Using the Resources Model in Virtual Environment Design

Shamus Smith, David Duke and Peter Wright

Department of Computer Science, University of York
Heslington, York, YO10 5DD, U.K.
(shamus, duke, pcw)@cs.york.ac.uk

Abstract

Traditionally, the development of virtual environments (VEs) has been limited to particular technologies and the associated constraints on environment definition and interaction. However, with advances in both graphics hardware/software and the availability of new input/output devices, many of the restrictions on VE development have been removed. As VEs become increasingly realistic, there is a need to investigate where requirements and design information are located within these environments and how information can be structured and analysed. This is complicated by the close binding between the user and the environment. The aim of many systems is to make the man/machine barrier transparent so that the user feels "immersed" in the environment. To enable this, the virtual environment designer must be aware of the information resources that are available in the man/machine system and how they are distributed.

In this paper, we approach the description of VEs in terms of distributed cognition. Distributed systems can be studied by analysis of the way in which distributed representations are co-ordinated, propagated and transformed. By analysing the process of representation and re-representation we are able to account for how tasks of users in a system are transformed and made more or less difficult. As a starting point for the cognitive analysis of this information, we propose to use the Resources Model (Wright99). We demonstrate how the Resources Model can be used to represent and understand interaction in a way that can be used early in the design phases of VE development.

Keywords: Virtual environments, Resources Model, distributed cognition, resources, virtual environment design, interaction.

1 Introduction

Traditionally, the development of virtual environments (VEs) has been limited to particular technologies and the associated constraints on environment definition and interaction. However, with advances in both graphics hardware/software and the availability of new input/output devices, many of the restrictions on VE development have been removed. Rendering rates are now reaching acceptable levels on desktop machines, new interaction technologies are being enabled (both through physical devices e.g. Spaceballs and 3D position trackers, and logical metaphors e.g. cursor based flying (Massink99)) and high-level application programming interfaces (APIs) are being used in the VE development process.

As these technologies mature and become adopted in a wider range of applications, there is a need to better understand how this technology can be accommodated in software engineering practice. This process is being enabled in part by the development of 'generic' virtual environment toolkits such as dVise and SuperScape. It is difficult to find reports that detail the *process* used to develop virtual environments, but given the maturity of the technology it would seem reasonable to suggest that prototyping and exploratory development play a significant role. However, if or when the technology of virtual environments becomes adopted in mainstream software systems and products, exploratory approaches become rather less attractive. Software developers must be concerned with making use of the most appropriate technology in a way that meets the requirements of the client, including quality criteria such as usability, robustness, maintainability, error-tolerance, etc.

The INQUISITIVE project is a three-year research effort funded by the UK Engineering and Physical Sciences Research Council between groups at the University of York and the CLRC Rutherford Appleton Laboratory (RAL). The aim of the project is to develop methods and principles that can be used to improve the design of interfaces for virtual environments. We see this as progress towards a VE design methodology to allow developers to move from requirements (domain, user, etc) of a system to the final implementation of a VE application.

One tension that is evident in VE systems is between the desire to interact freely with a 3D space and the entities within it, and the limitations imposed by the available programming interface. We are particularly interested in looking at how end-user requirements on the interface can be implemented via a VE toolkit. Recently, several toolkits for virtual environments have been developed, for example (Mr95; Dive97; Maverik99). These provide a toolkit layer to insulate the VE application designer from the low-end VE implementation issues.

A number of models that represent user interaction at an abstract level have been developed specifically for VE interfaces. (Massink99; Smith99a; Smith99b) show how these models can then be subsequently specified to increasingly detailed descriptions, thus moving the design closer to implementation. At an abstract level, the interaction models can be refined towards the interface supported by a given toolkit. However, the specification of interaction within a VE is only one component in the design process as the definition of other, complementary, information requirements are needed for designs to move to an implementation through a toolkit.

In this paper we take a step back from the formal specification of VEs to investigate where requirements and design information are located within these environments and how it can be structured and analysed. More specifically, we are interested in considering VEs in terms of distributed cognition (DC) (Zhang94; Hutchins95; Scaife96; Fields97).

The remainder of this paper is organised as follows. In Section 2 we briefly consider the distributed cognition position and in Section 3 we introduce the Resource Model as a distributed cognitive model of HCI. Sections 4 to 7 present four examples of the use of the Resources Model for VE design. In Section 8 we finish with some conclusions and directions of future work.

2 Distributed cognition

The DC position is that the appropriate unit of analysis is not an individual but rather a distributed cognitive system of people and artefacts. The means by which this distributed system is studied is by analysis of the way in which distributed representations are co-ordinated, propagated and transformed. By analysing this process of representation and re-representation we are able to account for how tasks of humans in the system are transformed and made more or less difficult. For modelling human computer interaction (HCI), the DC paradigm has some obvious attractions. It might be used to understand how properties of objects on the screen can serve as external representations and reduce cognitive effort (Scaife96). It can also serve to bring together work on computer supported cooperative work (CSCW) and HCI by considering how technology mediates the propagation of representations between individuals. The deliberate softening of the boundary between the user and system inherent in the distributed cognition view also brings into focus the design question of the information requirements for interaction. For example, what information is required in order to carry out some task and where should it be located? Should it be as an interface object or as something that is mentally represented by the user?

In traditional computer systems, users interact with the computer-based environment from outside the environment itself. However, VEs provide a situation where the user can be immersed (physically and possibly mentally (Smith98)) within the environment. Although some elements of the environment may be well-defined, for example physical layout, other aspects are more troublesome. In particular the organisation and presentation of interaction cues and capabilities. The location of information, either represented externally in the artifact or internally by the user, may have wide-ranging effects on a systems usability (Scaife96; Wright99). As a starting point for the cognitive analysis of this information, we propose to use the Resources Model (Wright96; Fields97; Wright99) as a distributed cognitive model of HCI.

3 The Resources Model

The Resources Model is comprised of two components; *abstract information structures* and *interaction strategies*. Wright, Fields and Harrison (Wright96; Fields97; Wright99) identify six abstract information structures that can be used to classify the types of information that inform interaction; **plans**, specifying actions to be performed; **goals** and sub-goals to be achieved; the **current state** of the world or interactive system; historical information (or **interaction history**) about previous actions and what properties held of the state in the past; an **action-effect** model of the effect actions have on the system; and the set of **affordances** that the system currently supports. These structures can be grouped to make up a *resource configuration*. A resource configuration is a collection of information structures that can be defined for each step in an interaction and which can be used to inform action (Wright96). These resources can either be external in an interface or represented in the head of the user. When action is taken in an environment, the resource configuration changes. For example, the current state and the history structures may change. An interaction sequence can thus be seen as a number of steps between changing resource configurations.

The second part of the Resources Model is the *interaction strategies*. Interaction strategies are the linkages between resource configurations that can be used to make decisions about what action to take. Wright, Fields and Harrison (Wright99) define four strategies; **plan following**, **plan construction**, **goal matching and history-based selection/elimination**. They describe how interaction strategies presuppose certain configurations of resources to make them effective and conversely, how a configuration of resources can make a particular interaction strategy possible. By making the organisation of information explicit, the model facilitates reasoning, analysis and comparison.

A more in-depth description and motivation for the Resources Model can be found in (Wright99) where the use of the Resources Model to analyse interaction is described. Here, we take a more general approach to the application of the Resources Model as we feel that in VE design, the distributed cognition direction provided by the Resources Model can offer several useful insights for VE design.

This paper is an initial step towards applying the Resources Model within the virtual environment design process. In the following sections, we will demonstrate four examples of how the Resources Model can help VE design.

4 Designing interaction episodes

The Resources Model can be used to identify parts of an interaction episode where the scarcity of externalised resources places heavy demands on the user's knowledge and consequently might shape their choice of interaction strategy (Wright99). Wright, Fields and Harrison (Wright99) analyse of the Microsoft Excel chart wizard dialogue allowed them to retrospectively identify a number of characteristics of the design, which enhance usability, and also some potential weakness in the resources provided by the designers of the wizard.

In this section, we wish to use a similar approach in the design of VE interaction episodes. This has two main phases. Firstly, an analysis of the interaction episode in context of the real world to identify the required resources and secondly, definition of alternative resource configurations for VE implementations.

A simple task (but one that is common in many immersive VEs) will be used as an example. This is the task of pushing a button to turn the lights on in a room. We assume that the user has navigated to within reaching distance of the button (mounted on a wall) and focus on the task of reaching out and pushing the button. There are three states (each with their own resource configuration) associated with the two actions ("reach for button" and "press button"). Tables 1, 2, and 3 show these resource configurations in the real world. (NB. Irrelevant resources, at each step, have been omitted for clarity.)

Resource	**Relevant content**	**Resource location**
Goal	Move hand to button	User - mind
Current State	Hand at side Button out Lights out	User - felt by user Environment - seen by user Environment - seen by user Environment - seen by user
Affordance	Reach for button	User - mind
Action Effect	Hand at button	User - mind

Table 1: Task is "Press the button".

Resource	**Relevant content**	**Resource location**
Goal	Press button	User - mind
Current State	Hand at button Button out Lights out	User - felt by user Environment - seen by user User - felt by user Environment - seen by user Environment - seen by user
Affordance	Press button	User - mind Environment - seen by user
Action Effect	Button in, lights on	User
Interaction History	Reached for button	User - mind Environment - seen by user

Table 2: After action "Reach for button".

Resource	**Relevant content**	**Resource location**
Current State	Hand at button Button in Lights on	User - felt by user Environment - seen by user User - felt by user Environment - seen by user Environment - seen by user
Interaction History	Button pressed	User - mind Environment - seen by user

Table 3: After action "Press button".

In this example, there are three identified resource locations; the cognitive knowledge internalised in the user, the internalised tactile feel of the user's touch and the externalised visual presentation of objects and the environment. (Depending on the type of button/switch an externalised audio resource may also be present.)

The first thing that is obvious from the description is that as the user gets further into the interaction, the tactile sense plays an increasingly important role. For example, the resource location of the "button out" information. In Table 1, this information is only presented via the externalised visual but in Table 2 the user's feeling of touch augments this. Although, in many situations, the visual of the button would be sufficient to convey this information, it is not difficult to imagine situations where this is not the case, e.g. an environment visually impaired by smoke, fog or water, or a low resolution VE. In a VE without tactile feedback, this information would have to be presented via an alternative source. Therefore the feedback from the tactile sense channel must be moved to a different channel. In the current example, the visual channel is already being utilised, so the next logical choice is the audio channel. However, designers must be aware that such channel switching can have unwanted side effects on an implementation. For instance, the audio channel may become overloaded or the tactile to audio feedback mapping may require special user knowledge.

The resource definition above also identifies where the tactile feedback is required (in addition to the externalised view of the environment). This is between the hand and the button. The user needs feedback on when the hand is at the button and when the button has been pressed. Possible alternatives for VE implementations typically involve the use of various collision detection feedback techniques. For example, object (hand/button) collisions can be indicated by visual feedback (button goes green on collision) or an audio clue ("clink"). Alternatively, this may be combined with or without visual collision feedback, e.g. the hand stops when it hits the button. (However, this has the disadvantage of breaking the physical/logical hand mapping with the logical hand stopping while the physical hand can still move forward.)

Although the distribution of resources in this example are almost equally spread between the internal (mind (7) + touch (5)) and the external (environment (11)) usability may be improved by externalising resources. For example, the affordance that the user can "reach for the button" is only in the mind of the user. One way to externalise this would be to have the image of the users' hand pose change when in the proximity of a selectable button. Here an affordance of the object is being conveyed to the user via visual feedback. A similar technique is used in QuicktimeVR to indicate interaction options. The usual arrow cursor changes to a hand cursor to indicate that an action can be taken at the current cursor position. Such context sensitive cursors are common in many GUI applications but are surprisingly rare in VE applications.

Another example externalisation that is common in many modern user interfaces is the use of a pop-up text box to indicate the function of the current object of interest (Wright99). This can serve as an external representation of the action-effect mapping, providing a resource that aids matching the user's goals with the current affordances. In this example, a "lights on" pop-up could appear (or a "lights on" audio message) when the user's hand is over the light button.

5 Device classification using the Resources Model

One of the difficulties of using VEs is that although interaction is presented as being more "intuitive" than traditional computer based systems (Erickson93), this is rarely the case. There are many novel physical devices for interaction in VEs (e.g. Spaceballs, 3D mice, data gloves, etc) but it is typical for device manipulation to be intrusive and/or without a close mapping between the physical device and the perceived virtual representation. For example, consider the 3D mouse. A typical VE configuration for this device has a magnetic tracker embedded in a joystick grip for added button functionality. Hardware constraints on the range of the magnetic field limit the distance the user can be from the transmitter (without experiencing visual "jitters" caused by tracker lag). Also there are cumbersome cables which can interfere with the device's use.

Norman notes that computer systems already come with built-in physical affordances. The computer, with its keyboard, screen, pointing device and selection buttons affords pointing, touching, looking, and clicking on every pixel of the screen (Norman99). This is also true in the virtual world although there is less separation between the devices and the environment. In many VEs, a rendered disembodied hand commonly represents the current position and orientation of the 3D mouse. Therefore the user has to mentally map the physical affordances of the device (a joystick grip with buttons) onto the view of a virtual hand. There are several mismatches that can make this mapping difficult. Firstly, the virtual hand displays a static pose. Hands in the real world allow poses to be formed and gestures generated. Secondly, the virtual hands are usually rendered in an open hand (pointing) pose. The user, however, will have a closed hand, gripping the 3D mouse. Thirdly, the user has the tactile sense of holding the device and pushing the buttons. This sense information is not commonly represented in the virtual world.

The failure to provide a congruent conceptual mapping from the perceived qualities of virtual representations to the physical affordances of physical devices helps to confound the "intuitiveness" of VE interactions. This also intrudes into the user's "VE experience" and therefore interferes with their possible degree of immersion.

If these types of mismatched mappings can be identified and eliminated in the design process, we feel that it will significantly simplify VE design and benefit the usability of the final system. Within the HCI literature, the classification of interaction devices has been focused to physical, logical and input/output devices and their comparison for particular tasks (Card90). However, the Resources Model provides a unique way of looking at devices independent of their type and context of use. Devices can be classified by the types of Resources Model information structures that apply to them (e.g. current state, affordances, action-effect etc).

Device	**Sense**	**Resource**
2D Mouse	Tactile 2D displacement Button press Surface feel Visual Physical device Audio Button click Friction noise	Current state 2D displacement User - Propreceptive User - visual, audio Affordance 2D Arm affordances
6D Tracker - hand (e.g. Polhemus)	Tactile 6D displacement Surface feel Visual Physical device	Current state 6D displacement User - Propreceptive User - visual Affordance 6D Arm affordances
6D Tracker - head (e.g. Polhemus)	Tactile 6D displacement Visual Physical device	Current state 6D displacement User - Propreceptive Affordance 6D Head affordances
Data glove	Tactile 6D displacement Surface feel Visual Physical device	Current state Hand pose User - hand/arm Propreceptive Affordance 3D Arm affordances Hand affordances

Table 4: Physical device classification.

Device	**Sense**	**Resource**
2D Mouse pointer	Visual rendered image	Current state Image, 2D location Perceived Affordance 2D Cursor Affordances
6D Tracker pointer	Visual rendered image	Current state Image, 6D location Perceived Affordance 6D Cursor Affordances
Hand Avatar	Visual rendered image	Current state Image, 6D location, Pose Perceived Affordance Hand Affordances

Table 5: Logical device classification.

Once in this format, the mappings between different devices can be examined to allow an appropriate matching between devices (in the real and virtual worlds) to be defined. In Table 4 and 5 four physical and three logical devices are classified.[1]

In each table, the devices are classified by what perceived sense information is available to the user and what is defined in the information structures of the Resources Model. Some of the definitions have been generalised (e.g. hand affordances) to simplify the classification at this level of abstraction.

As a first example we can compare the 2D mouse and the 2D mouse pointer where two main observations can be made. Firstly, the pointer requires for its current state an image and 2D location. The 2D location can be directly mapped to the 2D displacement provided by the physical device but the image must be defined in the artefact. Secondly, there is the mapping between the 2D cursor affordance (actually a combination of physical constraints (e.g. you can not move the pointer outside the screen) and cultural constraints (e.g. how the pointer cursor is used)) and the 2D Arm affordances. The addition of an action-effect resource would provide insight into this mapping in terms of the physical mouse's movement (forward, back, left, right) to the 2D cursor movement (up, down, left, right).

As a second example, consider the 6D tracker (attached to the users hand) compared to the Hand Avatar (a disembodied rendered hand). Although the 6D location is shared by both definitions, there are some components required by the Hand Avatar which are not provided by the physical device, for example, pose and hand affordances. (As before the image is in the artefact. Of course, all missing resources might be able to be moved to the artefact but this would also require augmenting the "perceivables" of the logical device that may be expensive to do.) Also comparing the data glove with the Hand Avatar reveals deficiencies in the mappings. However, combining the 6D tracker with the data glove provides the required direct mapping between the logical and physical devices in terms of the 6D positions and the hand affordances.

In the virtual world, where depiction often stands in for reality, many aspects of physical affordances are denied the designer: the alternatives are constraints and conventions (Norman99). However, if a better match between the physical devices and the logical devices is used in this depiction, then there is a reduced need for artificial (or new) constraints and conventions. This can provide a more intuitive interaction environment without the need for an unnecessarily complex design.

6 Veracity and resources

The level of veracity (or realism) that is required of a VE is an important issue for VE design. As an example of veracity, consider a kitchen in a typical house. This can be

[1] In Tables 4 and 5, *Affordance* = physical affordance and perceived affordance = "the perceivables" (the perceived affordances and behavioural constraints (physical, logical and cultural)).

experienced in the real world ("reality") or a computer based environment (simulated reality). It might be as a 2D plan view, a 3D wireframe model or a head mounted display (HMD) based 3D CAD (Computer Aided Design) environment. Each of the computer-based examples has an increasing level of realism when compared to the real world in the way they are perceived by the user.

In many applications, the realism of the tasks to be performed is one of the prime considerations. Although it should be noted that if the overall required veracity is low, then the use of a VE may be inappropriate. However, if the domain involves tasks that require high levels of realism (in look and feel), for example, training surgeons in a surgical procedure, a VE with a high level of veracity will be required. In this section, we will examine two VE veracity issues. Firstly, how the requirements of tasks can affect the required veracity in a task decomposition and secondly, how methods for action articulation can provide different levels of veracity.

Although a particular sub-task within a task decomposition hierarchy may require high veracity (in terms of presentation and interaction realism), not all the higher level tasks may be required to be defined at the same level of detail. For example, a surgeon wishing to just practise heart surgery may wish to quickly move to manipulating instruments within the heart (ignoring the procedure required to access the heart). However, if the whole heart surgery procedure is to be practised (for example to gauge fatigue factors), then the steps leading to the actual heart manipulation will be required. Therefore, to a certain extent, the task context defines the required veracity level of task content.

Within the Resources Model, the level of veracity has a direct effect on the interaction history resource, both in the system and in the mind of the user. If the interaction history is not required to be detailed for a particular task (e.g. detailed knowledge of past actions are not necessary to perform the task) then reduced veracity can be used at higher levels of the task decomposition. In this example, we view veracity in terms of level of detail that the previous tasks have been represented.

However, if the current task requires that the user have particular information gained through previous steps in the interaction, then the veracity of the information in the interaction history resource requires consideration. The user may need to have enough detailed history information to complete the task in the context of the overall goal. Therefore the veracity requirements of the current task can have an effect on the veracity requirements of the previous tasks.

Perceptual cues within VEs are particularly important for VE interaction (Kaur99) and especially when tasks in the environment are required to have a high veracity. However, if these perceptual cues are not supported with corresponding methods of action articulation then the realism of the environment is compromised (Tromp94). For a task to be performed there may be a minimum required level of veracity (in presentation and interaction). However, as stated above, depending on the context of the task, varying degrees of veracity can be used, thus affecting the type of action articulation required.

This is most easily demonstrated with an example. Table 6 shows four versions of the "Open door" task in four different contexts.

The first row is a description of the task in the real world. This is our "ideal" and has the person approaching a door, reaching out to grip a door handle and pulling the door open with visual and tactile feedback. Next, three VR based descriptions are presented; a desktop 3D game (e.g. DOOM), an architectural or CAD environment and a fire evacuation simulator. Each of these contexts requires different degrees of realism. This can be easily be seen by comparing the action, articulation and effect entries to the "ideal" real world entry.

Task	**Action**	**Articulation**	**Effect**
Open door Real world Using body	Approach door Extend hand Grip handle Pull door open	Walking Arm movement Hand movement Arm movement	Body near door Hand near handle Handle gripped Door open
Open door Desktop VR game Using keyboard	Approach door Open door	VR walking Press space bar	Body near door Door open
Open door VR CAD Using 3D mouse	Approach door Point at door Select door Pull door open	VR walking (flying) Hand movement Press button Arm movement	Body near door Pointer over door Door selected Door open
Open door VR fire evacuation simulator Using data glove	Approach door Extend hand Form ‘grip’ pose Pull door open	VR walking (flying) Arm movement Hand movement Arm movement	Body near door Hand on handle Handle gripped Door open

Table 6: Four versions of the "Open door" task.

In the game environment, the "open door" has been simplified (as not to interfere with the game play). Also the nature of the applications platform has placed physical constraints on this particular articulation. In this case, the "open door" action has been reduced to the two preconditions, "body near door" and "request to open door". The first precondition provides the context of the action and the second the action enablement.

The third entry is a move closer to more detailed approximation of the real world action. For a VR CAD domain, the user is commonly only interested in opening and closing doors and not in the realism of this process. Although this may also be true of the game entry, the game entry only provides a discrete articulation. The door is either open or closed. In this domain the user may wish to have a range of door open positions to guide their room (or house) design process. This can be accomplished without requiring full action articulation to be provided. Hence the use of the "select door" option to target the door object and the hands relative movement to open/close the door.

The last entry is in a domain that requires the highest level of veracity of the three examples. A fire evacuation simulator is an environment where real-time events (the spread of the fire and smoke) and timed actions (walking, door opening/closing) play an important role in success of the system, especially if it is to be used for training exercises. In Table 6 we can see a close mapping between this entry and the real world entry. One interesting difference is in the action-effect columns. The real world has the "hand near handle" while this entry has "hand on handle. This is due to the assumption that full tactile interaction is not available and some type of visual collision detection feedback is providing the perceptual cues to allow the user to grip the virtual object.

In each of these examples, the degree of realism required in the domain has allowed the task to be described at varied levels of veracity. The explicit description of veracity in terms of actions, action articulation and action effect allows design requirements to be identified. For example, in the entries above, if "arm movement" is required for articulation then design issues related to device behaviour need consideration, e.g. the use of a device that allows arm movement to be directly mapped or a device that mimics the behaviour of arm movement.

Also by describing tasks across a domain in this manner, the design process can be guided to resource configurations which more closely match the types of interaction required for all the tasks. In this way, consistency of interaction can be preserved both over an implementation and to its veracity to the real world.

7 Externalising plan resources for navigation in VEs

Much of the work described in this paper has not considered the use of plans as an abstract information structure. This is due to the fact that many of the presented examples have focused on examining individual interaction episodes outside the context of an overall plan. In this section we will investigate how Resources Model plans can be represented in VEs, especially for navigation tasks.

Navigation in VEs is difficult, especially in large virtual worlds (Darken96). Contributing factors to this are the lack of veracity in the presentation and utilisation of navigation metaphors, problems with interaction hardware and cognitive load that is placed on the user. A common navigation metaphor is virtual flying. However, Mine notes that users, particularly novice users, can find this type of metaphor unintuitive and problematic (Mine95). This is compounded by the way virtual flying is commonly articulated. The use of devices, such as 3D mice, requires extensive user training and a high level of user understanding between the mappings of the physical device to the navigation technique. However, the pros and cons of navigation metaphors and their articulation are outside the focus of this section. Here we wish to concentrate on cognitive load (of internalised vs. externalised resources) that is placed on the user and what benefits can be had by resource externalisation.

When navigating, unless the user has a good knowledge of the environment, they are unlikely to build complex plans before starting navigation. Virtual world navigators may wander aimlessly when attempting to find a place for the first time (Darken96). Typically navigation is based on locally obtained information, for example the current state of the environment (externalised in the visual representation) and the goal of the navigation and knowledge of the navigation technique (both internalised in the mind of the user). The user then proceeds with a mixture of goal matching and history-based choice interaction strategies. In goal matching the user decides what to do in a localised way by matching the effects of an action to the current goal and checking if the current system state satisfies the goal (Wright99). For example, setting local sub-goals (e.g. "I should move forward") to see if this gets them closer to the overall goal. With history-based choice, the interaction history is used (with the current possible actions) to determine the next course of action. For example, if the user has navigated forward into a wall, the history ("move forward") can be used to eliminate another "move forward" action and select an alternative action.

Both these strategies (when applied to navigation) only provide localised decision-making and may require the user to extensively explore a VE (building their own mental spatial model) before either reaching their goal (through trial and error) or building internalised plans to reach currently unexplored areas. Depending on the spatial abilities of the user (and the veracity of the environment), users can easily become lost, disoriented and unmotivated.

An alternative to localised action selection is to apply a plan following strategy. Plan following is an interaction strategy that involves the user co-ordinating a pre-computed plan with the history of action so far taken (and, optionally, with the current goal) (Wright99). However, a pre-computed plan is central to the plan following strategy. In the following examples, we present three possible external representations of plans for VE navigation. Each will be examined in terms of their resource configuration and how this effects the ease of navigation.

<u>Example 1: "Follow that line…"</u>

In this example a path representing the navigation plan is shown to the user. This path may be a line on the floor, a length of tread, a footpath, etc. As well as externalising the navigation plan, it is part of the visual representation of the current state, it externalises the navigation goal and it is a partial external representation of the interaction history.

In terms of the interaction history the user can gauge their progress along the line. Also if the user follows the line explicitly, when it ends they have reached their destination. Hence the goal of the navigation has also been externalised.

<u>Example 2: "Use the force … field"</u>

In many VEs, there is no resistance to movement (apart from standard user-object collision detection). However, suppose we impose a forcefield constraint to the users movement. If they move closer to the navigation goal, they can move freely but if they move away from the goal, then the navigation becomes more difficult (e.g. a time taken to distance travelled reduction). The plan can be externalised as a "forcefield tunnel" where the user is penalised (by restricted movement) if they try and move outside the tunnel. If the tunnel also closes behind the user, then a history-based elimination strategy is also being locally enforced by penalising backward movement.

Example 3: "Next location please…"

In some environments, the need for user navigation may be superfluous. If the spatial layout of the environment is not a prime consideration of the current task, then navigation can be simplified by jumping from one location to another. This may require the system to present an externalised view of the current goal so that the user can decide if they wish to "teleport" to the new location. Here the plan is externalised and compressed into a discrete action.

A variation on this could allow the user to select and jump between waypoints (intermediate locations) on the way to the final goal. This is analogous to navigating via a slide show, where each slide in sequence is a step towards the final goal. A similar concept can be seen in the London Underground. Travellers may plan their trips via a conceptual view of waypoints because there is no need to learn actual distances or routes between locations as the plan to move them from location to location is externalised in the set paths of the trains.

Finally, combinations of the three examples above could be used to provide flexibility in users' use of externalised plans in virtual environments. Suppose a virtual tourist is navigating around a complex "real world" environment. They may wish to have a change of scenery or see some of the sights. Pulling up the virtual version of the guidebook, the system presents them with six images of possible navigation goals. With a goal selected, the user might be given the choice of a possible externalised plan before starting a plan following strategy. They might choose to teleport to the new location (discrete external plan execution). Alternatively, they may have a red line displayed that they can follow to new location with an option where the system uses a forcefield tunnel approach to guide their progress while stopping them from getting distracted or lost from the path.

8 Conclusions

One difficult aspect of virtual environment design is that there is a close binding between the user and environment in the final implementation. The aim of many systems is to make the man/machine barrier transparent so that the user feels "immersed" in the environment. To enable this, the VE designer must be aware of the information resources that are available in the man/machine system and how they are distributed.

The use of a distributive cognition approach allows designs to be focused on the user/artefact separation and can identify particular resources that can be moved from the user to the artefact and vice versa. We are using the Resources Model to help us analyse information resources in VE design as a step towards a better understanding of VEs as distributed cognitive systems.

In this paper we have presented four examples of how the Resources Model can be used in the virtual environment design process. Designing interaction episodes can help identify where the scarcity of externalised resources places heavy demands on the user's knowledge. Device classification allows a better match between physical devices and logical devices and allows designers to identify missing device functionality early in the design process. By matching the veracity of tasks to the available resources, the final implementation can provide the level interaction required for the current task. Finally, the externalisation of plans in VEs can aid in reducing the users cognitive load when navigating in the 3D virtual world.

Much of the work in this paper has focused on the abstract information structures of the Resources Model. Current work in the INQUISITIVE project is investigating the effect of Resources Model interaction strategies on VE design. The presence or absence of resources leads users to adopt different strategies. The ease of supporting particular resources in a VE, be it desktop VR or immersion (HMD) VR, and hence the supported strategies, may play an important part in deciding the final implementation platform for a system.

Also the location of resources (external or internal) within an environment effects the ease with which particular strategies are used. Efficient use of interaction strategies may indicate an effective resource configuration. The Resources Model may be able to be used in VE evaluation to provide a retrospective account of user behaviour from a resource based analysis of a completed system.

9 Acknowledgements

The authors wish to thank Bob Fields for his helpful comments on the Resources Model examples. This work was supported in part by the UK EPSRC INQUISITIVE project, Grant GR/L53199.

References

Card90 S. K. Card, J. D. Mackinlay and G. G. Robertson. "The design space of input devices", In *Proceedings of CHI'90*, pages 117-124, 1990.

Darken96 R. P. Darken and J. L. Sibert. "Wayfinding Strategies and Behaviours in Large Virtual Worlds", In *Proceedings of ACM CHI 96*, pages 142-149, 1996.

Dive97 *Distributed interactive virtual environment (DIVE).* Swedish Institute of Computer Science (SICS). Sweden, 1997.

Erickson93 T. Erickson. "Artificial Realities as Data Visualization Environments: Problems and Prospects", In A. Wexelblat, editor, *Virtual Reality: Applications and Explorations*, pages 3-22. Academic Press, 1993.

Fields97 B. Fields, P. Wright and M. Harrison. "Objectives, strategies and resources as design drivers", In S. Howard, J. Hammond and G. Lindgaard, editors, *Human-Computer Interaction INTERACT'97*, pages 164-171. Chapman and Hall, 1997.

Hutchins95 E. Hutchins. *Cognition in the Wild*. MIT Press, 1995.

Kaur99 K. Kaur, N. Maiden, and A. Sutcliffe. "Interacting with virtual environments: an evaluation of a model of interaction", In *Interacting with Computers* **11**(4):403-426. 1999.

Massink99 M. Massink, D. J. Duke, and S. Smith. "Towards Hybrid Interface Specification for Virtual Environments", In D. Duke and A. Puerta, editors, *Design, Specification and Verification of Interactive Systems '99*, Lecture Notes in Computer Science. Springer-Verlag, 1999.

Maverik99 *GNU MAVERIK*, AIG Group. University of Manchester, 1999.

Mine95 M. R. Mine. "Virtual Environment Interaction Techniques". Technical report TR95-018. University of North Carolina, 1995.

Mr95 *MR (Minimal Reality) toolkit*. University of Alberta, Canada, 1995.

Norman99 D. A. Norman. "Affordance, conventions and design", In *Interactions: New visions of Human-Computer Interaction*, **V1.3:** 38-43. 1999.

Sacife96 M. Scaife and Y. Rogers. "External Cognition: how do graphical representations work?" In *International Journal of Human-Computer Studies* **45**: 185-213. 1996.

Smith98 S. Smith, T. Marsh, D. Duke and P. Wright. "Drowning in Immersion", In *Proceedings of UK-VRSIG'98*, Exeter, UK. UK Virtual Reality Special Interest Group, 1998.

Smith99a S. Smith and D. Duke. "Virtual Environments as Hybrid Systems", In *Proceedings of Eurographics UK 17th Annual Conference (EG-UK'99)*, Cambridge, UK, pages 113-128. Eurographics UK Chapter, 1999.

Smith99b S. Smith, D. Duke and M. Massink. "The Hybrid World of Virtual Environments", In P Brunet and R. Scopigno, eds, *Computer Graphics Forum*, **18(**3), pages C297-C307. Blackwell Publishers, 1999.

Tromp94 J. G. Tromp. "Of Metaphors and Magic", Position paper for *ECHT94 Workshop on Spatial Metaphors*. 1994.

Wright96 P. C. Wright, B. Fields and M. D. Harrison. "Distributed information resources: A new approach to interaction modelling", In *ECCE8: European Conference on Cognitive Ergonomics*. Granada, 1996.

Wright99 P. Wright, B. Fields and M. Harrison. "Modelling Human-Computer Interaction as Distributed Cognition", In *Submitted draft*. University of York September, 1999.

Zhang94 J. Zhang and D. Norman. "Representations in distributed cognitive tasks", In *Cognitive Science* **18**, pages 87-122. 1994.