3. Modeling and Displaying Virtual Environments

Chapter 3

Modeling and Displaying Virtual Environments

"Only two things are certain: the universe and human stupidity; and I'm not certain about the universe."

Albert Einstein

In this thesis, not only are we interested in the technical aspects of distributed VE systems - their architecture - but also in the methods used to model VEs. To better understand what we are trying to achieve when we model a VE, our natural environment is examined and ways of defining and classifying VEs are explored. To conclude this abstract examination of environments, a number of modeling processes that may be used to capture the essence of the environment being modeled are discussed.

As the reader knows, we interact with any environment via our senses. The information we gather from these senses is processed by the various perceptual systems in our brain. An effective VE system will generate displays that enable human perception, e.g. visual and auditory, to operate naturally. If the VE displays present the information in a confusing way, then the VE system is not doing the model of the environment justice.

To illustrate this point, the implications of misusing the visual display are discussed - currently the norm rather than the exception. Rectifying the problems with the way in which this display is used has ramifications for VE system design. These technical details are discussed in this chapter as a prelude to the consideration of the more general system requirements presented in the next chapter.

3.1 A New Modeling Paradigm

There are many questions that should be asked when designing a VE. What sort of information should be provided? How should it be structured? How can it be described? These questions face all VE designers, whether the environment is intended for data visualisation, teleoperation or vehicle simulation. In the hope of gaining a better understanding of the task at hand this section examines our natural environment. How we interact with our environment, its important features and its implications on VE design are discussed with the aid of several VE definitions and classification schemes. By analysing how we interact with the real world we can gain insight into how effective virtual worlds may be constructed.

For argument's sake, let us say that (for now) a VE is a synthetic version of our natural environment. Logically, our next question would be "what is *our* environment?" How do we describe the environment in which we live? This is a question that has been given a great deal of thought by many people working in every discipline: Physics, Psychology, Physiology, Philosophy, the Arts, just to name a few. Each of these disciplines offers its own unique view on the subject. Regardless of their definitions, which can be quite different, they are all valid and each has its own place and use. Physics can provide us with information on how the environment is constructed in physical terms of force, mass, energy, etc. Physiology deals with how our body functions within the environment, Philosophy deals with more abstract concepts, whilst Psychology concentrates on the more cerebral activities of our body, including our *perception* of the environment that we are in.

3.1.1 Definition of a Model

Before we tackle the thorny issue of defining a VE, it is useful to give some thought to what we mean by a "model".

> *A model is an* ***implementation***
> *of a* ***representation***
> *of an* ***abstraction***
> *of a* ***thing****.*
>
> Barzel (1992), p27.

The *thing* being modeled is not part of the model, indeed the model is a simplification of the thing, consisting of a subset of the properties that make the thing. This subset is the *abstraction* and can be thought of as the set of ideas that underlie the model. As such, the abstraction is an entity without substance and therefore cannot be manipulated. The *representation* is a complete description of the model and is concrete in the sense that it may be edited, copied, analysed and contains sufficient information to build the model. It is possible to have many representations for any given abstraction and a representation to be shared by multiple abstractions. The execution of the model is the *implementation*, of which there may be many for any given representation, each with their own quirks.

This introduces us to a way of describing models referred to by Barzel as an Abstraction Representation Implementation (ARI) structure. Barzel presents ARI for use in physically-based modeling but it can be used to decompose most types of model. For example, the model of a computer program may be analysed using the ARI scheme: the conceptual specification is its abstraction, the design document is its representation and, naturally, the software itself is the implementation. If we adopt this methodology for the modeling of VEs then we must first find a suitable abstract model. From this we should be able to derive a suitable representation, maybe in the form of a language, and eventually the implementation of a system which can execute our VE.

3.1.2 An Ecological Approach

The nature of the environment and how it shapes the evolution of animals contained within, was a key concern of the eminent psychologist, James J. Gibson. When describing the Gibsonian approach, the key word is *affordance* (Gibson, 1979). The *affordances* of an environment are what it *offers* the animal in terms of action and interaction, what it *provides* or *furnishes*, for good or bad, e.g. a fire can afford warmth but it also has the power to destroy. An important point is that affordances do not reside in the environment, they are the result of interactions between the animal and the environment.

Objects within the environment are classified as being either *attached* or *detached*. In Newtonian physics, all objects in space are detached, but from an alternative perspective it is obvious that some items are attached and cannot be moved without breakage. In order for an object to afford behaviour, it must be both detached and comparable in size to the animal under consideration. Exactly how small or large an object has to be until it does not afford behaviour is unclear, but those objects that are comparable can afford a wide variety of behaviours. Objects can all be said to have properties or qualities, e.g. colour, texture, composition, size, mass, etc. Orthodox psychology asserts that we perceive these objects insofar as we discriminate their properties or qualities, but Gibson suggests that what we perceive when we look at objects are their *affordances*, not their qualities. However, to perceive an affordance is not to classify an object, e.g. a stone is a missile but it can also be a paperweight, part of a wall, etc.

By describing the environment in terms of animals, Gibson rightly makes the point that each animal has its own view of the same environment. Or to put it another way: given an infinitely detailed environment, each animal will extract only that information which it needs. Because different subsets of the environmental properties are being used, the animal's *perception* of the environment (and the objects within in it) will be different. For example, when we look at a tree we may be interested in it as a material for construction or maybe as shelter from the rain. A dog, on the other hand, may be assessing it for more basic needs.

There is a very simple reason for this situation. If you examine an environment in detail it will present properties that are conducive to certain animals and properties that make the environment hostile towards others. Look at any species that survives today and you will see an animal whose perceptual systems have evolved to complement its environment. An animal implies an environment and an environment implies an animal.

If the affordances of a thing are perceived correctly, we say that it looks like what it *is*. However, when evaluating the properties of an object, it is important for us to take a step back and view them in the context of the environment and not just from the human perspective - a task that is easier to state than accomplish. Gibson's ecological framework has already motivated the design of a VE Computer Aided Design system (Smets *et al.*, 1993, 1994). Familiar modeling tools such as hammers and saws are replicated in the VE and afford behaviours found in everyday life, although they are not limited to these functions.

3.1.3 Tools of the Trade

We receive information about the environment through our senses. The limitations of our senses dictate the parameters to our perception of the environment. We cannot decide what an environment is without also examining the capabilities of our own senses.

Sensory Modality	Sensitivity/Resolution
Touch	10-100 micron vibration 1-2 mm spatial resolution
Smell	7 dimensions?
Sight	~400-700 nm in the electromagnetic spectrum 10 minutes of arc at 6 metres
Sound	20 Hz to 20 KHz depending on the intensity ~-10 dB to 120 dB
Taste	4 dimensions: salty, sour, sweet, bitter?

Table 3.1 Common senses and their sensitivity/resolution.

The five commonly accepted sensory modalities taught at primary school level are touch, smell, sight, sound and taste (Table 3.1). However, there are more: interoception, proprioception and exproprioception (Lee, 1978). Proprioception is the ability to sense the position and movement of body parts relative to each other whilst exproprioception is the sense of body position in relation to the environment. Interoceptors indicate the internal state of the body, e.g. hunger, thirst, tiredness, whilst our vestibular system (in our inner ear) provides us with information to help us balance. It has been proposed that taste has four dimensions and arbitrary tastes may be synthesised with combinations of these primaries (Carlson, 1986). Similarly, it is possible that smell may have many dimensions (possibly as many as seven) and so it may also be synthesised. Predictably, the senses commonly stimulated by current VR systems have already been quantified more precisely. With the ability to pick out millimetre detail at 6 metres, it is unsurprising that most people are disappointed with the display technology used in current HMDs.

The resolution of our senses would be a good place to start when determining what information to use to represent our environment and at what accuracy, but it would also be short-sighted. By exclusively adopting the human perspective we will inevitably lose some of the environment's actual fidelity, although it would not be noticed until an unconventional view was attempted. For example, assuming the behaviour of another animal, such as a cat, will involve a different set of environmental properties in order for the participant to interact effectively. Regardless of the practicalities of this, it is important to realise that the senses of a human may be supplemented through various equipment such as infra-red night vision goggles. Robinett (1992) also notes that if sensors can detect phenomena that are imperceptible to human senses, they could be linked to display devices. This would mean that these imperceptible phenomena could be rendered visible, audible, touchable or otherwise perceptible to a human being. In a way, creating a synthetic sense.

If we restrict, for example, the modeling of the surface properties of an object to how things look in the visible spectrum, we will not be able to simulate it correctly when seen through night vision goggles. We may also wish to view the environment from another animal's perspective, e.g. a dog sees in monochrome, not colour, and its hearing is far more sensitive than our own, to name but two differences. Using the knowledge of our sensory abilities to aid the design of human-computer interfaces is essential (Anderson, 1993; Caird and Hancock, 1993; Mon-Williams *et al.*, 1993), but as a guide to modeling the environment it can be shown to be ultimately inadequate.

3.1.4 Virtual Environment Taxonomies

This section presents four different definitions/classification schemes for VEs. Each of them tackle the task at a different level and some are more detailed than others. The major points are presented here and comparisons drawn.

Model Source	Artificial	Real
	Dynamic Model dynamically changes during the participation based on the actions of the participant or other events. Model database changes dynamically.	Direct 1:1 mapping between the space or interface parameters as experienced by the participant.
	Constructed Model is defined a priori as a fixed space and objects. Model database is static.	Sampled Limited spatial or interface parameter resolution.
	Recorded Time recording of the space of interface parameters.	Modified Modified space or interface parameters such as gain frequency response, of time variable.
		Recorded Time recording of the space or interface parameters.

Transparency
(relative contribution between artificial and space components to create the environment)

Figure 3.1 Confection of artificial and real environments.

Time	Space
Direct 1:1 correlation between time in the environment and the participant environment.	Direct (x,y) matching of the participant space and the environment.
Multiple (nt) time modification between participant space and the environment.	Distance (mz) distance scaling of the participant space and the environment.
Fixed (T) fixed time between participant space and the environment.	Scaled I(x, y, z) or (mx, ny, oz) scaling of distance for the spatial dimensions between participant and the environment.
Remapped f(t) functional remapping of time between participant and the environment.	Functional f(x, y, z) functional remapping of distance for the spatial dimensions between participant and the environment.

Figure 3.2 Type of time and space.

3.1.4.1 A Conceptual Virtual Reality Model

Latta and Oberg (1994) have proposed a conceptual VR model which embraces Gibson's work. VR interface technology is viewed as integrating perceptual and muscle systems but it was noted whilst deriving this model that fully integrating these systems would be impossible due to the complexity of the human interface. So the model only examines some perceptual systems, not all. An operational VR system is seen as providing a computer interface to specific human perceptual and muscle systems for the purpose of allowing the participant to perform operations that would not be possible without aid, e.g. a flight simulator. The model's emphasis is placed upon providing an interface to perceptual systems, not on describing what the interface looks like.

The conceptual model consists of a human and a technical view of the VR system. The human view is interested in the physical and psychological issues of stimulating and detecting the actions of the participant, whilst the technical view is concerned with the environment.

> "*The environment provides the stimulus that creates sensation while the individual takes action through movement. The environment ... is the total space, both real and artificial.*"
>
> Latta and Oberg (1994), p25.

The definition and integration of the real and artificial environments is viewed as defining the participatory experience. The mapping of the physical sensors and effectors supports definition of the participant's perception of the environment and their actions on it.

Technical confection and the real environment make the technical view of a VR system. Confecting is the process of preparing or making, especially by combining. Latta and Oberg believe that in VR we are confecting a participatory environment by combining a real environment with an artificial one (Figure 3.1). The technical confection includes a confection model that achieves interface control, defines the artificial environment and mediates between the participant and the real environment. A confection model can support independent models for each perceptual system. It also supports independent models for each muscle system, but the participant's detection of the action is usually correlated with perceptual systems.

Figure 3.2 shows the ways in which space and time may be altered from their natural direct state to modify the experience. Latta and Oberg believe that there is a natural hierarchy in managing and controlling a VR system based on the parameters of the technical confection model. First the mapping of the sensors and effectors supports definition of the participant's perception of the environment and their actions on it. At the next level the model source defines the static and dynamic aspects of the environment. Finally, space and time have equal importance: they are independent of each other but dependent on the first two levels of the confection model.

3.1.4.2 An Experience Taxonomy

Warren Robinett has proposed a tentative taxonomy to classify all varieties of technologically mediated experience (Robinett, 1992). This distinction is offered for "experiences":

- *Natural experience*. Directly perceiving the properties or behaviour of something physically present before the perceiver.

- *Synthetic experience*. Perceiving a representation or simulacrum of something physically real rather than the thing itself.

There are nine dimensions to the taxonomy of a synthetic experience: *causality*, *model source*, *time*, *space*, *superposition*, *display type*, *sensor type*, *action measurement type* and *actuator type*. The first five dimensions deal with the technological aspects of the devices used in the experience, whereas the last four are concerned with the sensor and motor channels used.

Causality refers to the way the VE is experienced, either via a previous recording or transmission, e.g. teleoperation, or totally simulated where actions in the VE have no effect on the real world. The second dimension states that the human user perceives a virtual world that is defined by a possibly changing database called the model. This model can be scanned, constructed, computed and edited. Both time and space may be either aligned, displaced, differ in scale, or be related by a distortion mapping. The time possibilities are 1-to-1 time-scale, accelerated (or retarded) time, frozen time and distorted time. Space may be registered, displaced or expanded (or miniaturised). The last technological dimension, superposition, basically refers to the possibility of merging the VE upon the real world, e.g. using a see-through HMD, or at another extreme, totally isolating the participant within the VE.

Display type and sensor type are the next two dimensions of the classification scheme which enable sensory-enhancing equipment such as infra-red goggles to be described. The last two dimensions present the potential for local input devices to be linked to remote devices in order to effect a change in the remote environment, e.g. teleoperation.

3.1.4.3 Multiple Environment Integration

A proposed definition of a VE, not visibly influenced by Gibson, is:

> "*A multi-dimensional experience which is totally or partly computer generated and can be accepted by the participant as cognitively valid.*"
>
> Jense and Kuijper (1993), p50.

Jense and Kuijper also view a VE as an integration of environments, in this case, three (Figure 3.3):

1. Computer-generated environment.
2. Physically modeled environment.
3. Real environment.

The computer-generated environment is created using a system consisting of sensor, control and actuator subsystems. The physically modeled environment contains objects that are also present in the real environment being simulated, e.g. a replica of an aircraft cockpit may be used to enhance a flight simulator. The real environment is also viewed as an important component in a VE because it can stimulate senses that may be used to add realism to the simulation. However, the difference between the stimuli created by the physically-modeled environment and the real environment is not an easy distinction to make.

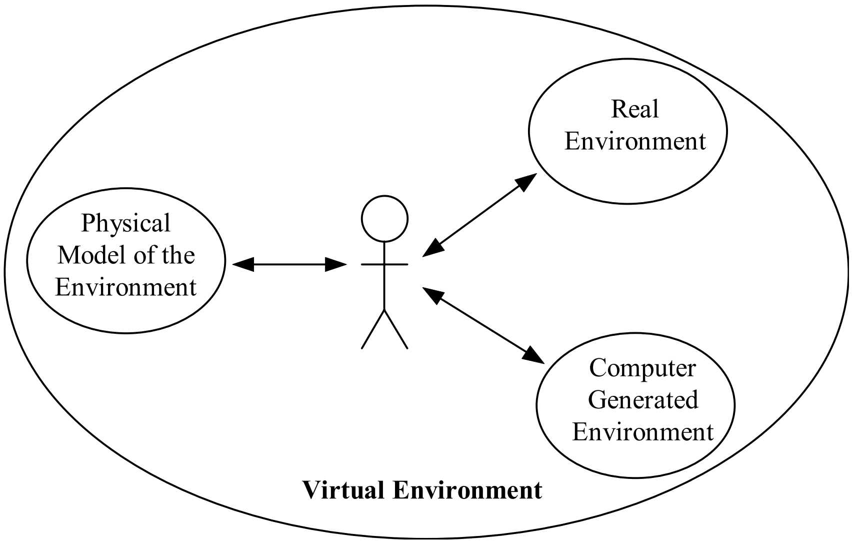

Figure 3.3 Fundamental elements of a Virtual Environment.

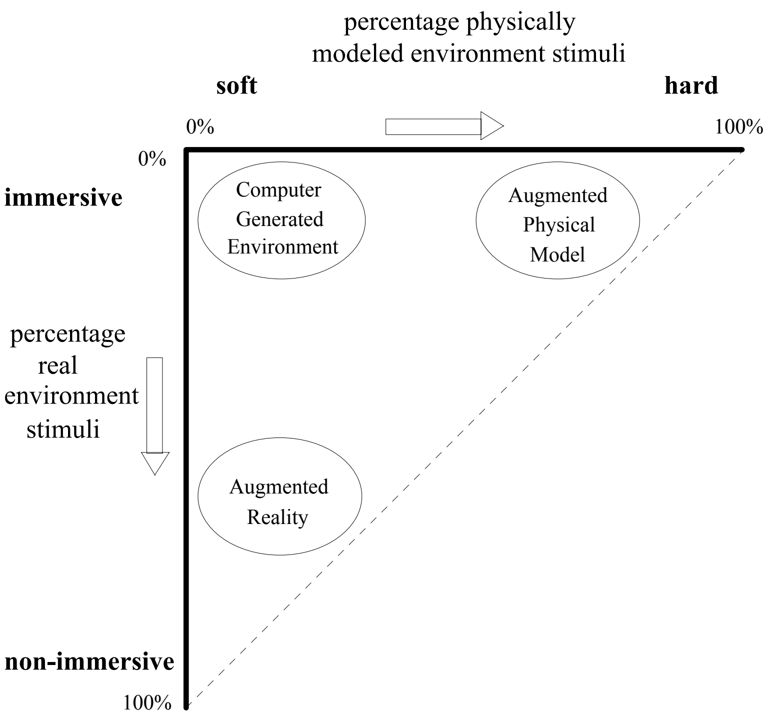

Figure 3.4 A classification scheme for Virtual Environments.

Another classification scheme is proposed based upon the amount of stimuli created by each of these three types of environment (Figure 3.4). A soft VE does not use any physical models to generate stimuli whereas a hard VE uses little else. Immersive VEs cut the participant off from the outside world and non-immersive systems use the real world in the VE. Most systems fall somewhere in between.

3.1.4.4 Content, Geometry and Dynamics

> "*... we can define virtual environments as interactive, virtual image displays enhanced by special processing and by nonvisual display modalities, such as auditory and haptic, to convince users that they are immersed in a synthetic space.*"
>
> Ellis (1994), p17.

A formal definition for the environment, the theatre of human activity, is offered in Ellis (1991), which consists of three parts: *content*, *geometry* and *dynamics*.

The content of the environment is its objects, these are described by *state vectors* which are a description of the *properties* of the objects. *Actors* are similar to objects but may be distinguished by the fact that in addition to properties they have *capacities* to initiate interactions with other objects. The *self* is a distinct actor in the environment which provides a point of view from which the environment may be constructed. Anything outside of the self can be considered the field of action.

The description of the environmental field of action is called the geometry which has *dimensionality*, *metrics* and an *extent*. The dimensionality is the number of independent descriptive terms that are needed to specify the position vector for each element of the environment. Curved or straight lines are established through metrics which are systems of rules that are applied to the position vector. The extent is the range of possible values for the elements of the position vector. Following on from this, the field of action can then be described as the product of all the elements of the position vector over their possible ranges. Kinematic constraints restrict the vast number of possible paths an object may take through the environment.

The dynamics of an environment are the *rules of interaction* among its contents. The transfer of energy or information that occurs during interaction alters the state vectors of the objects involved. All interactions can be reduced to binary interactions which may be ordered based on the ranking of the elements involved. Dynamical rules describe the result of interactions between the environments contents.

3.1.5 An Abstract Model

The definitions of a VE offered by Jense & Kuijper, Latta & Oberg and Robinett all acknowledge the integration of different types of environment, whilst Ellis places more emphasis on the human interface technologies. By evaluating the amount of real and physically modeled stimuli created by each of the three types of environment proposed by Jense and Kuijper, it is possible to classify VEs at a high-level and this provides a basis for comparison. Although a more detailed evaluation would be better undertaken using Latta and Oberg's classification model. Only Robinett and Ellis, however, recognise the importance of perspective on the environment. Even though our senses are limited, it may be desirable to simulate a wider bandwidth of information. This would permit the simulation of sensory-enhancing equipment or for the participant to view the environment in an unconventional way. In other words, each animal within the environment may have a unique view of that environment and hence is concerned with a subset of the environment's total properties. A preferable ecological definition for a VE would therefore be:

> *A totally or partly computer-generated environment that contains enough information so that it may support affordances for different animals simultaneously.*
>
> Where an "animal" is an entity that could be a human with augmented senses, an object with some notion of artificial intelligence, or anything that has a unique perspective on the environment.

In order for a system to be able to support VEs of very different properties, it must have a flexible structure for modeling. Following the ARI decomposition of a model, the chosen abstract model of the VE may be represented in many different ways, each of which may have strengths and weaknesses. Each representation can also be implemented using many different methods, each having good and bad points. However, underlying all the possible implementations and representations should be a sound abstract model.

The model presented by Ellis is quite detailed and uses physics-based concepts to the point at which it could be confusing, at best, and restrictive, at worst, when considering a VE that does not behave according to natural physical laws. Ideally, the abstract model should provide a simple and flexible basis of representing any type of environment. The author believes that such a model exists in the basic structure of our universe and it is the model that should be used. A plausible description of this structure is:

> A *Universe* contains all things that exist. These things may be described as *Entities*. An *Entity* consists of one or more *Properties*. An *Entity* may or may not interact with other *Entities* as dictated by *Universal Laws*. A *Universal Law* is an equation of constraint expressed using *Properties*, *Universal Constants* and other *Universal Laws*. A *Universal Constant* is a quantity that does not change throughout the whole *Universe*.

3.1.6 Representations

The use of terminology in the abstract model is meant to reflect its origins and not its possible applications, fortunately it fits quite well in the context of describing a VE. This model is, in fact, a very basic description of any form of structured data. A universe might be compared to a database: an entity is equivalent to a record, the properties are the record's fields, the universal laws correspond to the relationships between records and so on.

Given that we have established a suitable abstract model for our VE, the next order of business is to find a suitable representation, something we can edit, manipulate and generally play with until we are happy that we have a description that embodies our ideas. In essence, a specialised data description language. This task is undertaken in chapter 4 but before language design is examined, we should first consider modeling methods.

An entity modeled using the abstract model detailed above may have many different representations. There may be a visual representation, an aural representation, tactile, thermal, etc. Each of these is interested in a number of properties, some are shared between them and often some are unique to the representation. They are all governed by a subset of the total universal laws and are applicable to a subset, if not all, of the entities in the Universe.

One possible solution would be to model these representations independently, but this can introduce a great deal of data redundancy. For example, the physical appearance of an entity would only seem to be of interest if you are building a visual model. However, a tactile model is also heavily based on the geometry of the entity and, of course, how the entity distorts sound is based on geometry as well as other factors (Astheimer, 1993). If the shape of the entity changes then the relevant properties in the other models would also have to be changed. A shared structure of information would therefore seem appropriate, at least until design decisions for the implementation of these models need to be taken and then we are faced with the time old battle of distribution versus replication.

It is at this point that we should also consider the design process. Without doubt modeling, whether it is geometrical or mathematical, can be as time consuming as developing the code to execute it, if not more so.

3.2 The Modeling Process

How the information in the model is organised and shaped into the final form is not just dictated by the thing being modeled, but how the model is derived. This section takes a cursory glance at the possible approaches to actually building a model and their effect on the design process. Consideration of these factors aids the design of the modeling language (representation) and the supporting system.

There would appear to be three levels of "reality" (for lack of a better term) that can be created:

- An *observer-oriented* reality would provide adequate simulation of the inputs and outputs required by a human at the required accuracy.
- An *environment-oriented* reality would provide adequate simulation of all inputs and outputs affecting the environment[1] at the required accuracy.
- A *universe-oriented* reality would provide adequate simulation of all inputs and outputs at the highest possible accuracy.

All current systems cater (in one way or another) for the first category, an observer-oriented reality. Most image generators only model the attributes of a surface which are acted upon by visible light. Few give consideration to the rest of the electromagnetic spectrum, e.g. ultraviolet, infra-red, radio etc., because it is not generally required. In the same way, acoustic systems only deal with the range of frequencies that we can hear, even though many others affect us, e.g. ultrasound. The technology for the simulation of stimuli for smell, taste and

[1]In this context *environment* means the volume of entities surrounding the participant. The size of the volume is arbitrary.

touch are only just starting to be developed but clearly an entity's complete set (or subset) of properties must be modeled to permit their use. These observer-oriented systems also fail to easily accommodate simulation of things that do not directly affect us but we may wish to visualise, e.g. the path of radio waves, infra-red light, and so on.

A universe-oriented reality is the ultimate goal and would model everything in fine detail and without exception. In this context, "universe" is intended to mean the thing that is being modeled, in its entirety. Of course, it is possible that the amount of processing power and storage required to simulate the universe would exceed its size in the first instance! Nevertheless, it should be considered as one of the ultimate goals of a VE system, however vain.

The next best thing would be an environment-oriented reality where the microcosm would possess a subset of the properties of the universe. These would be simulated to a high enough level of accuracy to allow their examination and a more accurate and realistic simulation of the participant's environment. How big the environment should be is a good question. Probably any volume that does not encompass the universe could be modeled in this way.

3.2.1 Model Construction

Some attempts have been made to provide higher-level modeling systems (Hemmje & Strohmer, 1993; Luciana *et al.*, 1991; to name a couple), but these still concentrate on a particular type of information or specific application and are not applicable to the general task of modeling a VE. There would seem to be two basic approaches:

1. Take a very general, flexible and computationally expensive model and simplify/remove the parts that are not relevant to the case in hand.

2. Take a skeleton model and then build on it, successively specialising and tweaking.

Both of these methodologies can be seen to use a hierarchical approach in different ways. Using the first method, the designer is given the most complicated model that can be described and then they selectively remove/simplify the parts that are not relevant for the intended simulation. Each branch of the tree would therefore represent a progressively simple subset of the general model. Method 2 does just the reverse and could be likened to the object-oriented language feature of inheritance. Take a simple abstract class that provides the basic structure and fabric of a VE and then derive classes from it that provide it with some "flesh". Each new derived class would increase the realism of the simulation and also its computational complexity. Each of these approaches is valid and may be compared to the programming design methodologies of *bottom-up* and *top-down* design respectively.

3.2.1.1 Methodology Choice

However, method 1 requires a lot more initial work because a great deal of consideration needs to be given to all of the simulation's goals and requirements. This is a potentially impossible task and, if anything, it will be limiting. Its advantage is that little or no work needs to be done to the model to get a fully working simulation running. Unfortunately, the same cannot be said for the entity descriptions themselves, each one must have all of its parameters meticulously evaluated and initial values found.

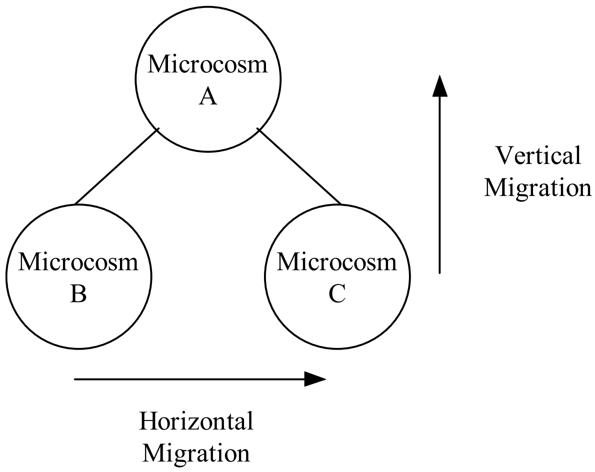

Figure 3.5 Universe hierarchy tree showing possible entity migration paths.

Method 2 requires that a small extensible structure is derived and represented and thus provides the most flexibility. Its disadvantage is that the model's representation has yet to be created which, depending on the simulation, may take some time. The main advantage is that with smaller models there will be less preparation needed for the entities - only those parameters needed by the model will be evaluated. The universe could be represented as being composed of sub-universes or microcosms, each of which has its own laws to govern. Entities in each of the microcosms will possess enough properties such that the microcosm's laws may determine their behaviour. But what would happen if an entity from one microcosm would wish to move into another?

3.2.1.2 Entity Migration

The issue of entity migration really only exists in the second modeling paradigm. With method 1, migration would just mean using a different subset of the entity's properties and would require little or no intervention on the part of the designer. However, using method 2 the microcosm that the entities immigrate to must have sufficient laws to govern them correctly. If it is to do the same job as the emigrated microcosm then it must possess its laws and properties, this would mean that the emigrated microcosm is a specialisation of the higher level. This sort of migration may be thought of as *vertical*, up the inheritance tree (Figure 3.5).

It is equally likely that an entity will want to use *horizontal* migration, i.e. moving from one microcosm to another, each with a common ancestor. The implication of this action is that some of the entity's properties will be shared, some will be left behind and others will be gained when entering the new microcosm. One logical course of action is to assign default values to these new parameters, although in practice this is unlikely to be a very satisfactory solution. Unfortunately, without some insight into the "purpose" of the entity, little else can be done automatically.

3.2.1.3 Modeling Process Summary

It would therefore seem sensible to use method 2 to develop VEs because it requires less initial work and presents a clear structure to the designer. At a reduced level it would be possible to simulate method 1 by redefining inherited laws, etc., to be simpler. Program design is often a combination of both bottom-up and top-down so it would seem reasonable to expect a similar approach to VE design.

3.2.2 The Design Process

All of the systems reviewed in the previous chapter treated the modeling process as an independent task that is performed initially, the result of which is executed. Some systems permit minor modifications to the model to be made at run-time, but this is usually limited to changing the values of selected properties, e.g. entity colour. An entity may leave one VE and enter another through migration but major changes to the entity itself or its environment are not possible.

Such changes may be the addition of new entity properties, the alteration of a law governing those properties, changing the value of a constant, etc. The ability to change the VE at run-time has several advantages:

1. *Development*. By integrating the modeling and execution phases a prototype model can be refined and extended into the finished product without stopping the simulation. Whereas existing systems require some description to be written in a language, compiled/interpreted and then executed, integrated development tools would remove this distinction.

2. *Experimentation*. A better development environment will encourage experimentation in the form that the VE takes. This is, of course, currently possible but the time cycle is large enough to become frustrating. A friendly modeling system increases the likelihood of better VEs and, hopefully, better designers.

3. *Evolution*. The ability to modify the VE need not remain in the hands of the designer. On a restricted level it could be given to the participants in the VE or, more interestingly, to the entities themselves. This reflects an animal's ability to influence its environment, especially true in the case of humans.

3.3 Real-time Virtual Environment Displays

So far, this chapter has examined the somewhat abstract topic of VE modeling. Consideration has been given to our natural environment in the hope that it will add some insight into what we are trying to achieve when modeling a VE. A good VE should be intuitive to use; in other words the participants should have no trouble navigating around the environment, interacting with it, and completing any task that they set out to achieve. However, a sound VE model in itself will not achieve these goals. Unless interaction is effortless (or "natural") then even the most detailed model, built using the most advanced techniques, will fail to deliver the experience intended by the designer. This quality assessment is made via our senses and perceptual systems. If our perceptual systems are working normally then our energies will be expended on the task at hand. However, if we are fed information that disrupts the natural processes of our perceptual systems then we will either become aware of this problem or our performance will suffer. Therefore it is just as important how the environment is *displayed*, as the type of information contained within it. To understand the potential problems with current VE displays we must first establish the cause.

The purpose of a display is to take raw information from the environment, process it, interpret its meaning, and then present it in a form that enables the viewer to extract some meaning. A suitable practical example is that of a visual display which is driven by a CIG - although aural or tactile would also make good examples. A CIG must process the geometrical information in the model, including lighting, surface texturing, etc. The more information it processes, the longer it takes to complete the rendering. If this display is presenting the participant's view on the VE then the time taken to render the view may well depend upon where the participant is looking. Since the viewer will make decisions based upon the information the displays show them, e.g. what they see, then it is important that things appear where they should, when they should. Unfortunately the time between requesting a rendering of a new view and actually seeing it can be relatively large. The same statement can be applied to all types of displays, each of which may perform at different rates. This is not a situation which we have to tolerate when interacting with our natural environment. Consequently, at the very least, the viewer is presented with incorrect information for any given moment in time and, at worst, interaction with the VE is impossible.

The remainder of this chapter examines in more detail why the update rate of VE displays should be constant. In order to present the two possible solutions to this problem it is necessary to consider the workings of a VE system in a little more detail than before. This discussion is a precursor to the detailed system design described in the next chapter and clarifies one of the primary system requirements.

3.3.1 Problems with Variable-Rate Systems

In this section we present an example of the effects that a variable update rate has on interactivity. This is quantified by the application of a visual perception theory. The other benefits of a constant update rate are also discussed.

3.3.1.1 Display Artifacts

Consider a virtual ball moving straight towards you at a constant velocity of 1 m/s. It starts its journey 10 metres from you and you are attempting to catch it. Let us assume that a simulation of this will use a typical variable-rate CIG and a monitor (showing the catchers view) with a refresh rate of 60 Hz. When the ball is in the distance and hence quite small, the CIG manages to generate a new frame 30 times a second. This means that every 2 monitor refreshes a new picture will appear.

If the CIG maintains this frame rate then the velocity of the ball will indeed be constant. However, if the CIG should manage to complete its work within a 60th of a second then the ball's velocity will appear to have doubled to 2 m/s! On the other (more likely) hand, if the CIG's workload takes longer than 33.3 ms to complete and hence only produces a new frame every 3 monitor refreshes, then the velocity of the ball will appear to reduce by 1/3 to 0.66 m/s.

If the frame rate was to go up or down each time an image was being rendered[2] then catching the ball will be made more difficult. In this case we are likely to see a drop in update rate because as the ball comes towards us, it expands. If the ball was textured and the background blank, this would mean that there are more pixels to fill and hence more work to do. Certainly, we are not seeing what the designer of this simulation wanted us to see.

Another more practical example is that of a driving simulator. Given the task of following a vehicle and ensuring that you do not crash into it would be made difficult if the vehicle would seemingly slow down and speed up quite uncharacteristically.

3.3.1.2 Judging Time-to-contact

Lee (1976) presented the Tau theory which suggests that our ability to judge our time to contact with a given target is based upon the rate of expansion of the target on the retina. This may be applied to our ability to catch balls as well as how we control our deceleration, among other tasks (Lee, 1993).

The time-to-contact (TTC) of the virtual ball may be expressed as:

```
TTC = Distance / Velocity
```

Figure 3.6a shows the TTC assuming that we maintain a constant update rate of 30 Hz which gives us a perceived constant velocity of 1 m/s. The impact of a variable update rate is shown in Figure 3.6b. Each time the update rate changes so does the TTC, forcing the catcher to continuously readjust. In this case, the catcher will probably catch the ball because the update rate has slowed down so much that the perceived velocity of the ball at 5 Hz is 0.16 m/s, making the task trivial. They are unlikely, however, to be using TTC information to help them catch.

3.3.1.3 Affects on Latency

If the time between sampling input devices and updating the display is too long it can contribute to simulator sickness (Pausch *et al.*, 1992). Just how long is too long is not clear, additionally it is not clear whether the systems used provided a constant or variable display update rate. There is evidence to suggest that humans can adapt to a constant degree of lag (providing that it is not too great) after a reasonable period of time, but how effective the interaction is depends on the task being performed. If the lag varies then adaptation is less likely and it is possible that this will add to simulator sickness.

---

[2] The word "render" is used in this thesis to embrace both of the classical geometrical and rendering stages used to produce an image.

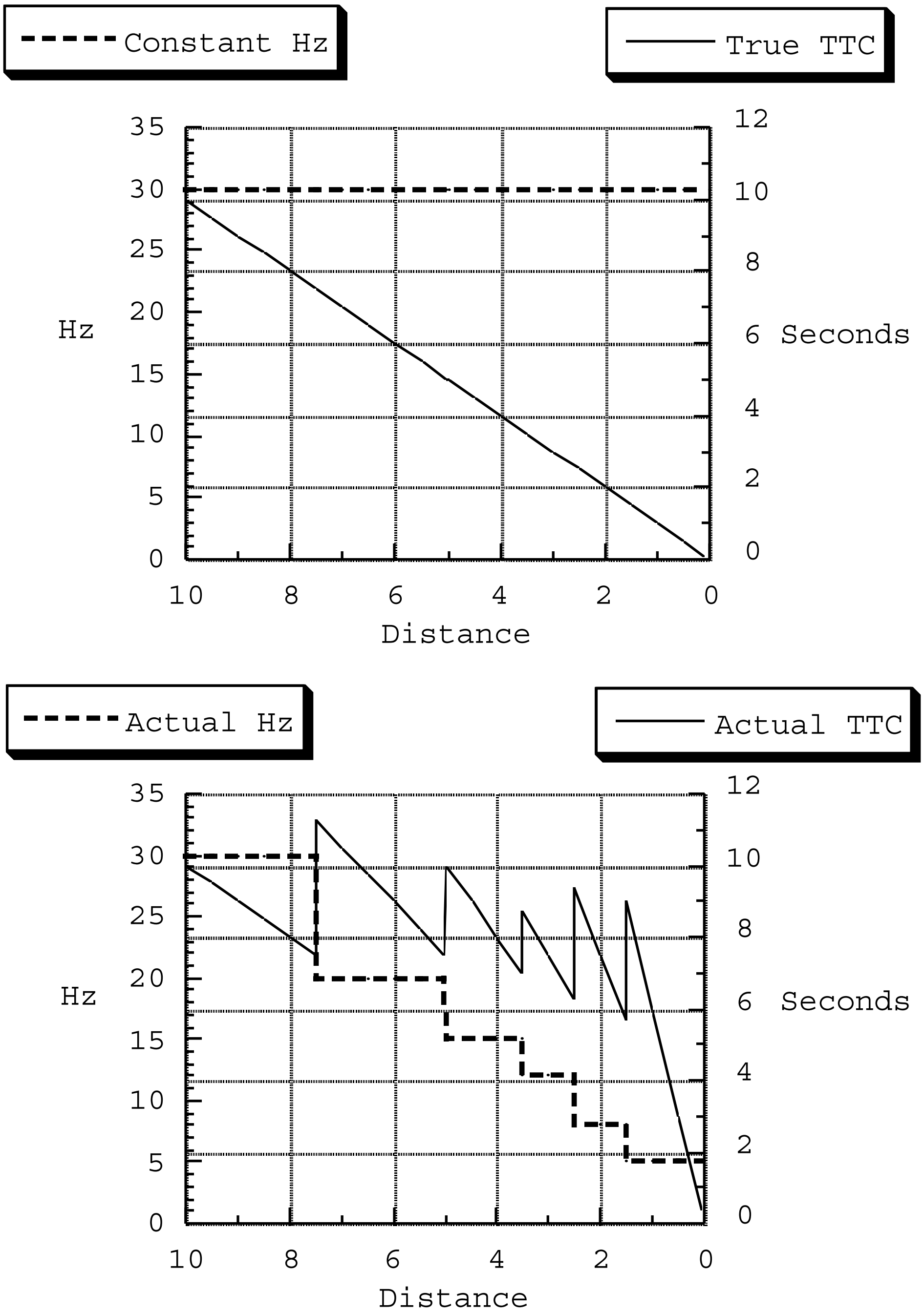

Figure 3.6 The effect of update rate on time-to-contact (TTC).
a) with a constant update rate TTC decreases correctly at a fixed rate (top); b) a variable update rate causes a continuous readjustment of TTC (bottom).

3.3.1.4 Predictive Techniques

There are methods for reducing the impact of lag on the participant. Kalman filters can be used to compensate for the effects of lags within the system (Friedman *et al.*, 1992; Liang *et al.*, 1991; Dunnett *et al.*, 1995). Such filters have been used to predict the movement of 6 d.o.f. sensors attached to parts of the body, e.g. head and hands. In the case of Head-Mounted Displays this means that the CIG can be asked to generate an image of the participant's viewpoint a short while in the future such that the image reaches the display at the right time. The effectiveness of these filters relies on the constancy of the lag and hence the update rate, without it the results of the filtering would be meaningless.

If the progression of time happens at a known rate it is also possible to ensure that entities within the VE appear at their correct positions when the image is eventually displayed. This is especially useful when trying to compensate for the single frame delay introduced in double-buffered CIG systems.

3.3.1.5 External Device Synchronisation

It may also be desirable to synchronise the VE display with an external data capture system. An example of such a device is the Ober/2™ infra-red eye-tracking system (Permobil Meditech AB, Sweden). A lot of effort has been expended by the manufacturers to ensure that a fast, constant sample rate is achieved, to such an extent that the host machine is configured solely for the purpose of controlling the eye-tracker. Sample rates over 1000 Hz may be achieved although 180 Hz is sufficient for tasks monitoring basic eye movements (Permobil, 1993). In order to determine where the participant was looking within the VE display requires the meshing of two data sets, each with a different sample rate. Whilst, on a variable-rate system, it would be possible to record the update rate and then fit the eye-tracker data set to this, the result would be an uneven spread of data points over time. With a constant update rate system, the eye-tracker rate can be set at a multiple of the update rate which makes meshing much easier and produces a consistent number of data points per second.

3.3.2 The Variable-Rate Paradigm

A typical simulation processing cycle is:

1. Sample input devices.
2. Perform dynamics calculations.
3. Update output devices.

The VE system may consist of many components, both software and hardware. With each component comes a response time, a best and worst case for receiving data, processing it and outputting a result. Exactly where the bottleneck in the system is depends on the nature of the VE or application. Typically the bottleneck is the CIG. This is especially true in low-end systems where the CIG is more (or totally) dependent on the host processor to complete its task. In this case, image generation often has to be scheduled along with input/output device handling and the dynamics calculations. It is also quite typical for the workload of each component to vary. This is especially the case in the CIG where scene complexity may vary drastically (Airey *et al.*, 1990).

3.3.3 The Fixed-Rate Paradigm

In order to provide a constant update rate there are two possible approaches:

1. Derive some predictive algorithms that will enable us to determine the workload of each component and thus the system as a whole.

2. Restrict the update rate to the worst-case.

Both these methods are working to complete the 3 steps in our simulation cycle before a given deadline. Once this deadline has been met it is recycled and used again for the next VE display update.

If we adopt the first approach then we may use the knowledge of each component's performance to degrade the services it offers such that the deadline for each component will be met. Alternatively, we can demand less of the system such that, even in the worst-case, it always meets its deadline. This inevitably means using some components at less than optimum performance. Both of these techniques will now be discussed in further detail.

3.3.3.1 Service Degradation

This technique requires a scheduler to determine acceptable time-frames within which each component in the system must complete its calculations. The addition of a scheduler brings us one step closer to a real-time system. Failure to meet a deadline will have different consequences depending on the application. A visualisation may be content with simply providing a lower update rate (albeit constant) whereas a highly interactive application may treat failure to meet the deadline as a fatal condition.

It should be noted that some systems have decoupled the rate at which component services are requested and the update rate of the CIG (Shaw *et al.*, 1992; Wloka, 1993; UVa, 1995). Therefore the simulation may progress as fast as possible, while the CIG generates images as fast as it can.

However, CIG performance can still benefit from service degradation. Holloway (1992) draws as much of the visual scene as possible whilst still attempting to meet the deadline. To achieve this, the Viper system uses a special feature in the Pixel-Planes Programmers Hierarchical Interactive Graphics Standard (PHIGS) implementation which allows traversal of a particular part of the database hierarchy to be terminated based on a conditional check of a global flag. In addition, visual objects[3] were given either a high or low priority. High priority objects were always drawn and low priority objects only if time allowed. There is no guarantee that the image will be rendered within the allotted time since Viper uses successive estimates to decide whether it has enough time to render any more and is at the mercy of the underlying operating system (OS).

[3] The visual component of an entity.

Wloka (1993) proposes a system for time-critical graphics which uses knowledge of the dynamics behaviour of the simulation and a modified graphics database model combined with a scheduler to implement this technique.

As Wloka notes, few CIGs support service degradation techniques. The nearest facility that most provide is Level Of Detail (LOD) which attempts to reduce workload by automatically substituting models of different visual complexity based on distance or screen pixel coverage (Reddy, 1995). SGI's IRIS Performer™ goes one step further by providing a mechanism known as dynamic LOD scaling. This provides enough basic information for Performer to decide which combination of LOD models will complete rendering within a certain amount of time (SGI, 1995). The other work done in this area is at the application level as opposed to adding functionality to the CIG. Airey *et al.* use LOD along with other pre-processing techniques to support an adaptive refinement system that trades image realism for speed. Funkhouser and Séquin (1993) use *cost* and *benefit* heuristics to determine which LOD model should be used. The cost of an object is the time it takes to render an object with a given LOD using a certain rendering algorithm, whilst the benefit is an estimate of the contribution of the model to human perception. Encouraging results are obtained using this approach, however, even this technique is not sufficient to cope with extreme cases such as changing the view from looking at the sky to looking at a fully textured model of a town.

3.3.3.2 Worst-case Operation

Establishing what the worst-case is for a given VE can be accomplished by either working out by hand the worst performance of each component or by "exercising" the VE over a period of time. The latter method is very convenient and relatively effortless to perform, however its effectiveness is dependent on exercising the parts of the system that will present the worst performance, either on their own or combined with other components.

A major advantage of this approach is that it may be used on systems without real-time extensions and although scheduling still plays an important part, it is done on a decidedly pessimistic basis. The price paid for this type of predictability is the under utilisation of the available services, which is sometimes quite extreme if there is a large bottleneck in the system.

3.3.3.3 Implementation

Regardless of which method is chosen, the control of the CIG is the same and the possible scheduling options limited. A prototype implementation of the ideas presented here is given in chapter 5.

3.3.4 Conclusions

Producing successive displays of a VE at a variable rate can be shown to cause interactivity to suffer. The sense of presence in VEs is another area where variable rates may have an effect. In the study performed by Barfield and Hendrix (1995), five different update rates were used to examine the sense of presence. Efforts were made to ensure a constant update rate but it would also be interesting to see the effect that a variable update rate has on presence - which is currently a far more realistic situation.

Increasingly, other complex standalone hardware (such as eye-trackers) are being incorporated into VE systems. Without a common time-frame, attempts to synchronise this equipment with a VE system can produce anything from erroneous to useless results.

Whilst a constant update rate permits object positions to be calculated into the future, predicting the actions of a human interacting in the VE is another matter. Estimation of the participant's head and possibly hand movements may be accomplished using Kalman filtering, but there is no way of anticipating what they will *do* next. Because of this there will always be a latency between human action and displayed reaction with an order of one or two updates. However, it is surely better to base a judgement on a VE whose state is correct for that moment in time, than to base judgements on out-of-date information.

3.4 Summary

Attempting to define a VE in one sentence is next to impossible but its most important features can be expressed concisely. The general consensus is that a VE is actually far from virtual. It is a combination of our natural, physical environment and the computer-generated environment that is presented to the user through a wide variety of displays. It would seem that Virtual Environment is not a suitable term for describing such a phenomenon, Artificial Environment or Synthetic Environment would probably be more appropriate. However, almost all of the literature talks in terms of VEs so it would seem sensible to stick with the most commonly used term.

After establishing exactly what a model is in the general sense of the word, the search for an abstract model began. It is clear that an environment is perceived differently depending on the viewer's perspective. This change in perspective may be due to the augmentation of our natural senses or even a change in species. Additionally, in order to enable unconventional input devices and displays, e.g. tactile, to function correctly, it is necessary to model more information than usual. Consequently, a method of modeling the diverse information present in the environment must be found.

As a prologue to finding a suitable representation for the proposed abstract model (presented in chapter 4), the modeling and design processes were considered. A hierarchical approach using inheritance to extend and specialise successive models was the favoured modeling methodology. This increases the flexibility available to the designer and, with the correct system support, will hopefully aid them in the production of better VEs.

After dealing with rather esoteric issues, a more down to earth problem was discussed. Variable update rates can destroy the visual illusion because this effect is not experienced by us when interacting with our natural environment. The synchronisation of audio with visuals can also fall victim to such a situation (as will most displays). Correct interpretation of data from input devices can also suffer in systems that have a variable duration between device sampling and display output. The technical details of a constant-rate visual display are presented in chapter 5.