Flow of action in mixed interaction modalities

Ernst Kruijff *Stefan Conrad* *Arnold Mueller*

Fraunhofer Gesellschaft Sankt Augustin
Institute Media Communication (IMK)
Competence Center Virtual Environments
Schloss Birlinghoven
53754 Sankt Augustin, Germany
Ernst.Kruijff@imk.fraunhofer.de

Abstract

The increasing complexity of immersive virtual environments (VEs) is leading to multivariate interaction. In order to access and display the functionality of the application, developers start mixing interaction modalities by integrating two and three-dimensional input and output devices. Hence, the input structure of an application gets composite, and is difficult to organize. In order to optimize the input structure, and there from forthcoming output structure, this paper focuses at the flow of action within VEs using mixed interaction modalities. We describe the basic concepts behind flow of action in VEs. Successively, we present how these concepts have affected interaction in our software environment AVANGO, by describing aspects of a collaborative virtual engineering application. Overall, we focus on three main areas. The areas, mapping of interaction on devices, allocation of interface elements, and continuous feedback, address major problems of flow of action in VEs.

1 Introduction

The increasing amount of functionality that can be accessed within nowadays VEs poses a distinct need on the employment of flow of action methods. Most VEs are characterized by a wild mixture of interaction techniques. Often, these techniques do not match the tasks, or input devices at hand or impose a severe cognitive load on the user. Furthermore, some of the functionality within a VE are intrinsically of two-dimensional nature, and can be poorly mapped by a three-dimensional interaction technique. In order to match the task characteristics and structure, application builders are starting to mix multiple devices, thereby regularly using several input modalities. Not only do task-specific (specialized) 3D input devices find their way into VEs, developers also make use of 2D I/O devices again.

Flow of action foremost refers to the user's input structure to an application. It takes into account which actions are performed to reach a certain goal, and how these actions can be perceived as a chain of (sub-) actions. The analysis of the flow of action has been discussed in desktop applications. Nevertheless, within immersive applications the focus has rather been on separate actions, and hardly on how these actions can be coupled in a chain of actions.

In this paper, we present a model of flow of action in VEs, that predominantly focuses at the usage of mixed interaction modalities. Coupled to this model, we present general problems that occur

during interaction with a VE. We state how the framework affected our interaction with virtual reality software, AVANGO, by describing a first implementation of concepts.
In this paper we focus at three main areas, we largely overlap with major problems of flow of action in VEs. The areas are mapping of interaction on devices, allocation of interface elements, and continuous feedback.

2 Related Work

Since the rise of desktop applications, quite some research has been made on more general issues of flow of action. This research is foremost based on task analysis and focused at the users and system behavior over time. A good example of research on the syntax of interaction is Buxton's theory on chunking and phrasing (Buxton 1986). More recently, some research has been focused at so called *continuous interaction*. Continuous interaction takes into account the structure of (also) non-conventional control methods like speech interfaces or gestural commands, which form a continuous input flow to an application. Continuous interaction can be seen as the opposite of discrete interaction, in which a user produces more atomic-like actions (Doherty and Massink 1999). The fields of ergonomics and human control also play an important role within the field of flow of action. Control issues, as investigated over a longer period of time in the area of machinery design, show the relevance of dialogue structure controlling (possibly) more complex systems. A general overview of control issues can be found in (Bullinger, Kern et al. 1997).
Within the domain of VEs, flow of action can be regarded as a specific issue for three-dimensional user interfaces (3DUI). Research on 3DUIs has foremost been focused at separable actions like navigation and selection, and not necessarily how actions can be coupled (Bowman, Kruijff et al. 2001). System control in VEs, as being a major focus in this paper, could greatly benefit from a better flow of action in an application.

3 Flow of action in Virtual Environments

In this section, we focus specific factors of flow of action in VEs, especially those that cause problems during interaction with more complex applications. We base our theoretical model on a detailed task analysis performed on a collaborative virtual engineering application, characterized by a high level of functionality. Furthermore, the application exemplifies (also in the interaction framework and case study sections) the usage of mixed interaction modalities by using several 2D and 3D I/O-devices.
The basic issues of flow of action can be traced back to the several stages of information processing (figure 1). In a VE, the interaction with a system is continuous. The user's input device delivers a continuous input stream, and the user's senses receive continuously receive output. Due to the continuity, multiple loops (as shown in figure 1) will be gone though every second. A break in the loop can lead immediately to a disturbance of the flow of action.
The main aim of supporting flow of action in a VE is to allow the user to make use of the (complete) functionality of a system without being burdened by interrupts in the performance of actions or chains of actions. This requires a clear view on how to reach the functionality of a system, how to apply the functionality in a continuous flow, and finally, the communication of the current state of action (interaction mode).

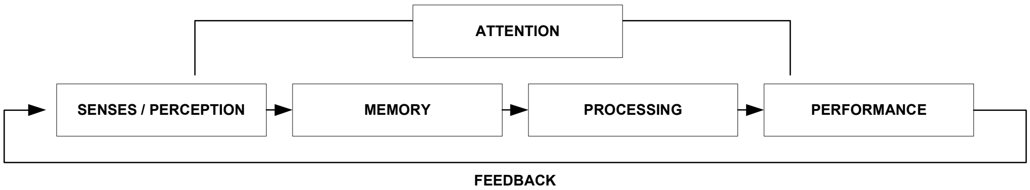

Figure 1: A basic model for flow of action from an information processing perspective

In case of mixed interaction modalities, the input structure to an application gets compound. Even though the mixing of devices can map tasks better, the user's *memory* can easily get overloaded. Many 3DUIs require a minimum level of user experience, since they are regularly not self-explanatory - often functionality is invisible to the user. With the usage of multiple devices, task may be better mapped, but if the user does not know how the functions are mapped to the devices, the task cannot be *performed*. Access of functionality (change of mode of action) is either achieved by pressing a button or by using some kind of system control technique like a flying menu. This change of action is the connector between several tasks or subtasks. Disruption caused by the change of action is a direct disturbance in the flow of action, which can be worsened by failing feedback to the user. Hence, there should be a *continuous feedback* stream to the user, that may be independent of the device. That is, feedback should preferably make use of the same metaphor or technique all the time. This may eventually lead to using one particular sensory channel to supply the user with feedback over time.

Returning to the issue of menus, the inclusion of a desktop-like menu in a VE, or the usage of a PDA or PenPC, mostly leads to the user switching between several focal depths. The menu is two-dimensional and normally in a focal plane close to the user, whereas the actual work area is further away.

Finally, of specific interest for task processing and performance, is the usage of two sensory output channels at once (like gesture and voice (Bolt 1980)), or the application of two-handed interaction techniques, like (Mapes and Moshell 1995). The task is split up in several subtasks being processed by two human output channels simultaneously (parallel processing).

Some general problems can be identified. These problems may seem obvious, but are still regularly occurring pitfalls in many applications.

The *wrong mapping of techniques* on devices is one of the most occurring failures. Developers have to pick from a limited amount of interaction techniques that do not always map the input device in use. Also, input and output devices can be wrongly combined. Overall, wrong mapping leads to performance loss, which directly influences of flow of action in an application. It is a myth that different input devices are capable of transmitting comparable content.

Another regularly occurring problem is *system control overload*. By increasing amounts of functionality, some developers are simply placing more flying menus in the VE. Often these menus have to be large too, for readability issues. The allocation of menus generally overlaps the main work area, leading to attention problems. Rapid eye and head movements of the user are observable, moving from front (menu) to back (work area) focal planes.

Finally, with the increasing complexity of immersive applications, *feedback* to the user is of utmost importance. The user needs to know what is the current mode of interaction, and if a performed action has been succeeded. Often, users need to check multiple times if a task is performed, leading to unnecessary disruptive action loops.

4 Implemented concepts

We have implemented some of the concepts of flow of action in a complex application. As a basis for this application, a collaborative virtual engineering environment, an in-depth task analysis has been performed that specifically looked at task syntax factors affecting flow of action. In the application, a distributed VE, up to 4 remotely connected partners perform design review of an industrial object using direct audiovisual communication. Users have to manipulate the geometry, or design new parts via physical based modeling. During interaction, the users make use of both two-dimensional and three-dimensional data. The system runs at multiple l-shape displays (like the Responsive Workbench^{tm}), using a magnetic tracking system, an adapted stylus, a Cubic Mouse^{tm} (Froehlich and Plate 2000), and a PenPC (Paceblade). The application has been developed in the AVANGO^{tm} software framework (Tramberend 2001).

With respect to the mapping of the functions to the devices, we have analyzed which three-dimensional actions can be performed by the general-task device (the stylus), and which should be performed by the task-specific device (the Cubic Mouse). General task devices often need to make use of some kind of system control technique to change their mode of interaction, whereas task-specific devices regularly integrate the interaction mode into the design of the device. Intrinsically, task-specific devices perform a small amount of actions very well. However, the performance structure of actions performed with the task-specific device regularly include small *repeating interaction loops*. These loops are actions that may not be mapped to the device, like selection of a new object, since performance normally decreases. Nevertheless, since the switching of devices disturbs flow action considerably, we have applied so called *multi-mapping techniques.* Simply said, some actions can be performed by all the devices to avoid device switching. Here the tradeoff between device switching or worse performance is certainly in advantage of the latter, since the repeating interaction loops occur very often.

The two-dimensional tasks, like the session management, are placed on the PenPC. In order to avoid some of the device switching, we added a pen-tip to the stylus, thereby being able to directly control the PenPC with the stylus. The allocation of the graphical user interface elements have particular effects on the attention of the user. The interaction mode changes of the stylus can be achieved by a hand-oriented menu, whereas, as stated before, all two-dimensional actions (including symbolic input) are placed on the PenPC. The PenPC is bidirectional synchronized with the VE, showing the current state of action visualized in the applications scenegraph The PenPC is connected directly in front of the user on the Responsive Workbench. The allocation of the interfaces implies that the user has several focal areas. Foremost, these focal areas are the active work area (the 3D model) and the display of the PenPC. The advantages of using the PenPC are its high readability and low overlap with active work area displayed on the Responsive Workbench. Having two focal areas may have a bad influence on flow of action, since the *focus of attention* may be changing regularly. In our application, we have mapped the functions in such a way, that most of the times, multiple task will be performed serially with one device. In this way, that there is no direct need to switch between a 3D input device and the PenPC in a parallel way. Some exceptions exist, though. When the user is performing symbolic input, the stylus and the PenPC are used in combination. Also, the PenPC can be used for feedback purposes. In order to avoid confusion, having the PenPC in the same place with respect to the user has a big advantage. Due to the fixed position, the focus of attention of the user is always directed to one spot (*directed focus of attention*), thereby minimizing unnecessary search behavior of the user. This behavior is often observed when floating menus are applied, since the user has to check through several focal areas to find the searched widget item. Since the user is switching between devices, continuous feedback needs be taken special care of. This feedback can be attached directly to the input device, for example via a small icon at the stylus tip, or via a feedback method that is independent of the input device in use. Therefor, we have implemented the concept of *cross-device feedback.* First of all, we always display the current interaction mode by a small text-icon, that is placed at the same focal depth as the active work area, in order to avoid switching between focal planes. Nevertheless, for more complex actions, to communicate the active interaction mode is not enough. Therefor, we have applied a scenegraph-oriented interaction mode at the PenPC (Mueller, Conrad et al. 2003). At the PenPC, the currently active node in the scenegraph is displayed, showing detailed data on this node, and the current action performed at the moment. This implies that users can always fall back to the scenegraph-oriented interface, either to receive detailed feedback, or to perform a preferred action directly via discreet input. Even though looking at the PenPC to receive feedback implies a change of focus of attention, it resolves confusion of the user immediately, since the PenPC gives a complete state overview (based on node selection) of the last performed action. The possible attention loss is an affordable tradeoff.

5 Conclusion

In this paper we presented a rough model of flow of action for interaction with a VE using multiple interaction modalities. We identified some major problems of flow of action in VEs, and stated several particular tradeoffs of combining 2D and 3D input devices. We focused at mapping of actions, allocation of interface elements, and feedback issues. We have implemented several concepts in a collaborative virtual engineering environment. Some particular fields of interest are multi-mapping to avoid device switching, directing the focus of attention of the user, and cross-device feedback. In this article several formal tests have been integrated, but more formal evaluation is needed. For example, the analysis of eye and head movements can give new insights in focal area problems.

References

Bolt, R. (1980). "Put-that-there": voice and gesture at the graphics interface. SIGGRAPH'80, ACM.

Bowman, D., E. Kruijff, et al. (2001). "An Introduction to 3D User Interface Design." Presence: Teleoperators and Virtual Environments **10**(1).

Bullinger, H., P. Kern, et al. (1997). Controls. Handbook of Human Factors and Ergonomics. G. Salvendy, John Wiley & Sons.

Buxton, W. (1986). Chunking and phrasing and the design of human-computer dialogues. IFIP World Computer Congres, Dublin.

Doherty, G. and M. Massink (1999). Continuous Interaction and Human Control. European Conference on Human Decision Modeling and Manual Control, Loughborough, Group-D Publications.

Froehlich, B. and J. Plate (2000). The Cubic Mouse, A New Device for Three-Dimensional Input. CHI, ACM.

Mueller, A., S. Conrad, et al. (2003). Multifaceted Interaction with a Virtual Engineering Environment using a Scenegraph--oriented Approach. WSCG, Plzen.

Tramberend, H. (2001). Avango: A Distributed Virtual Reality Framework. Afrigraph'01, ACM.