

The Virtual Bodies Project

Sam Hill
Ision Internet PLC / Reading University
161-165 Greenwich High Road
London
SE10 8JA
UK

Dr. Rachel McCrindle
Department of Computer Science
The University of Reading
Whiteknights, Reading
RG6 6AY
UK

Publication Date: 15th January 2001

Abstract

The Virtual Bodies Project (VBP) is concerned with producing Next Generation Intelligent Agents (NGIA) to help both abled-bodied and disabled people interact with their desktop and when registered with a central server to interact with each other across the Internet. The form and behaviour of each VBP agent can be customised through an interactive body-bank of parts thereby allowing the user to create their ideal 'virtual' companion. This VBP software has full connectivity features enabling the user to access the Internet, receive information and messages from other VBP agents and participate in other interactive services over the Internet. This paper describes the rationale behind the Virtual Bodies Project, outlines the development of the VBP software and describes how the NGIAs improve communication between a user and their desktop and between groups of users across the Internet.

Keywords

User interface and interactions; intelligent agents; web accessibility; collaborative systems

Word Count

7840 words



The Virtual Bodies Project

Sam Hill
Ision Internet PLC / Reading University
161-165 Greenwich High Road
London
SE10 8JA
UK

Dr. Rachel McCrindle
Department of Computer Science
The University of Reading
Whiteknights, Reading
RG6 6AY
UK

Abstract

The Virtual Bodies Project (VBP) is concerned with producing Next Generation Intelligent Agents (NGIA) to help both abled-bodied and disabled people interact with their desktop and when registered with a central server to interact with each other across the Internet. The form and behaviour of each VBP agent can be customised through an interactive body-bank of parts thereby allowing the user to create their ideal 'virtual' companion. This VBP software has full connectivity features enabling the user to access the Internet, receive information and messages from other VBP agents and participate in other interactive services over the Internet. This paper describes the rationale behind the Virtual Bodies Project, outlines the development of the VBP software and describes how the NGIAs improve communication between a user and their desktop and between groups of users across the Internet.

Keywords

User interface and interactions; intelligent agents; web accessibility; collaborative systems

1. Introduction and Aims

Digital technologies such as hypermedia, virtual reality, digital video broadcasting, video conferencing, co-operative working and the world wide web have been the subject of intense development over recent years and together may be said to comprise the second information technology revolution. These developments coupled with the continually decreasing costs of the enabling technology have resulted in significant expansion in their use. Indeed, it is already noticeable that such applications are impacting heavily upon our environment whether in the workplace, in education or in entertainment [1]. Driven by this increase in the use of computers there is a continual need to improve the way in which people interact with computers. The aim of the VBP has been to develop a novel more user-friendly and intuitive way for people to do this with benefits accruing to many different user-client groups.

Inexperienced users in particular often have great difficulty in understanding or using computers. They frequently find themselves facing situations that extend beyond their knowledge of the program they are

using, or in which they are unable to achieve a task with the software and require help. Surveys [2] have shown that the people who suffer from this lack of understanding of computers are predominantly children and older adults, and that this can lead to technostress, which is directly related to technophobia, basically a fear or hatred of computers. There have been many documented cases of Technostress, which highlight the needs of many computer users. For example, a survey for Channel 5 news, discovered that many users "feel that their computer makes them look stupid" [3], since they are often in a situation where they are unsure of what to do. Technostress can result if this problem gets out of hand and may manifest itself as violence towards the user's computer. The root cause of this is a lack of understanding on the part of the software as to what the user of the computer is attempting to do. If a computer program can be developed to address this issue then "finally, someone has addressed the human side of the information revolution." [4].

Experienced users will also benefit from more intuitive interfaces by being able to achieve tasks in a simpler, more efficient and user-oriented way, rather than in that dictated to them by the computer system [5]. There is also a small but very significant group of the general and computing population who currently through some form of disability are unable to exploit fully, the capabilities of current interfaces, and hence take full advantage of the benefits offered to them by the new computing technologies. It should be possible however, to design adaptations or enhancements to user interfaces in order to compensate for sensory, motor or mental impairment of disabled users, thereby creating powerful communication and co-ordination mechanisms [6].

The Virtual Body Project explores ways in which to achieve more natural interaction between the user and the computer and develops a novel application, which is simple to use and which can be easily expanded to encompass the entire software environment. The resultant Next Generation Intelligent Agents (NGIAs) are fully customisable in respect of their looks and personality. They can intelligently react and interact with the user, based on situations the user may find themselves in and provide assistance or entertainment in a friendly and informative way. The NGIAs can also incorporate sound and speech capabilities to enhance their interactive capabilities and increase their applicability.

2. Project Objectives

In order to create the VBP software emphasis was placed on developing and integrating the following areas:

- ✎ A range of body parts from which to assemble and customise the NGIAs.
- ✎ Software to provide interactive facilities between the character and the user.
- ✎ Software to interpret and assist the user in all situations.
- ✎ Software to allow multiple characters to communicate.
- ✎ Software to enable include speech generation and sound effects.
- ✎ A modular framework for program flexibility and expandability.

3. Relationship to Existing Products

This project links the areas of human-computer interaction (HCI) and distributed systems, both of which are extremely important to the computer industry. HCI is one of the fundamental areas of computer science [7] and techniques have been evolving and developing since the days of the first computer, when the interaction method was to use holes punched into a piece of card to supply a computer with a given task. With the invention of the keyboard users could more easily and readily instruct the computer through typing the requests into predefined fields or selecting combinations of keystrokes to activate predefined tasks. The invention of the mouse, gave rise to much more flexible and non-linear systems, where the user can click to select tasks. This also led to the first multi-tasking operating systems. Since then however, the field of HCI has slowed down, with new methods of interaction tending to be improvements and tweaks to the existing systems rather than radical changes to the way in which users interact with their computer systems. The VBP project intends to advance the field through exploiting natural language processing techniques in order to create as user friendly and intuitive HCIs as possible.

Distributed Systems are the current forefront of computer technology. The original distributed systems date back to the dumb-terminal and file-server systems from the 1970s. Such systems then dropped out of favour in the 1980's, when computer administrators chose more standalone systems with a shared printer. Recently though, with the popularity of the Internet (which is a massive distributed system) distributed systems are making a comeback. The VBP operates as a distributed system since all the programs can interact not just with the user, but with each other through information and task sharing. In this way the VBP program is capable of combining computers to solve larger problems in a shorter period of time.

There are many programs that cover different aspects of the VBP, but to date little attention has been paid to combining the features into a single coherent application such as that developed within the VBP. One of the most famous programs that enables a degree of user interaction and limited customisation is the Microsoft Agent system, more commonly known as the "Paperclip" [8]. Here the agent program attempts to offer ideas and assistance in relation to the users current task and provides assistance with any problems the user may have whilst using any Microsoft Office program. This is achieved by the user entering an English language string into a dialog box and the program matching it with words entered in the help system. If any of the help topics provide a match then the program offers this as a solution to the users query. With regard to entertainment, applications such as TopCat and EzSheep place a character onto the users desktop which interact in a fun way - EzSheep meanders about the screen and occasionally does something different and TopCat chases constantly after the mouse and falls asleep when it finally catches it. The Virtual Bodies Project takes these basic ideas and expands them into a system to provide full control for the computer environment.

On the distributed system side, there are still relatively few programs that allows users to interact with each other. The most commonly used system is ICQ (I-Seek-You), which is a system where every user

is allocated a unique identity number, and when their computer is connected to the Internet, ICQ registers them with a central server and finds out which of their friends are online. This then allows messages to be transferred between these friends; online chats to take place and for the user to discover which of their friends are currently connected to the Internet. A system similar to this operates inside the Virtual Bodies Project (VBP), except that the VBP system is more flexible, allowing not just the exchange of messages, but whole chats and interactive ideas exchange to take place through online chats and program sharing.

4. Overview of VBP Features

There are two main areas of development for the VBP system – the end user (interaction) software and the server-based (communication) software, a distributed system to support this program.

The end user software is further divided into two main components – the *Creation* program and the actual *Interaction* program. The *Creation* component of the software is where the user can create their NGIA character for their desktop, this is achieved using a supplied range of body parts – such as arms, legs, and heads. This range of body parts can be added to and extended at any time. This program also allows the user to name their character and to adjust its personality which directly relates to how much interaction the character has with the user of the software. Once the user is satisfied with the final design of the character, the program will set up all the necessary files on the computer and launch the *Interaction* component of the software.

The second component of the user-end of the software is the *Interaction* program, which the user can choose to load automatically at startup (when the computer is initially switched on) or invoke specifically when the user wishes to use the software. The program is designed to occupy the smallest amount of computer desktop space as possible, so as not to dominate or distract the user but to be easy to find when required. The program thus occupies a small window on the desktop, which the user is free to position where they wish on the computer screen in a similar fashion to the way in which Microsoft Agents can be positioned.

The *Interaction* program provides the user with various interactive features. The character will animate at random intervals to give the impression that the character is more three-dimensional than just a simple computer program. The program will also monitor the users activity and if there appears to be the characteristics of the user being, for example bored, then the program will identify this and suggest an alternative activity for the user to do. The level of interaction also includes the interpretation of English Natural Language (although other languages can easily be added into the system) for the understanding of instructions, help requests, general conversations and the like. For example, a user can ask their character to load a certain program, open a specified document, or to find a file; they can pass the character a mathematical equation and the character will reply with the answer; they can ask the character to find some information out for them on the Internet and have the results pasted into a

word document; they can play several games with the character; and, due to the design of the system, incorporate any new additional tasks easily into the system.

The *Interaction* program can also be used to communicate with other people anywhere in the world running the Virtual Bodies Project software, as the system is completely network aware, it can find via its central server, other *Interaction* programs in-use anywhere on the Internet and pass messages and pictures, and even have a chat between two virtual bodies characters over the internet, whilst carrying on with the rest of the users work.

The *Interaction* component of the software also includes speech generation via the Microsoft Speech API [9] (which has to be installed onto the users system), to allow the character to interactively talk back to the user. An expansion of this area is to include speech recognition so the user can pass instructions to the system aurally and enable users to talk to their computer and have it understand their requests. The current development version of the software is fully capable of supporting this feature, although it has not yet been associated with a speech recognition engine.

The user is also free at anytime to dynamically alter the design of their character on the fly. The changes made to the design of the character take effect immediately from when the changes are made without the system needing to be restarted, paused or receive any additional interaction from the user.

In summary, the *Interaction* software offers:

- ✎ **Character Creation software** – for the users to design their desktop character using a predefined range of body parts.
- ✎ **User Interaction software** – ways for the user to interact with the character.
- ✎ **Character Interaction software** – methods for the character to use software already available on the users computer.
- ✎ **Speech generation** – enhancing the interaction experience by the character talking back to the user.
- ✎ **Additional Programs** – to enhance the experience.

The server-based (*Communication*) part of the system is a single standalone program, which runs on a server based at Ision Internet. This program operates as both a Virtual Bodies server and as a web server. The Virtual Bodies Server allows the *Interaction* part of the software to use this server as a gateway to announce its presence to any other interactive programs currently in use anywhere on the Internet, and to talk to these programs directly. The server works as a web server by supplying a web front end, which can be viewed by any program able to view HTML web pages. This allows people to locate their friends using the system, since it can supply details about any users of the system. It also provides the facility to allow people to display a graphical image on their own personal website showing their current status – either *Online* or *Offline* – as and when they use their computer.

In summary, the *Communication* software offers:

✎ **A communication method** –to allow communication to take place.

✎ **Communication software** –to allow different users to exchange messages and conversations between each other

5. Design of the VBP System

Figure 1 shows the overall design and structure of the VBP software, including the main components developed for the software and the links between the software components.



Figure 1 Overall Design of the VBP Software

Each small white box on Figure 1 represents a module in the Virtual Bodies system whilst the shaded grey boxes show the layers of the program and where the modules are contained. For the Virtual Bodies Project itself (i.e. the character on the users desktop) this incorporates several modules distributed across various layers and dividing the program into the two main areas – *Interaction* and *Communication*. The *Communication* part of the software contains the modules used for communicating with the server and other Virtual Bodies characters on the Internet. The *Interaction* part contains the main interaction software, such as the Natural Language parser and the main character software. It also has a subsection of *Plug-ins*, which incorporates all the programs which can be used or included in the Virtual Bodies Project, but which are not included in the main software. These include from the Virtual Bodies system the mail services, and can include any program from the local computer.

6. Development

The Virtual Bodies Project has been developed as a mixture of J++, the Microsoft implementation of Java and Sun native Java version 1.1.8. Java has been chosen as the language for the program since this offers the best cross-platform capabilities of any language. This is due to Java operating inside a virtual machine on each computer such that the program source code does not need to be changed between different operating systems. The server part of the VBP system has been written in native java, allowing it to be completely cross platform and capable of being used on any operating system. This is very useful since most web-based servers are either running NT, Linux or Unix in equal proportions and writing the server in any specific language would restrict which platform is supported.

The actual character part of the software has been written in Microsoft J++. This is based on the Java language and allows native Java applications to be developed, but it also allows windows specific programs to be created – allowing the programmer access to windows specific API's (Application Programming Interfaces). This means that the software is capable of directly interacting with the native computer without going via the Java Virtual Machine, thereby enabling the software to directly interact with its environment. This has proved invaluable in the development of the software as the character can now directly interact with the computer user. The only parts of the character software written with native windows specifics are the Graphical User Interfaces (GUIs) to the software and hooks into the local computer system for User interaction. The rest of the character software has been written using native java code – allowing the character to be recompiled for other platforms – provided a specific GUI for that platform is created. As a ratio, the amount of windows specific code to native Java code in the character part of the software is in a ratio of about 1:5.

7. Server Software

The Virtual Bodies Project server is a single stand-alone application written in Native Java so it can be used on any computer capable of running a Java Virtual Machine. This computer needs to have access to the Ethernet with TCP/IP connection to actually be useful. The program is entirely console based (see Figure 2), although it does have a web (HTML) based front end for status monitoring (see Figure 3).

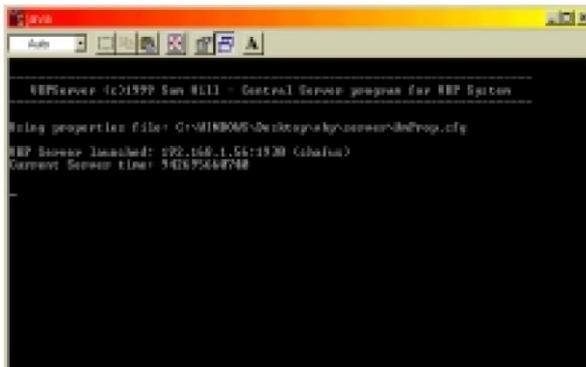


Figure 2 Console-based Program



Figure 3 Web-based Front End

This server's preliminary role is for the registration of Virtual Bodies Characters on the Ethernet to register their current status with the server. This allows other people to discover when one of their friends or colleagues is online using their Virtual Body character.

8. Character Software

As mentioned previously, the Virtual Bodies Project Character software is essentially divided into two distinct parts – the actual *Creation* program – where the user assembles their character from a range of body parts and the *Interaction* part where the user can interact with this character as shown in Figure 4.

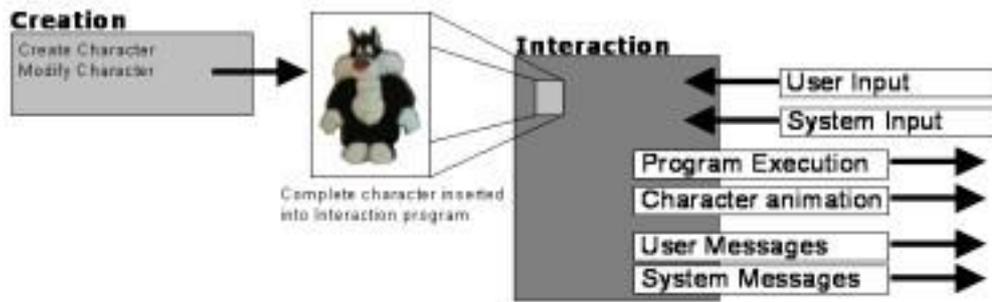


Figure 4 VBP Character Software

The main part of the program is the *Interaction* program, which is composed of a number of different modules, each fulfilling a different role within the VBP program. Each modules has been created as a separate executable program and may be summarised as:

VBP	The main module of the system
VBP-Talk	The natural language interpreter system
VBP-Conf	Used for conference facilities between users
VBP-Chat	To find out and communicate with any friends who are online
VBP-Mail	The VBP e-mail client
Talker	Speech Generation system

9. How the Character Operates

The **VBP** module is the main module for the Virtual Bodies Project *Interaction* program and acts as the main launching system for all the other modules, which are part of the *Interaction* program. The **VBP** module itself is launched when the program is first initialised and has the structure shown in Figure 5

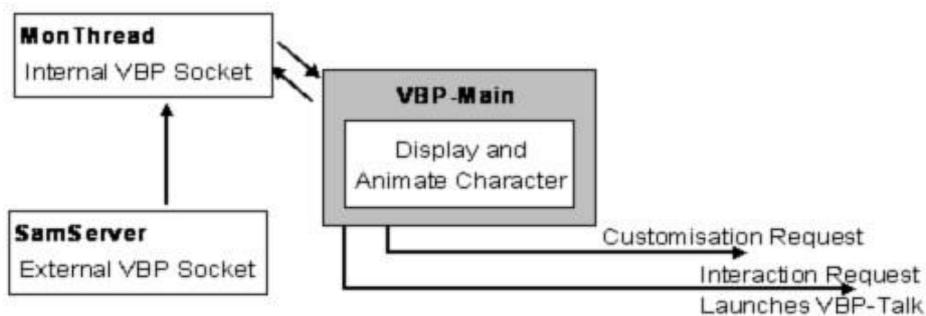


Figure 5 VBP Main Module Structure

This module operates by first opening two ports for communication. The first port (number 1922) is for internal system communications (*MonThread* in Figure 5); the different modules use this to share information between each other. The second port is 1921 and this is the external port and accessible by anyone in the world (*SamServer* in Figure 5) – this is the port used for communications to the Virtual Bodies Server, and for other Virtual Bodies Project characters to communicate with this character. The system is designed in this way to ensure the program is as stable as possible. The internal port is protected by an internal firewall unique to the system, which completely prevents anyone from outside the local computer from accessing this port. This has the great security advantage of fully protecting any personal information stored in the character and from any malicious users of the Internet from affecting

the program. The externally accessed port is kept structurally separate from the system, but is left fully open for anyone to access the port from anywhere in the world. This means that the port is open to attacks from any malicious users, but careful programming prevents this from causing too many problems. Every time a request is received on this port, the program launches a new thread to deal with this request and reopens the port for any subsequent connections – this means that any number of requests can arrive at the port at anytime and all requests will be dealt with – there is not a queuing system in use. If a request to the port is not a recognised request using the **VBP Protocol** (described and defined in section 14) then that connection is closed and the thread is terminated. Also, if a request is operating too slowly or not enough information is received then a timeout engages, the connection is closed and the thread terminated. Both of these ports contain internal self monitoring software, so if for any reason the port becomes blocked or connections are declined, then the monitoring software will terminate all threads running on the port having trouble, close down the port fully and re-launch itself reopening the port for connections. This is designed to operate fully transparently to the user of the system, so the actual Interaction software would not restart just the internal port handing software. This should ensure that this part of the software is as stable as possible and as resistant to hackers as can be arranged.

Once the program has initialised the ports, the main screen appears as shown in Figure 6.



Figure 6 Main VBP Screen

The program is divided into three areas – the menu bar at the top of the window contains all the different options for the creation and modification of the character, to add plug-ins to the system and to generally tweak the environment to a configuration that meets the requirements of the user. Below this is the character graphical representation and below this the button used to actually interact with the system as clicking on this launches the **VBP-Talk** program.

From a users perspective the program appears to be in a running yet inactive state (except for occasionally animating the character) and waiting for users input. In actual fact the system is busy processing information, which is received on the ports of the program and in communication with the central VBP server. This means that if a VBP character in use on someone else's computer chooses to send this person a message, then this message is received, interpreted and displayed to the user. Also if any system messages are received (such as the latest news headlines, share reports, or the lottery numbers) then these are also displayed by the system. Any programs operating on the users computer can also use the Virtual Bodies Character to display a message to the user of the system. Such as a

message from a program of “File has not been saved. Do you wish to continue?” can be displayed in a more friendly way through the VBP character than as a system message.

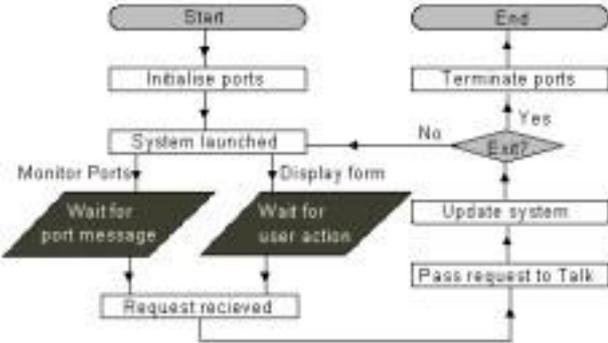


Figure 7 Flow Chart of VBP Module Operation

The flow chart in Figure 7 shows how this program operates at a high level. Since the program is fully multi-tasking and multi-threaded, any of these stages can occur at the same time. The **VBP** module does not deal with the requests and messages internally, these are passed across to the **VBP-Talk** module, which is described in Section 10. The system updates are when the user requests to modify the character design or requests the character animates, in which case, the request is handled internally and the system changed to reflect these requests.

10. Understanding Requests

The **VBP-Talk** module deals with all requests, system messages and anything else to do with the actual system. This module contains the Natural Language Parser, message display facilities and Internet search facilities etc. This is essentially the main program where the actual interaction with the system takes place.



Figure 8 VBP-Talk Bubble

When the **VBP-Talk** module is launched, it displays as a yellow rectangle, much like a speech bubble, containing a text box and two buttons (see Figure 8). The user of the system can then enter a sentence into the box, which is then passed to the internal Language Parser, which attempts to analyse the sentence and establish what the user is asking or wishing to do.

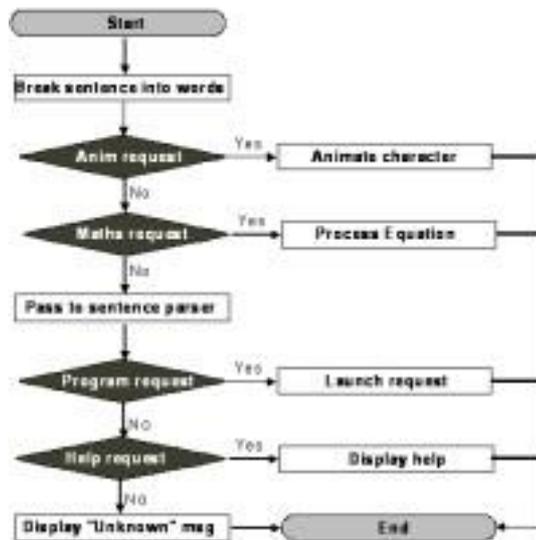


Figure 9 Flow Chart of VBP-Talk Module Operation

Figure 9 shows how the parser system works. Firstly the sentence entered is split into its component words and all punctuation, case and certain words appearing in an 'ignore list' are removed. The resultant string is then passed to various analysers, which evaluate the sentence to see what the user is trying to do. If no match is eventually found the user is asked to try again with their input.

Each program to be supported by the Virtual Bodies System or VBP plug-ins are described by a series of keywords. For example, the Microsoft Word application may have the key words "word, document, letter, write, create" associated with it. If any of the words contained in the sentence match a keyword for a program or plug-in, then this program or plug-in is assigned a ranking value. The higher the ranking value the better the match. Once the sentence is parsed, in theory a single program or plug-in should have the highest-ranking value. This program corresponds with the users request of a program to load. If more than one match has an equal rank then the closest choices are displayed and the user is given a choice of program to load – such as if the user enters "do e-mail" the parser is unsure whether the user wishes to check their e-mail or send e-mail or perhaps both. Since different programs can be assigned to these different tasks, the user is given the choice of which program to choose to use.

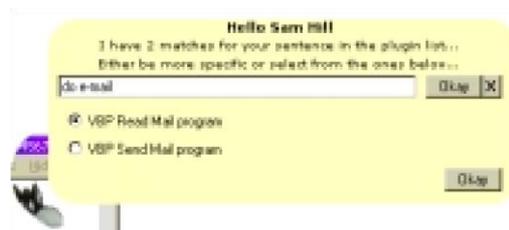


Figure 10 Program Options for User Selection

11. Interactions with Other Users

Interaction between users can take place in many forms such as finding which friends are on-line (Figure 11), sending them messages or instigating on-line chats with them (Figure 12).

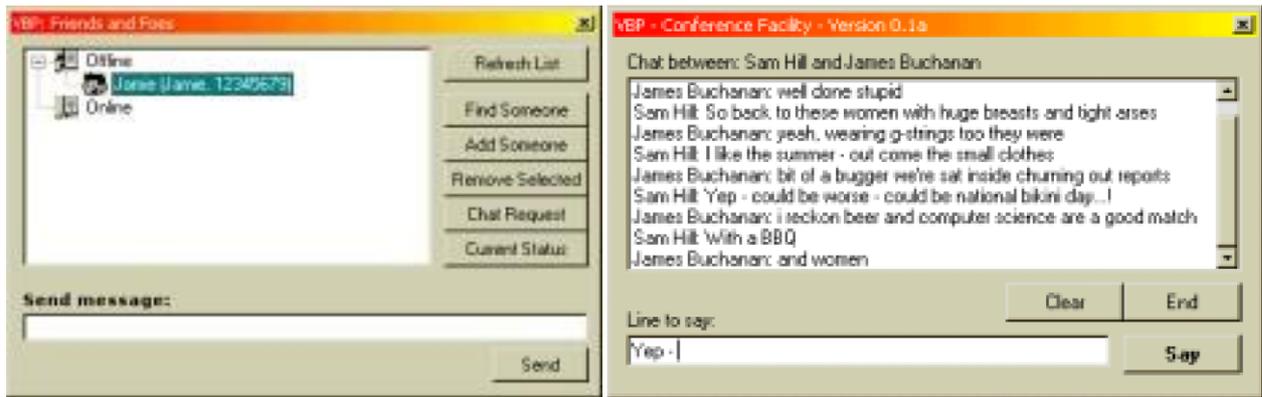


Figure 11 Finding friends & Sending Messages Figure 12 On-line Chat Session

12 Creating a Character

The Character Creator is a single standalone program where the user can assemble their desired character from a range of body parts. Figure 12 shows the **Virtual Bodies Bank**, this consists of a single screen where the character is created. At the top of the screen is a series of banks, which contain given body parts – the first, for example, contains a range of heads for the character.



Figure 13 The Virtual Bodies Bank

Clicking on one of these banks empties out the body parts into the selection tank underneath the body banks, this displays previews of the body parts, the full range of which can be viewed by turning the wheels at either end – the one on the left scrolls the range to the left and the one on the right, to the right. Clicking on a body part in this tank causes that body part to be placed onto the character in the character holder, where a full preview of the final character is displayed.

When the user is satisfied with the final character, they can assign the character a name by entering a new name into the name text box. The VBPID number is the unique ID number assigned to their character, this number is set by the system and should not be changed, although if required or if the user reregisters or changes their registration details then this number may change and need to be modified and so the facility is provided. Finally, the slider for Personality is used to define the language used by the character to reply to the user and also how animated the final character is – a 'Calm' choice

means the character is fairly static and very polite, an 'Aggressive' selection produces a very animated character which uses much stronger language back to the user. Once all the settings are complete, clicking on Okay stores the created character and dynamically updates the VBP character displayed on screen if the program is currently running.

The body parts supplied with the program were created by capturing animations from real video footage (Figure 14), breaking this down into the component parts and converting these into animated GIF images. Most of the body parts supplied with the program have been captured from cuddly toys. Firstly the character is animated in front of a video camera – using the fishing-wire technique for animating the character. Each body part to be animated has a piece of wire attached to it and by moving this wire creates the resultant movement. This video clip is used as the basis for the animated body parts. The clip is broken into areas, each area represents a body part from the character, these areas are then extracted from all the frames of the animation, the background is converted to transparent (for later layering of the body parts) and a final animated body part results (Figure 15). This process is then repeated for all the other characters included in the program. This includes human limbs, which were captured using the same method.



Figure 15 Animating a Character
Figure 14 Creating a Body Part



Figure 16 Creating a

Character

The system used for the body parts in the program is the Animated GIF format, invented by CompuServe. This format has been used since it is flexible enough to support animation and also transparency effects very easily. This means that when a body is assembled for use as a character in the program, the character can be assembled out of a series of layers (Figure 16). Each layer represents a part of the character and can be manipulated independently of the rest of the character – allowing a very flexible range of animations to be taken. To animate the characters head, the only change needed to the character is to change the image displayed on the head layer (the top layer).

This would cause the animated head to be displayed on top of the rest of the character, which is not viewed as individual layers but as a single coherent character. This makes for a very flexible character creation system as any choice of head, body, arms and legs can be placed together in combination, each with its own animation and which can be displayed independently of any of the others. For instance,

the character can be made to wave its left, right or even both arms at any time, along with animating its head if required by the user. All these body parts could come from different character graphic ranges – such as Eeyores head on the cats' body, with alien arms and legs.

13. Extending the System

The Virtual Bodies Project can be easily extended by adding additional plug-ins to the system to extend the range of features supplied to the user. A plug-in to perform any task can easily be added to the system through the use of the **VBP-Plug-in Manager** (Figure 17).



Figure 17 VBP Plug-in Manager

Each available plug-in has an associated executable program – which is launched for the plug-in; a description which is used to explain what the plug-in is or does; a list of keywords which associate with the plug-in; and a Parameters number – which are internal defined parameters – such as values to pass to the program on the command line – this could be either a preset number which the plug-in requires or a variable, the difference is determined by the % symbol.

The example in Figure 17 shows the configuration for the VBP Read Mail plug-in. This uses the **VBPMail** program, has the description of “VBP Read Mail program” and a series of keywords describing the program. The parameters box contains the value 1, which is past directly to the VBP-Mail program, this means to the VBP-Mail program – launch in Read mode; for sending e-mail, this parameter would have the value of 2, meaning launch in Send mode.

14. VBP Data Structures

Several data structures have been developed for use with this system. The main data structure is the communications protocol used for all communications between the various parts of the VBP System. These include communications between a VBP Character and a VBP Server; and also communications between plug-ins and the VBP Character. The VBP Protocol takes the place of layers 5 to 7 of the standard TCP/IP Protocol as shown in Figure 18 [adapted from figure 15.13 in 10]

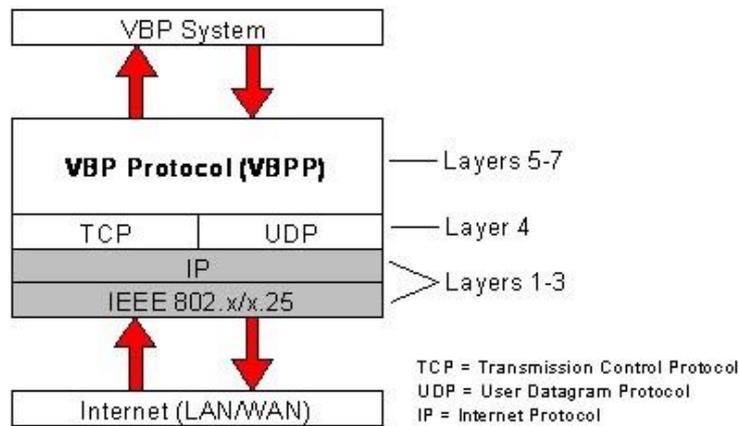


Figure 18 VBP Protocol layers

The VBP Protocol works by using the standard TCP/IP interface. This is chosen over UDP as UDP packets are transmitted in a connectionless state – the message is sent and the program then waits for a reply with no idea of whether the message was in fact received or if it was a reply was actually transmitted. By using TCP, this is a connection-orientated method where a direct connection is established between the two Ethernet (i.e. international networks such as the Internet, LANS, WANS and WAPs) devices enabling direct bi-directional communication to take place between these two ports.

The VBP protocol sits over the standard TCP/IP layer, with the result that it was easier to implement than sending a raw IP packet across the Ethernet. TCP/IP operates by each and every host being assigned a unique host identification, which consists of a 32-bit integer host identification number (ID) in the form of xxx.xxx.xxx.xxx; and then each service on the system operates over a unique port number. The messages sent using the VBP protocol are similar in nature to FTP and Telnet service commands. That is an instruction is sent from the client to the server, which the server acknowledges and requests the any additional information for that instruction, processes the information and then replies with either a second acknowledgement or the information requested.

Figure 19 shows the message flows through the system from a client to a server (here the server is not restricted to the VBP Server).

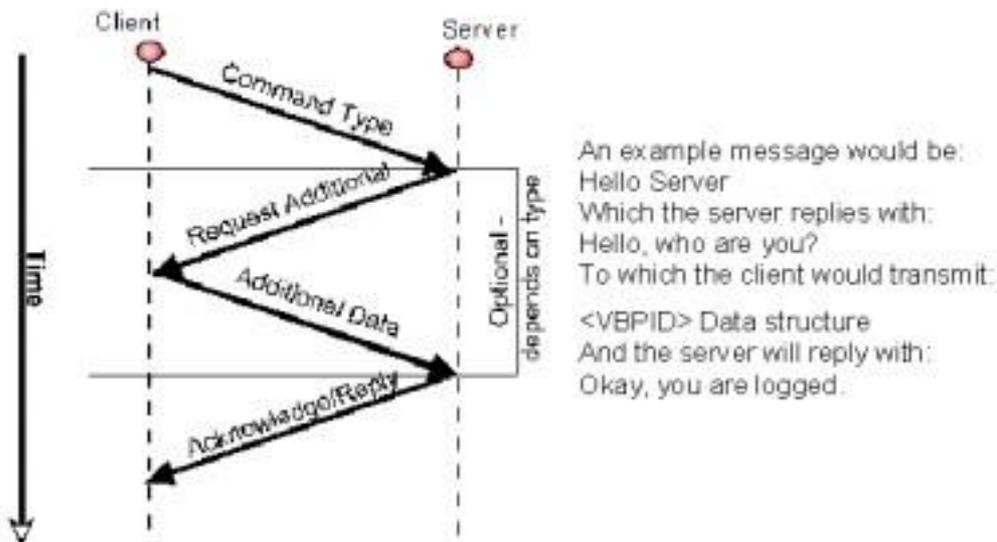


Figure 19 VBP Client-Server message Flow

15. VBP Communication Processes

Figure 20 shows the information flows through the Virtual Bodies system. It shows the communication routes between the VBP system, the real world and other applications. The only route to access the VBP System on any local computer is via the `SamServer` connection. This is a one-way socket, only enabling connections to it, and does not provide any information back. This then feeds the information into the VBP System by the `MonThread` (which is short for Monitor Thread as it monitors all incoming messages). This `MonThread` also passes messages back to the outside world. A program running locally on a computer can communicate directly with the VBP system through its `SamServer` socket.

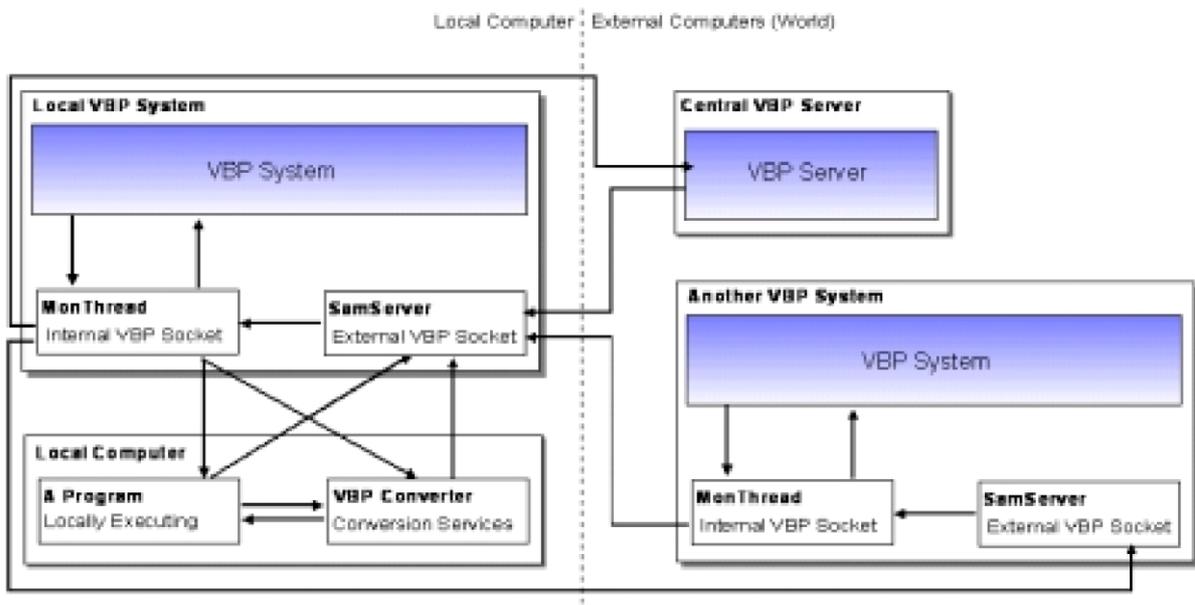


Figure 20 Information Flows Through the VBP Server

Other VBP characters and the `Central Server` also communicate to other VBP's via the `SamServer` socket, to ensure complete security and stability and consistency to the system. The arrows on Figure 20 show the routes of information flow on the system and the diagram needs to be expanded for each individual plug-in or character running, which can communicate to the system. The program shown could be any Virtual Bodies plug-in. Figure 20 is a very simplified version of how the information flows – for example when a chat service between two VBP's is initialised the following events would take place:

- The `Central Server` is asked if a certain character is online.
- If the character is online a chat message is sent from the VBP requesting acceptance to the otherVBP.
- This can then accept or deny the chat request.
- If it is accepted then the Chat module needs to be launched:
- This then creates a direct connection between the Chat module as part of one VBP system and the chat module on the other VBP system.
- This chat module also communicates back to the central VBP Server what is going on in the chat – and also makes requests of the VBP System as events take place.

This whole system become increasingly complicated as more and more VBP characters begin to interact with each other and their respective plug-ins. Messages and communication lines can easily flow in almost any direction across the system between any number of these programs and 'spaghettification' of the system could occur due to the sheer number of communication lines taking place between the different parts of the system.

Referring back to Figure 20, the shaded boxes illustrate the two main VBP systems – the `Local VBP` system that consists of the character and the `Central Server` – which runs remotely and is there to enable communications to take place between two different VBP systems. Considering the local VBP System first, this actually consists of two basic programs – the first is the VBP character software itself, which consists of the receptor for the incoming messages from `MonThread`, the animation facilities for the graphical representation of the character and the basic system structure. When the user wishes to interact with the character, then a separate program is launched, which contains an English language parser to interpret the requests from the user and act upon the requests in the required manner. The typical result of this is to shell to another modular program – such as a request to Chat to another VBP character would launch the Chat; a request to check e-mail would launch the mail services module, or indeed any other plug-in modules the user may have installed on the system all of which are free to communicate to each other or the central VBP system.

Conclusion

The VBP project provides a simple software package that can be easily installed on a users computer, where the NGIA character produced can be fully customised with respect to looks and personality.

The character intelligently reacts to the user through the VBP-Talk interface module evident through the natural language entered into the program being intelligently parsed and assessed to produce a result, which assists the user, either by launching the desired software package or by offering help. The character also animates itself to the user in an amusing and entertaining manner whilst displaying the information in an informative way. The speech extension has been included to the level of speech generation (albeit through the use of a third parties software package), and the system has been designed in such a way to enable speech recognition to be added to the system as a later date to further improve the accessibility aspects of the software.

This project has brought together many different disciplines of computer science and existing projects and programs into a single package and in doing so created a new and unique system for Human Computer Interaction, which has not been created before.

This project also presents new ways of communicating and developing communities over the Internet. This project offers the chance for anyone using the Virtual Bodies Project to exchange ideas, messages and at a later date programs with each other directly allowing a new way of shared working to develop. This is enhanced by interaction through the natural language parser and speech generation software such that the Star Trek ideology of talking to a computer and having it understand your requests becomes a step closer to reality. Many of the ideas and methods presented in the project could easily be adapted for use in other projects; the natural language parser especially could form the basis for a new operating system GUI shell, where the mouse and keyboard may become a thing of the past.

Possible Future Work

There are a number of areas where this project can be further developed and expanded at future dates. The entire system is developed in a modular fashion, with each part of the system operating as a fully independent program – so any part of the system can be upgraded without effecting the rest of the system and new parts (or modules) can be added frequently and easily to the system to add additional features, services or other improvements.

Simple areas for improvements would be the inclusion of support for more than just the English language; the system is designed to be able to support more than one language, although only English language has been implemented. Another area is the inclusion of speech recognition to the system – this merely involves typing a specific speech recognition engine into the interaction system to pass the recognised text from the engine into the natural Language parser built into the system – meaning that instead of the user typing requests into the system these requests can be spoken into the system.

A longer term objective is to make the Virtual Bodies Project portable using WAP [11] technology whereby data can be transferred directly from the Internet down to a mobile phone anywhere in the world. Transferring the Virtual Bodies Project into an interactive character that resides on a mobile phone would be a very interesting expansion of the system and the technology is available to support this. The

changes needed would be to switch where the bulk of the software operates from, that is, instead of the character running directly on the phone, all that would need to reside there would be the graphical representation for the character and facilities to pass user input from the system back to the server which would interpret the results and pass these back to the mobile phone. Amalgamating this with the speech recognition software as suggested above would make this system very interactive and completely mobile. This is an exceedingly innovative and exciting use of the technology but which has not yet been attempted due to time restraints.

Alternative uses for the technology, as touched on earlier would be to develop and integrate the natural language parser part of the software into a new computer operating system and perhaps one day enabling a computer to respond to voice commands making the mouse and keyboard a thing of the past.

Acknowledgements

The authors would like to thank Ision Internet plc, www.ision.net.uk, for supporting this work and allowing the VBP program to reside on their Sun Solaris server in Telehouse, London, UK.

References

-
- [1] Harlow, J. and Gadher, D. *The Second Internet Revolution*, Focus, The Sunday Times, Oct 17th 1999.
 - [2] <http://gopher.igc.apc.org/00/orgs/pacifica/tel/1>, **Technophobia: The Social Disease of the '90s**, 1993
 - [3] Channel 5, **Lunchtime News**, May 27th 1999
 - [4] James D. Best, author of **The Digital Revolution**, 1999
 - [5] Shneiderman, B., **Designing the User Interface: Strategies for Effective Human-Computer Interaction**, 3rd Edition, Addison-Wesley, 1998.
 - [6] McCrindle, R.J., **The Impact of New and Distributed Technologies on Disabled Users**, Proceedings IEE European Workshop on Distributed Imaging, London, November 1999.
 - [7] Dix, A., Finlay, J., Abowd, G. and Beale, **Human-Computer Interaction**, 2nd Edition, Prentice Hall, 1998.
 - [8] Microsoft, www.microsoft.com/windows/98 (**Windows 98 overview**), 1998.
 - [9] Rozak, M., **Overview of Microsoft Speech API**, Microsoft, 1998.
 - [10] Silberschatz and Galvin, **Operating System Concepts**, Addison Wesley, 1998
 - [11] Hill, S., www.samhill.co.uk/computers/prog/wap.htm or www.wapalize.co.uk, **Wireless Application Protocols**, 2000

Authors

Sam Hill works as a Developer for ISON Internet's UK office. His job entails research and development of the latest tools and systems for the Internet, including automation facilities, search engine technology, webmail, dynamic content based websites, pgp encryption technologies, fingerprint recognition software, user-interface design and methods, and many other future technologies. Other work has included TV appearances on 'www.personal services' on channel 5, explaining how the internet operates and how it can be used in particular for stream media such as webcams. He has also presented a series of programmes on .TV the technology channel on Sky entitled cre@te, where different ideas and methods for producing and creating material using computers is discussed.

Rachel McCrindle is a lecturer in the Department of Computer Science at The University of Reading. She has recognised achievements in the development of innovative graphical user interfaces, multimodal interfaces and multimedia environments and their application to real-world problems. She has developed multimedia systems to enable the development and maintenance of software systems; documentation of company-wide information bases and more recently the application of such environments to inclusive design and development of software products for the disabled. Particularly exciting is the development of an interactive gaming environment for the blind, based on an ambisonic (3-D audio) and force feedback environments. Other key research is being conducted in the areas of digital video broadcasting, intelligent agents for desktop interaction, co-operative working and virtual community building. She is responsible for teaching GUI, multimedia and web-based design at undergraduate and postgraduate level; is a member of the BCS Disability Group and sits on a National Advisory Panel with regard to advising on user interface accessibility issues for the disabled and elderly members of the population.