Using Multimodal Interaction to Navigate in Arbitrary Virtual VRML Worlds

Frank Althoff, Gregor McGlaun, Björn Schuller, Peter Morguet and Manfred Lang

Institute for Human-Machine-Communication
Technical University of Munich
Arcisstr. 16, 80290 Munich, Germany

{althoff,mcglaun,schuller,morguet,lang}@ei.tum.de

ABSTRACT

In this paper we present a multimodal interface for navigating in arbitrary virtual VRML worlds. Conventional haptic devices like keyboard, mouse, joystick and touchscreen can freely be combined with special Virtual-Reality hardware like spacemouse, data glove and position tracker. As a key feature, the system additionally provides intuitive input by command and natural speech utterances as well as dynamic head and hand gestures. The commuication of the interface components is based on the abstract formalism of a context-free grammar, allowing the representation of device-independent information. Taking into account the current system context, user interactions are combined in a semantic unification process and mapped on a model of the viewer's functionality vocabulary. To integrate the continuous multimodal information stream we use a straight-forward rule-based approach and a new technique based on evolutionary algorithms. Our navigation interface has extensively been evaluated in usability studies, obtaining excellent results.

1. INTRODUCTION

The study of user interfaces (UIs) has long drawn significant attention as an independent research field, especially as a fundamental problem of current computer systems is that due to their growing functionality they are often complex to handle. In the course of time the requirements concerning the usability of UIs have considerably increased[15], leading to various generations of UIs from purely hardware and simple batch systems over line-oriented and full screen interfaces to today's widely spread graphical UIs. As principally restricted to haptic interaction by mouse and keyboard, the majority of available application programs requires extensive learning periods and adaptation by the user to a high degree. More advanced interaction styles, such as the use of speech and gesture recognition as well as combinations of various input modalities, can only be seen in dedicated, mostly research motivated applications thus far. However, especially for average computer users those systems would be particularly desirable whose handling can be learned in a short time and that can be worked with quickly, easily, effectively and, above all, intuitively. Therefore current UI research endeavors aim at establishing computers as a common tool for a preferably large group of potential users.

PUI 2001 Orlando, FL USA

1.1 Motivating Multimodal VR Interfaces

Human beings are able to process several interfering perceptions at a high level of abstraction so that they can meet the demands of the prevailing situation. Most of today's technical systems are incapable of emulating this ability yet, although the information processing of a human being works at a plainly lower throughput than it can be reached in modern network architectures. Therefore many researchers propose to apply multimodal system interfaces, as they provide the user with more naturalness, expressive power and flexibility[16]. Moreover, multimodal operating systems work more steadily than unimodal ones do because they integrate redundant information shared between the individual input modalities. Furthermore users have the possibility to freely choose among multiple interaction styles and thereby follow their individual style of man-machine interaction being most effective in contrast to any pregiven interface functionality.

Virtual-Reality (VR) systems currently resemble the top level step in the development of man-machine communication. Being highly immersive and interactive and thus imposing enormous constrains on hard- and software, VR evolved to a cutting-edge technology, integrating information, telecommunication and entertainment issues[3]. Since multimodal data handling provides the technical basis for coordinating the various elements of a VR systems, research in multimodal systems and VR technology is often combined. With regard to the analysis of human factors, a fundamental task is designing VR interfaces consists in solving the problem of orientation, navigation and manipulation in three dimensional (3D) space. In this way our work contributes to multimodal virtual reality research by providing an easy to use interface for navigating in arbitrary virtual VRML worlds. The user can navigate by freely combining keyboard, mouse, joystick and touchscreen with special VR hardware like spacemouse, data glove and position tracker as well as semantically higher-level and more intuitive input modalities like natural speech and dynamic head and hand gestures[7]. An impression of the multimedia working environment can be taken from the photo shown in figure 1.

Figure 1: Multimedia working environment

Figure 2: Screenshot of the navigation interface

1.2 Related Work

Engelmeier[5] et. al introduced a system for the visualization of 3D anatomical data, derived from Magnetic Resonance Imaging (MRI) or Computed Tomography (CT) enabling the physician to navigate through a patient's 3D scans in a virtual environment. They presented an easy to use multimodal human-machine interface combining natural speech input, eye tracking and glove-based hand gesture recognition. Using the implemented interaction modalities they found out that speed and efficiency of the diagnosis could be considerably improved.

In the course of the SGIM project Latoschik[10] et. al developed an interface for interacting with multimedia systems by evaluating both the user's speech and gestural input. The approach is motivated by the idea of overcoming the physical limitations of common computer displays and input devices. Large-screen displays (wall projections, workbenches, caves) enable the user to communicate with the machine in an easier and more intuitive way.

Pavlovic[18] demonstrated the benefits of a multimodal user interface in a military motivated planing domain. He specially focused on combining the modalities speech and gestures in a complementary way achieving a more natural communication form without having the need to deal with special VR hardware. Like in many other systems for integrating multimodal informations contents a frame-based fusion technique[22] is used, coupled with an adapted problem-specific state-machine. The above mentioned three projects represent typical examples for the tendency towards applying multimodal strategies in VR-interfaces.

2. NAVIGATION INTERFACE

Our navigation interface is mainly based on the functionality of common VRML browsers, fulfilling the VRML97 specification standard[9]. Although technically possible, in our system we do not apply a third-person-view, i.e. we explicitely do not include an avatar in the scene. Thus the user of the VR interface only experiences what the virtual avatar would see, he feels directly set into the virtual scenario. The user can freely use the continuous spectrum of both translational and rotational movements. Changing the current location and orientation is equal to moving a virtual camera through the scene. In VRML Objects and interactions are defined with reference to a fixed world coordinate system (wcs). Navigating in this world is equal to transforming an avatar coordinate system within this system. As for example looking to the right is mapped on a rotation of the camera around the vertical axis of the viewing plane.

Handling the display of the current VRML scene context, the browser component represents the major visual frontend of the navigation interface. To cope with different target platforms, we have integrated both a VRML viewer running under the Linux operating system and another viewer running under Microsoft Windows. These two systems and the audio-visual feedback component will shortly be described in the following. A screenshot of the overall navigation interface using FreeWRL is shown in figure 2.

2.1 FreeWRL viewer frontend

Our first viewer component is based on FreeWRL[21]. Besides the three navigation modes WALK, FLY and EXAMINE the browser additionally provides functions for basic interaction with objects in the virtual scene, moving to predefined viewpoints, changing the display quality (light, shading, rendering etc.) as well as basic forms of managing multiuser interaction. In the context of a multimodal navigation interface we slightly changed and extended the original FreeWRL functionality. Among these changes we introduced discrete incremental movements that are split in rotational and translational parts. This was needed as continuous navigation by speech is hard to realize due to a missing direct feedback. Furthermore we introduced a step-size mechanism, regulating the amount of changes as well as a *repeat* and an n-stage *undo* function.

2.2 Blaxxun Contact viewer frontend

The second viewer frontend, Blaxxun Contact[8], is a multimedia communication client supporting both numerous business and entertaining applications on the internet. Compatible to standard WWW-browsers the viewer can simply be used as a plugin, also integrated in other customizable applications. The display module directly supports 3D hardware accelerated graphic chipsets and thus facilitates fluent movements in the virtual scenarios suggesting high level of immersion. In contrast to FreeWRL the Blaxxun viewer fully implements the VRML 97 standard, among others providing gravitation and collision detection. Additional non-standard features include the display of an avatar, effective display of text messages, separatly rendered scenes and special effects like fire, snowfall, and particle systems. The technology for interfacing both browser modules to the individual input devices of our multimodal interface is described later in the text in the context of multimodal data handling.

2.3 Audiovisual Feedback

Arranged directly beneath the viewer module, our interface provides a fully integrated feedback component, presenting feedback and status information in both visual and acoustical form. The feedback window contains text fields as well as self-explaining graphical icons informing the user of the current navigation mode, the history of the last system interactions, and the status of the multimodal integration process. Changing with reference to the current status of the interface, the feedback window is continuously updated. An additional acoustical feedback to each user interaction is given over the integrated loudspeaker system in form of modus-specific audio signals. The various parameters of the system feedback, e.g. the level of information and status messages can online be modified according to individual user needs and preferences.

3. INPUT DEVICES

To navigate in arbitrary virtual VRML worlds our system provides various input devices, which can be classified in haptic modules, special VR hardware modules as well as speech recognition and gesture recognition modules. If technically possible the individual input devices support the full range of possible browser functionalities, i.e. they are not restricted to device specific interaction forms in general. The individual devices are briefly described in the following.

3.1 Haptic Modules

As shown in figure 3, the various haptic modules are classified in discrete and continuous input devices with some devices covering nearly the complete spectrum. Keyboard use can be combined with button interactions in the feedback window and direct manipulation in the viewer window. As keyboard interactions and window buttons are either mapped on discrete movements or directional changes, realistic continuous navigation is currently only possible in mouse and touchscreen mode.

The keyboard navigation mode can be compared to classical computer action game navigation. Hotkeys are used for mode setting (walk, fly, etc.) and directional keys for indicating the direction of movements (left, right, up, down, etc.). Any keypress initiating a movement corresponds to certain incremental changes of the camera location and orientation. Although being highly effective this kind of navigation requires intensive training periods. Therefore, it is most suitable for those group of users, that are very familiar with both keyboard interaction and complex navigation patterns. On the contrary, button interfaces can be used immediately because the graphical icons speak for themselves illustrating their functionality, i.e. indicating directions by arrow symbols. Compared to keyboard interaction, using the buttons is by far not that effective but much easier to comprehend. Therefore, buttons also provide a kind of fallback modality. In our interface keyboard and button interactions currently represent strictly discrete input devices.

The most intuitive and at the same time highly efficient form of navigation is given by the third class of haptic input devices: direct mouse and touchscreen interaction. Areas of the screen can be touched or interacted with the mouse and the scene view is modified with reference to the current navigation mode. Depending on the current context mouse and touchscreen can be used as both a discrete input

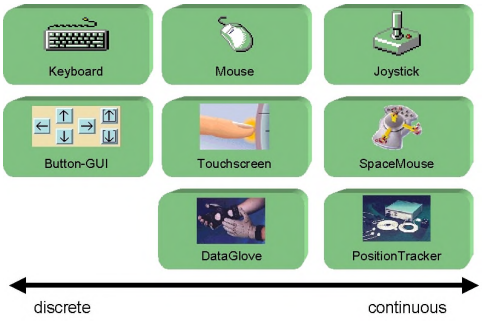

Figure 3: Spectrum of available haptic input devices

device, i.e. by dereferencing objects in the scene for manipulation or target points for a BEAM-TO navigation, and as a continuous device, i.e. by using a drag mechanism. Finally, joystick support has been integrated to alternatively provide an easy to use input device which, in comparison to mouse and keyboard, is explicitely preferred by those users being very familiar with playing computer games.

3.2 Special VR Hardware modules

Also belonging to the class of haptic devices, specially designed VR hardware modules provide additional system interaction which can be used parallel and synergistically to the conventional haptic devices discussed above and the semantically higher level modalities like speech and dynamic hand and head gestures.

Compromising both translational and rotational movements in a single compact device, the spacemouse represents the consequent advancement of the classical mouse. As each degree of freedom for navigating in 3D space can be simulated, it is also referenced as 6DOF mouse (six degrees of freedom). Additional buttons facilitate the control of various system parameters, e.g. navigation mode, light conditions, etc. Same as for keyboard interaction using the spacemouse can be extremely efficient if the user is well trained. Reducing the interaction to each two degrees of freedom for translation and rotation results in easier handling of the device also for non experienced users.

Moreover, our interface provides navigation input by a Pholemus position tracker and a standard data glove. This combination has already proved its usability in numerous applications (among others in [10][5][19]). In contrast to the 6DOF mouse, the tracker allows for intuitive navigation in 3D space with minimal training effort only. Users can simply control the system by moving their hand. The data glove additionally provides a set of static gestures which can be employed to trigger specific interface commands.

3.3 Speech Modules

The task of navigating in arbitrary virtual worlds is cognitively complex. Having the plan to enable any class of users to operate such a system requires the design of an interface being as intuitive as possible. By introducing speech as a new input modality our navigation interface provides a natural form of human-machine communication[20]. Speech moreover allows navigation without losing eye-focus on the scenario by glancing at haptical input devices and leaves one's hands free for further operations. As with reference to special navigation scenarios diverse users show massively

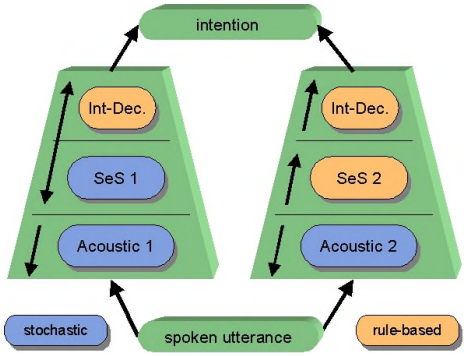

Figure 4: Architecture for recognizing speech

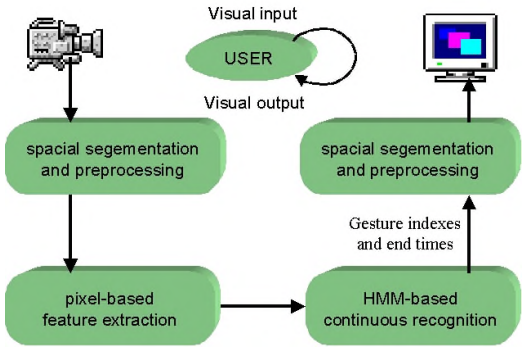

Figure 5: Architecture for recognizing gestures

varying linguistical extend of verbosity. In this respect both command-based speech and natural spontaneous speech utterances are supported.

Two different approaches providing different specializations have been implemented. As shown on the left in figure 4, the first recognition module is a semantic one-pass stochastic top-down decoder, topped by an intention layer. Based on a system developed by Müller and Stahl[14] the acoustic layer is completely integrated in the recognition process extending the search in the semantic-syntactic (SeS) layer by acoustic and phonetic sub-layers. This principle demands for a longer computing time, but delivers robust speech-signal interpretation results given complex utterances without unknown vocabulary. One further restriction is the need of a manual segmentation.

The second recognition module, shown on the right-hand side, is a two-pass decoder topped by the same intention interpreter. On the acoustic stages we use a commercial speech recognition software[6] delivering recognized word chains plus additional single-word confidence measures. The generic idea of not allowing world specific vocabulary strongly forces us to cope with out-of-vocabulary occurrences (OOVs). This is realized only in this second solution by neglecting uncertain semantic units in an intermediate stage on the basis of acoustic confidence measures. On the semantic-syntactic layer we apply an algorithm operating basically rule-based and strictly signal-driven. Ambiguities however are resolved by the use of static probabilities. Regarding these as product of first-order dependencies the a-priori probability for an assumed sub-intention will be set dynamically in a next step, smoothly constraining the rival alternatives. This can be achieved by integrating additional knowledge bases like a user model or by intention prediction seeing the actual system state and dialogue history. The approach in general proves itself most suitable: a recognition rate on a word-chain-basis of 96.3% can be achieved with a corpus including 318 OOVs. When integrating the acoustic layers we achieved a very fast real-time recognition even with automatic segmentation at 67.7% signal-interpretation rate.

Using both decoders together competing against each other in a pre-integrating unit yields very robust recognition results. The effective recognition rate like this exceeds 80% as similar recognition results are mapped on the same commands for controlling the VRML viewer frontend. The principle of single-modal integration interpreted by different instances additionally allows for a confidence measure even on the intention level.

3.4 Gesture Modules

Gestures often support speech in interpersonal communication. In various domains, e.g. when dealing with spatial problems, information can be communicated easier and more precisely using gestures instead of describing certain circumstances by speech. To make a step towards natural human communication behavior, our interface thus provides navigation input by dynamic hand and head gestures in addition to spoken utterances.

The gesture module is mainly based on a system for the recognition of dynamic hand gestures in a visual dialogue system[12]. Figure 5 gives a coarse overview of the underlying recognition process. The central design principle consists in using Hidden-Markov-Models (HMMs) for the temporal segmentation and classification of the pictures in a continuous video stream. Video images are grabbed by a standard CCD camera module and transformed to *yuv* format. For separating the hand from the background a color histogram based spatial segmentation method is used that calculates binary images solely containing the hand shape. Defined lightning conditions combined with additional low-level image processing guarantees optimal segmentation results. Afterwards the spatio-temporal image sequences are transformed into feature vectors, performing best in calculating Hu moments. For the classification we are using semi-continuous HMMs with 256 prototypes and 25 states per model. The temporal segmentation is done with a single-level, highly robust approach, however at enormous computational costs[13]. Basically, the classification principle consists in continuously observing the output scores of the HMMs at every time stamp. Peaks, which appear in the individual HMM output scores, allow to determine which image sequence and which gesture occurred at what time, respectively.

Totally, 41 different hand gestures can be recognized with a rate of over 95%. With an image size of 192x144 pixels the system is able to handle 25 images per second meeting realtime requirements of the multimodal interface. Usability studies clearly demonstrate that visual interaction using hand gestures is highly accepted specially by non-expert computer users and getting even better ratings when having the possibility to combine gestural input with speech.

The same technique used for classifying hand gestures is employed for handling the interpretation of dynamic head gestures, currently differentiating between ten different gestures. Although the module is still in a prototypical status,

already more than three-quarter of the gestures in the training material are classified correctly. Moreover, if clusters of similar gestures are taken into account the recognition rate can further be improved. As a special feature, our head gesture module integrates multimodal context knowledge and therefore provides a goal-oriented classification scheme. Various tests[1] showed that head gestures are selectively used to support the primary mode of communication and thus can be evaluated to obtain better recognition results of the overall user intention.

4. MULTIMODAL DATA HANDLING

In this work we present a generic approach for handling multimodal information in a VRML browser which can be generated by arbitrary and also multiple input devices. Therefore, special VR devices like data gloves, position tracker and spacemouse systems or even higher-level components like speech and gesture recognition modules can synergistically be used parallel to conventional haptic devices like keyboard, mouse, joystick and touchscreen.

4.1 Communication Formalism

For controlling the browser we are using an abstract communication formalism that is based on a context-free grammar (CFG). Completely describing the various functionalities of the browser, the grammar model facilitates the representation of domain- and device- independent multimodal information contents[2]. User interactions and system events are combined in a semantic unification process and thereby mapped on an abstract model of the functionality vocabulary. Thus for example, both natural speech utterances, hand gestures and spacemouse interaction can be described in the same formalism. The browser module just operates on the formal model of the grammar.

A single word of this grammar corresponds to a single command or an event of the interface. Multiple words form a sentence denoting a sequence of actions, e.g. a whole session. The language defined by the grammar represents the multitude of all potential interactions. A part of a typical context-free grammar we are using in our projects is given below. It is described in Backus-Naur form (BNF) and demonstrates a small part of the possible actions in the WALK-mode. According to that grammar a valid command of the functionality vocabulary would be: "walk trans forward". Messages and events created by the various low- and high-level devices from simple keystrokes to complex natural speech utterances are mapped on words of this grammar.

```
<CMD>  ::= <CONTROL> | <WALK> | <FLY> | ...
<WALK> ::= walk <WSEQ> | startwalk <WSEQ> | stop
<WSEQ> ::= <W> | <W> <WSEQ>
<W>    ::= trans <ALLD> | rot <LRD>
<ALLD> ::= <ALL> | <ALL> <ALLD>
<ALL>  ::= <LR> | <FB> | <UD> | <DIAG>
<UD>   ::= up | down
<LR>   ::= left | right
<FB>   ::= forward | backward
<DIAG> ::= lfwd | rfwd | lbwd | rbwd
```

The set of grammar commands can be subdivided in various command clusters. In general three major information blocks can be identified. Movement commands indicate direct navigation information, position oriented commands denote movements to specific locations and finally control commands describe changes of the various browser parameters and the feedback flow.

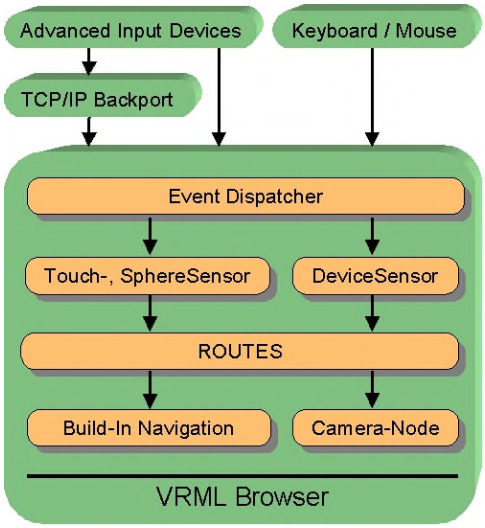

Figure 6: Modified VRML browser

The key feature of our approach using the CFG-module as a meta-device interfacing the various input devices to the browser is that it provides a high level access to the functionality of the individual recognition modules. By changing the underlying context-free grammar the interaction behavior of the target application, the VRML browser, can easily be modified also by non-experts without having to deal with the technical specifications of the input devices. The commands of the grammar are comprehensible straight forward as they inherently make sense. An additional advantage of this approach is that arbitrary new information sources can easily be integrated into the interface, too.

4.2 Controlling the Browser Module

Regarding the event handling of a standard browser implementation we find a rigid event routing between the input events like mouse movements and key strokes and the input handling, that realizes the navigation. In the framework described in this paper we use the mechanism of ROUTEs to break up the rigid event routing. Additional devices can plug in the event dispatcher module of the browser and dispatch their raw input to a DeviceSensor node in the VRML scene graph. In addition, a device can trigger certain browser actions by using the CFG formalism. Therefore, a TCP/IP socket backport has been implemented watching the net for relevant browser commands.

Figure 6 shows the structure of our modified browser module. At VRML scope, the newly defined DeviceSensor node gives access to all kinds of conceivable user input. The Camera node gives superb control over the camera used for rendering. Moreover, by setting velocity vectors, the virtual camera can be animated, computing the distance vector for the current frame in consideration of previous frame times and the results of the collision detection. Besides a 6DOF navigation mode the camera node supports an EXAMINE and a BEAMTO mode, thus covering all conceivable navigation modes while hiding the nasty details of 3D math from the VRML content author. However, in the absence of a Camera, a default built-in navigation should encompass

a minimal navigation feature set allowing all basic navigation actions such as the WALK, SLIDE, EXAMINE modes. Detailed information concerning concept and the implementation of the browser extensions can be found in [2].

4.3 System Architecture

In our research endeavors, we are especially interested in designing a generic multimodal system architecture which can easily be adapted to various conditions and applications and thus serve as a reusable basis for the development of multimodal interaction systems. Belonging to the class of systems working on a late semantic fusion level[17] our design philosophy is to merge several competing modalities (e.g. two or more speech modules, dynamic and static gesture modules as well as avrious haptic modules), not necessarily known in advance.

The individual components of our interface communicate via TCP/IP sockets in a blackboard architecture. For the implementation we propose a classical client-server approach. Physically the integrator functions as a central broadcast server, sending incoming messages to each of the connected clients, i.e. recognition modules, viewer and feedback applications and other modules integrated in the interface. Our architecture explicitely facilitates the integration of various agents in a shared environment, neither bound to any specific software nor operating system. Figure 7 gives a structural overview of our system.

The input modules provide recognition results which are translated by CFG-wrappers with regard to the above described grammar formalism. Afterwards they are sent to the integrator which interpretes the continuous data stream of the individual components and generates commands matching the user's intention. Finally commands are sent to the viewer and feedback component. The components of our interface communicate bilaterally with the integrator, i.e. the viewer sends backs status messages and potential internal error states. To improve recognition results the integrator shares its omnipotent context knowledge with the recognition modules.

All of the messages obey the same format, including unique target id, source id, timestamp and message content. Each module has an own parser component, checking if incoming messages on the socket are correct and designated for the module. Thus the parser also functions as a message filter, as not all messages are of interest for every module.

We have already successfully been porting the underlying architecture to other domains. One of the projects which is done in cooperation with major automobile industry partners deals with the multimodal operation of an MP3, radio and telephone interface in an automobile environment.

4.4 Integration Process

The integrator module interpretes the incoming multimodal information stream. It consists of different coworking components, implemented as independent threads communicating via shared memory. By following an object oriented approach the design of the integrator system is flexible, extensible, easy to maintain, and above all highly robust.

The parser checks incoming and outgoing messages for correctness, deciding if they are valid in the current system context. Containing meta-knowledge of the application, the recognition modules and the integration unit, the state machine provides the database for the integration process. The

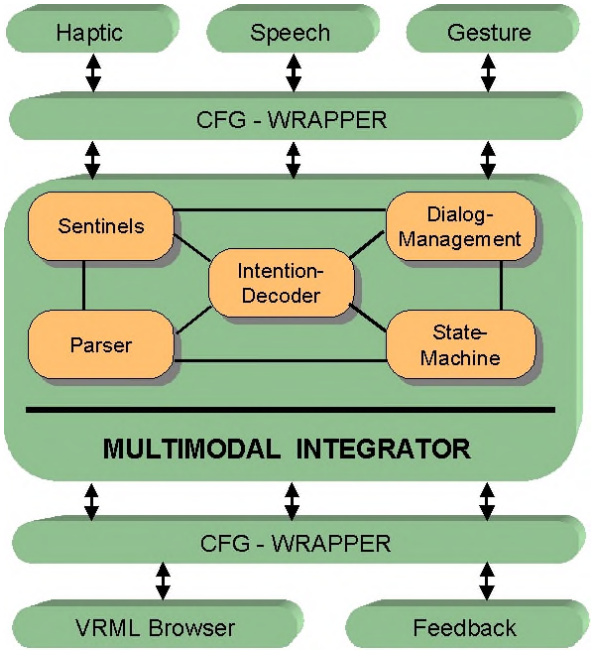

Figure 7: Overview of the system architecture

most important component within the integrator is the intention decoder, generating commands for the viewer. Sentinels help to supervise the data stream, watching for redundant, complementary and concurrent incoming data. Additionally a dialog management module initiates status and error resolving dialogs.

To integrate the various information contents of the individual input modalities we mainly use a classical straightforward rule-based approach, well established in various research applications (e.g. in [4][22]). The semantic unification is carried out with reference to the CFG, modeling device independent user interactions. Single elements, words and subwords of the grammar, are migrated and combined by evaluating a problem-specific look-up table, facilitating the multimodal integration process in realtime. Complementary information contents are most easy to handle as the individual grammatical elements support each other to create a valid browser command. Redundancy is handled by supervising the multimodal information stream and associating various events, that occur within a context sensitive timing window, with the same system command. Moreover, concurrent (competing) information contents are detected when obtaining conflicts in the look-up table indicated by negative scores for the overall user intention.

As a second approach to integrate multimodal user input we introduce a new, stochastically motivated technique. By employing biological inspired evolutionary strategies the core algorithm compromises a population of individual solutions to the current navigation context and a set of operators defined over the population itself. Various integration results are competing against each other. According to evolutionary theories, only the most suited elements in a population are likely to survive and generate offspring, transmitting their biological heredity to new generations and thus lead to stable and robust solutions, i.e. result in the correct interpretation of the user intention.

Figure 8: Distribution of unimodal (left) and multimodal (right) system interactions

Following a technique described in [11] our genetic algorithm is composed of an abstract data type and non-standard recombination operators that are specially designed to suit the problem at hand. A potential solution consists of three chromosome parts. The *administration* chromosome contains information about the pre-command (with exact time and certainty), alternative pre-commands of previous integration steps, mutation possibilities and potential following commands as well as matching partners for complementary interactions. The *command* chromosome contains the generated commands, i.e. a word of the grammar formalism with a specific confidence measure and information about necessary additional commands. Finally a number of *data* chromosomes, one for each input device, compromise detailed information on the recognized semantic units, recognition times, certainties as well as lists of complementary, supplementary and competing information contents. Both crossover and mutation techniques are applied to obtain optimal results. The fitness of an individual, measuring the certainty and the confidence of a generated command, is calculated according to a statistically weigthed scheme including the various information resources.

The key feature of this new technique is that the fundamental aspects of multimodal integration (redundancy, complementarity and concurrency) do not have to be handled as special cases because they are implicitly modeled by the structure of the evolutionary algorithm. As a big advantage the approach facilitates an easy integration of additional input modules without having to modify the integration algorithm itself. Finally the algorithm is extremely robust with regard to changing boundary conditions.

5. USER STUDY

We extensively evaluated the multimodal navigation interface in usability studies[1]. A total of 40 test persons participated in the test, which was separated in two series with 17 persons using a standard mouse as the primary pointing device and 23 persons using touchscreen interaction instead. A small control group of three persons participated in both test series. The individual test subjects were divided into three groups (beginners, normal computer users and experts) according to a weighted classification scheme of various answers given in the first question form. In the course of the test, the participants were asked to, but not forced to use combinations of mouse, touchscreen, keyboard, speech as well as hand and head gestures. The test was separated in five blocks evaluating the performance of four prescribed combinations and free interaction.

The user study has partly been realized according to the 'Wizard-Of-Oz (WOO)' test paradigm. In contrast to haptic interactions that are directly transcribed by the system, the hand, head and speech recognition has been simulated by a human person supervising the test subjects via audio- and video-signals and generating the appropriate browser commands. Although we have already implemented and tested the individual input modules, the WOO scenario has been chosen to cope with different running times and potential failures of the recognition modules. As we were mostly interested in natural interaction forms, we did not want the test results to be dominated by these factors.

The distribution of the various system interactions is related to the navigation tasks, which - in our case - were quite simple. Commands merely consisted of navigation *modes* (walk, fly, examine), *functions* (trans, rot, twist, etc.) and *values* (left, up, forward, etc.). Beginners significantly showed more full commands, containing all informations slots, than the other groups. Concerning the distribution of unimodal commands (shown on the left-hand side of figure 8), obviously all users favored keyboard input. For next best choice experts and normal users clearly preferred the mouse, whereas beginners employed speech and hand gestures.

Although only applied in about one fifth of all interactions, multimodal commands symbolized the core interaction style as they were particularly used to change navigation contexts, i.e. switching from translational to rotational movements. For all gropus the use of speech dominated the distribution of multimodal interactions (shown on the right-hand side of figure 8). Detailed analysis clearly proved that with growing complexity the use of multimodal interaction increased. Regarding the distribution of redundant, competing and complementary interactions, the latter occured most often, in the case of beginners very often coupled with additional redundancy. When combining multiple input modalities, especially beginners favored gestures and speech. The other two groups strongly combined speech (for setting mode and functions) with haptic devices to indicate directions.

A very interesting result was that although people generally stated that they did not like head gestures very much, unconsciously more than 80% used their heads to support their primary communication modality. In general people highly enjoyed having the possibility to freely choose among multiple input devices. The results of the usability study provided the basis for adapting the individual recognition modules and designing appropriate integration algorithms.

6. FUTURE WORK

For the nearest future we further plan to improve our interface. First and foremost we plan to integrate a balanced strategy to switch between discrete and continuous navigation patterns. Handling both input streams in separated units - synchronized by a periodic update cycle - promises major improvements in system usability. Second the current communication formalism is to be extended to better support the individual advantages of the various modalities. Third we want to improve the underlying communication architecture by porting the system to a CORBA based communication message layer, supporting platform independent integration and recognition modules. Moreover, we want to integrate additional input devices as usability studies clearly demonstrated that different people show strongly varying interaction patterns and with more available input possibilities each user has the freedom to apply those devices being most intuitive and effective for him. Finally we plan to integrate error management and adaptive help strategies for supporting also non-experienced users.

7. CONCLUSION

In this paper we proposed a multimodal interface for navigating in arbitrary virtual worlds. Meeting realtime conditions the user can navigate by keyboard, mouse, touchscreen, joystick, as well as special VR hardware like 6DOF mouse, data glove and position tracker and, as a key feature, by natural and command speech and dynamic head and hand gestures. We provided a generic architecture for the development of multimodal interaction systems. Based on the abstract formalism of a context-free grammar the individual system modules communicate with each other via a central integrator, comprising meta-knowledge of the application, the recognition modules and the integration unit itself. To handle the multimodal information stream we used a rule-based approach and additionally introduced a new technique by employing evolutionary strategies. Finally we presented some results of a usability study and identified potential extensions to our multimodal interface.

8. REFERENCES

[1] F. Althoff, G. McGlaun, and M. Lang. Combining multiple input modalities for VR navigation - A user study. In *9.th Int. Conf. on HCI*, August 2001.

[2] F. Althoff, T. Volk, G. McGlaun, and M. Lang. A generic user interface framework for VR applications. In *9.th Int. Conf. on HCI*, New Orleans, August 2001.

[3] H. J. Bullinger. Virtual reality as a focal point between new media and telecomunication. *VR World 1995 - Conference Documentation*, IDG 1995.

[4] A. Cheyer and L. Julia. Designing, developing and evaluating multimodal applications. In *WS on Pen/Voice Interfaces (CHI 99)*, Pittsburgh 1999.

[5] K.-H. Engelmeier et al. Virtual reality and multimedia human-computer interaction in medicine. *IEEE WS on Multimedia Signal Processing*, pages 88–97, Los Angeles, December 1998.

[6] Lernout & Houspie Speech Products N.V. Lernout & Houspie - Software Developers Kit, 1998.

[7] Details of the MIVIS system (October 2001). Internet-Publication, http://www.mivis.de.

[8] Developer site of blaxxun interactive (July 2001). http://www.blaxxun.com/developer/contact/3d.

[9] Specification of VRML 97. ISO/IEC 14772-1:1997, http://www.web3d.org/technicalinfo/specifications/vrml97/index.html, July 2001.

[10] M. Latoschik et al. Multimodale interaktion mit einem system zur virtuellen konstruktion. *Informatik '99, 29. Jahrestagung der Gesellschaft fr Informatik, Paderborn*, pages 88–97, October 1999.

[11] Z. Michalewicz. *Genetic Algoithms and Data Structures*. Springer-Verlag, New York, 1999.

[12] P. Morguet. *Stochastische Modellierung von Bildsequenzen zur Segmentierung und Erkennung dynamischer Gesten*. PhD thesis, Technical University of Munich, Germany, Januar, 2001.

[13] P. Morguet et al. Comparison of approaches to continuous hand gesture recognition for a visual dialog system. *Proc. of ICASSP 99*, pages 3549–3552, 1999.

[14] J. Müller and H. Stahl. Speech understanding and speech translation in various domains by maximum a-posteriori semantic decoding. In *Proc. EIS 98*, pages 256–267, La Laguna, Spain 1998.

[15] J. Nielsen. *Usability Engineering*. Morgan Kaufmann Publishers Inc., 1993.

[16] S. L. Oviatt. Ten myth of multimodal interaction. *Communications of the ACM*, 42(11):74–81, 1999.

[17] S. L. Oviatt. Multimodal interface research: A science without borders. *Proc. of 6th Int. Conference on Spoken Language Processing (ICSLP 2000)*, 2000.

[18] V. Pavlovic, G. Berry, and T. Huang. BattleView: A multimodal hci research application. In *Workshop on Perceptual User Interfaces (PUI 98)*, November 1998.

[19] I. Poupyrev, N. Tomokazu, and S. Weghorst. Virtual Notepad: Handwriting in immersive VR. *Proc. of IEEE Virtual Reality Annual International Symposium'98 (VRAIS'98)*, pages 126–132, 1998.

[20] B. Schuller, F. Althoff, G. McGlaun, and M. Lang. Navigating in virtual worlds via natural speech. In *9.th Int. Conf. on HCI*, New Orleans, August 2001.

[21] J. Stewart. FreeWRL homepage. Internet-Publication, http://www-ext.crc.ca/FreeWRL, June 2001.

[22] A. Waibel, M. T. Vo, P. Duchnowski, and S. Manke. Multimodal interfaces. *Artificial Intelligence Review*, 10(3-4):299–319, August 1995.