of such a language. The dimensions of interest that we have identified for such components are: the required interaction devices, the involved modalities (visual, audible, or tangible), the application type in the mixed reality continuum, the relationships with other components, the actual behavior of a component, the user task supported, the required user skills, and the target group. In this first attempt for a conceptual model, we concentrate on devices, modalities, application types, relationships, target groups, and behavior. Further work should be done in order to specify user tasks and required user skills.

3.1 Definitions

A 3D Interface Component (3DIC) is an identifiable element in the user interface of a mixed reality application employing 3D content. The description of any 3DIC includes the following elements, introduced as an example later in this paper 1: a name, the devices involved in its execution, the abstract events that could be extracted from the devices, a state machine, a set of parameters, the context in which it has been run, and a description of its execution. Not all el-

1 More details can be found online at [1].
ments may be required in particular cases, and current descriptions usually cover a subset of the ideal description for any 3DIC.

A particular 3D widget, gadget, or interaction technique could be represented by one or several 3DICs, depending on the level of abstraction of the description. We propose a division of 3DICs, based on how related they are to devices: Specific 3DICs (S-3DICs) describe components as they are presented in their source, and Generic 3DICs (G-3DICs) are abstract and reusable representations. The set of G-3DICs represent the reusable and platform-independent set of 3DICs, which may be the most important contribution of this representation framework. Initially, we are assuming that the transition from S-3DICs to G-3DICs is made by experts in the field, which certify its quality. We envision aid tools that will guide future developers in the process of identifying suitable G-3DICs from their S-3DICs.

3.2 An Example

Listing 1 shows our description of an early work in the field of AR.

```
GraspableObjectMovement [8] :
St{ ATTACHED_OBJECT DETACHED_OBJECT }
Par{ content.GetAsync3D displayRegion3D }
InvEv{ trackedObjectPosFVec3F trackedObjectOrientSFRotation }
OutEv{ virtualObjectGrabbedObject3D }
InvDev{ brickTracker }
Exec{
  initialize() {
    selectedObj = null;
  }
  trackedObjectPos( value, time stamp ) {
    prevGrabbed = selectedObj;
    if( displayRegion.hasInside( value ) ) {
      selectedObj = getSelectedObject( content );
      if( selectedObj != null ) {
        mapPos2Object( selectedObj, value );
        if( prevGrabbed != selectedObj )
          virtualObjectGrabbed = selectedObj;
      }
    } else selectedObj = null;
  }
  trackedObjectOrient( value, time stamp ) {
    if( selectedObj != null )
      mapOrient2Object( selectedObj, value );
  }
}
```

Listing 1: S-3DIC Description for graspable object movement

It is read as follows: GraspableObjectMovement is an S-3DIC defined in [8]. It can be in one of two states, if an object is attached or not. As parameters, it requires the set of virtual objects to be grabbed and a display region. Its input events are the tracker’s position and orientation, and the expected output is the grabbed virtual object. The input device is a tracker called brick, mapped to the input events (not shown here). The execution shows an initialization function, and two functions that are activated once input events are received. trackedObjectPos updates the grabbed object, and trackedObjectOrient updates the grabbed object’s orientation.

3.3 ICDL: A Unified XML-based Specification Language

We have developed the Interface Component Description Language (ICDL), an XML schema available at [1], for the description of generic and specific 3D interface components in terms of separate root elements. Such a language gives us a consistent description of 3DICs, allows us to describe arbitrary components, gives us space for future extensions, and provides the foundation for program manipulation. There are tags in ICDL for all elements in the conceptual model, plus descriptive elements. We reuse XML languages for state charts, bibliographic references, and documentation.

4 CONCLUSIONS AND FUTURE WORK

We have presented a conceptual model that can be used to describe components of 3D user interfaces in a more formal and extensive way than it is currently common practice. Relevant properties and characteristics of 3DICs have been identified and a conceptual model for 3DICs, including parameters, input and output events, devices and device mappings, states, and execution has been proposed. A detailed and extensive repository of generalized 3DICs may increase the accessibility of existing concepts and may support their selection and reuse. More details can be found at a publicly available repository, where people can access and upload specifications of 3DICs. A first version of such a repository can be found online at [1]. These community efforts along with concluding refinement and extension of the proposed model and language might eventually lead to their standardization.

REFERENCES