CHAPTER III

DESIGN AND EVALUATION CONCEPTS

We wish to perform our design and evaluation of interaction techniques for immersive virtual environments in a principled, systematic fashion (see e.g. Price, Baecker, and Small, 1993, Plaisant, Carr, and Shneiderman, 1995). Formal frameworks provide us not only with a greater understanding of the advantages and disadvantages of current techniques, but also with better opportunities to create robust and well-performing new techniques, based on the knowledge gained through evaluation. Therefore, this research will follow several important design and evaluation concepts, elucidated in the following sections.

3.1 Taxonomy and Categorization

The first step in creating a formal framework for design and evaluation is to establish a *taxonomy* of interaction techniques for each of the universal interaction tasks (note on the word 'taxonomy': we will employ both of its accepted meanings: "the science of classification," and "a specific classification"). Taxonomies partition the tasks into separable subtasks, each of which represents a decision that must be made by the designer of a technique. In this sense, a taxonomy is the product of a careful task analysis. For each of the lowest level subtasks, technique components (parts of an interaction technique that complete that subtask) may be listed. Figure 2.1 presents a simple generalized taxonomy, including two levels of subtasks, and several technique components. Taxonomies for the tasks of travel (sections 4.3.1 and 4.6.1) and selection/manipulation (section 5.4.1) are presented later in the thesis.

The taxonomies must come from a deep and thorough understanding of the interaction task and the techniques that have been proposed for it. Therefore, some initial informal evaluation of techniques and/or design of new techniques for the task is almost always required before a useful taxonomy can be constructed (section 3.4).

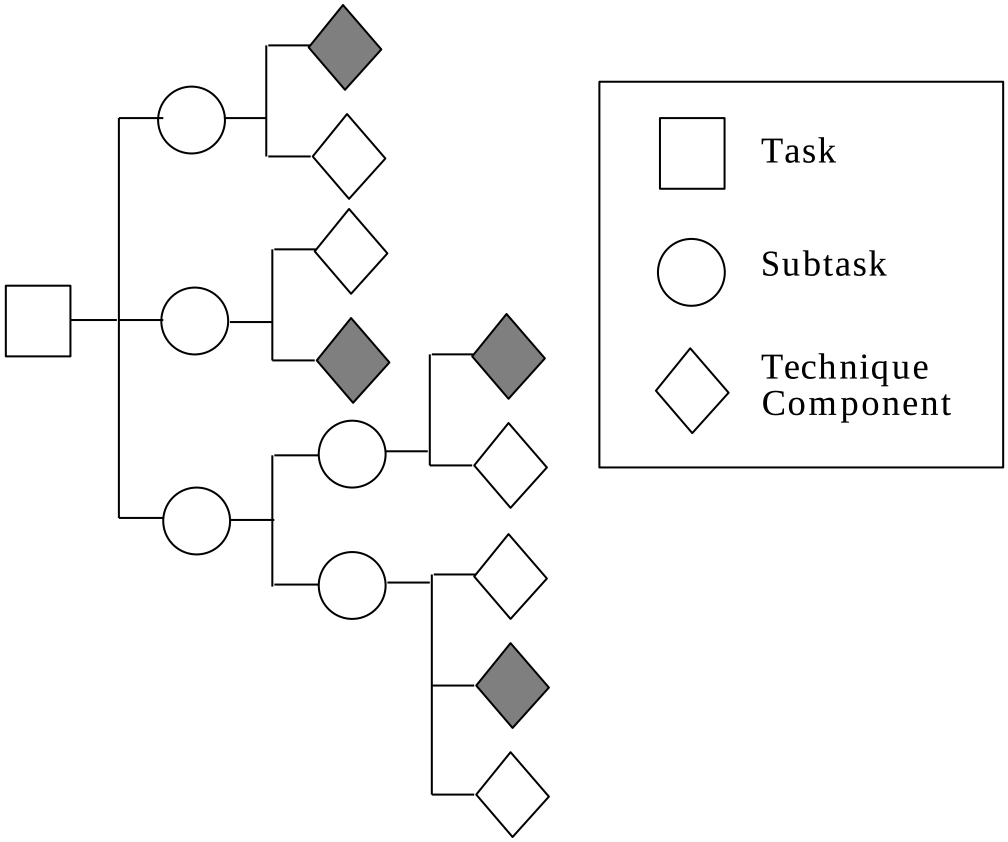

Figure 2.1 General Taxonomy Format

Let us consider a simple example. Suppose the interaction task is to change the color of a virtual object (of course, this task could also be considered as a combination of universal interaction tasks: select an object, select a color, and give the “change color” command). A taxonomy for this task would include several task components. Selecting an object whose color is to change, choosing the color, and applying the color are components which are directly task-related. On the other hand, we might also include components such as the color model used or the feedback given to the user, which would not be applicable for this task in the physical world, but which are important considerations for an IT.

Ideally, the taxonomies we establish for the universal tasks need to be complete and general. Any IT that can be conceived for the task should fit within the taxonomy, and should not contain components that are not addressed by the taxonomy. Thus, the components will necessarily be abstract. The taxonomy will also include several possible choices for each of the components, but we do not necessarily expect that each possible choice will be included. For example, in the object coloring task, a taxonomy might list touching the virtual object, giving a voice command, or choosing an item in a menu as choices for the color application component. However, this does not preclude a technique which applies the color by some other means, such as pointing at the object.

Moreover, we do not claim that any given taxonomy represents the “correct” partitioning of the task. Different users have different conceptions of the subtasks that are carried out to complete a task. Rather, we see our taxonomies as practical tools that we use as a framework for design and evaluation (see below). Therefore, we are concerned only with the utility of a taxonomy for these tasks, and not its “correctness.” In fact, we discuss two possible taxonomies for the task of travel, both of which have been useful in determining different aspects of performance. Rules and guidelines have been set forth for creating proper taxonomies (Fleishman & Quaintance, 1984), but we felt that the structure of these taxonomies did not lend itself as well to design and evaluation as the simple task analysis.

One way to verify the generality of the taxonomies we create is through the process of *categorization*. If existing techniques for the task fit well into the taxonomy, we can be more sure of its completeness. Categorization also serves as an aid to evaluation of techniques. Fitting technique components into a taxonomy makes explicit their fundamental differences, and we can determine the effect of choices in a more fine-grained manner. Returning to our example, we might perform an experiment comparing many different techniques for coloring virtual objects. Without categorization, the only conclusions we could draw would be that certain techniques were better than others. Using categorization, however, we might find that the choice of object selection techniques had little effect on performance, and that color application was the most important component in determining overall task time.

3.2 Guided Design

Taxonomy and categorization are good ways to understand the low-level makeup of ITs, and to formalize the differences between them, but once they are in place, they can also be used in the design process. We can think of a taxonomy not only as a characterization, but also as a design space. In other words, a taxonomy informs or guides the design of new ITs for the task, rather than relying on a sudden burst of insight (hypothesis 1).

Since a taxonomy breaks the task down into separable subtasks, we can consider a wide range of designs quite quickly, simply by trying different combinations of components for each of the subtasks. For example, the shaded components in figure 2.1 represent a possible complete interaction technique. There is no guarantee that a given combination will make sense as a complete IT, but the systematic nature of the taxonomy makes it easy to generate designs and to reject inappropriate combinations.

Categorization may also lead to new design ideas. Placing existing techniques into a design space allows us to see the “holes” that are left behind – combinations of components that have not yet been attempted. One or more of the holes may contain a novel, useful technique for the task at hand. This process can be extremely useful when the number of subtasks is small enough and the choices for each of the subtasks are clear enough to allow a graphical representation of the design space, as this makes the untried designs quite clear (Card, Mackinlay, and Robertson, 1990).

3.3 Performance Measures

The overall goal of this research is to obtain information about human performance in common VE interaction tasks – but what is performance? As computer scientists, we tend to focus almost exclusively on speed, or time for task completion. Speed is easy to measure, is a quantitative determination, and is almost always the primary consideration when evaluating a new processor design, peripheral, or algorithm. Clearly, efficiency is important in the evaluation of ITs as well, but we feel there are also many other response variables to be considered.

Another performance measure that might be important is accuracy, which is similar to speed in that it is simple to measure and is quantitative. But in human-computer interaction, we also want to consider more abstract performance values, such as ease of use, ease of learning, and user comfort. For virtual environments in particular, presence might be a valuable measure. The choice of interaction technique could conceivably affect all of these, and they should not be discounted.

We should remember that the reason we wish to find good ITs is so that our applications will be more usable, and that VE applications have many different requirements. In many applications, speed and accuracy are not the main concerns, and therefore these should not always be the only response variables in our evaluations.

Also, more than any other computing paradigm, virtual environments involve the user – his senses and body – in the task. Thus, it is essential that we focus on user-centric performance measures. If an IT does not make good use of the skills of the human being, or if it causes fatigue or discomfort, it will not provide overall usability despite its performance in other areas. In this work, then, we will evaluate based on multiple performance measures that cover a wide range of application and user requirements.

3.4 Range of Evaluation Methods

Research in HCI has introduced a wide range of interface evaluation techniques, as discussed earlier. Evaluators have a choice regarding the statistical validity of their tests, the number of users involved, the time and effort required, and the results they wish to achieve. In this research, we feel that many of these techniques are appropriate for various stages of evaluation.

Initially, we come to look at these interaction tasks and techniques with very little concrete information, except our experience with them in applications, and in a few cases the published evaluations of others. Our first goal is to establish a taxonomy and perform categorization, but this is difficult given limited information. Therefore, in many cases it is appropriate to perform some informal evaluation at the beginning to gain a base of understanding of both the task and techniques. This may take the form of a guideline-based evaluation, where one or more usability experts try the techniques and note obvious problems and successes. In many cases, since there are few guidelines or experts in this field to draw from, an informal user study would be useful, in which a few users try out the techniques on some representative tasks, and their general performance and comments are noted. Finally, if the techniques have already been implemented as part of an application, a usability study with some quantitative measures may provide some good information.

Once we are familiar with the task and some techniques, we can create an initial taxonomy and formal framework for evaluation. Within this framework, more formal experimentation can be performed. These experiments are likely to be quantitative, statistically valid, and low-level (meaning that the test does not involve a full application, but only a tightly-controlled system with low-level interaction tasks). In order to further our understanding, these experiments should focus on specific technique components and performance measures, so that it can be determined what the important variables are. From these results, we can refine our taxonomy and evaluation framework, and prepare for testbed evaluation, which is described in the next section.

All of these types of evaluation lead to both specific results and practical guidelines (hypothesis 2) that apply to VE interfaces.

3.5 Testbed Evaluation

The experimental methods and other evaluation tools discussed above can be quite useful for gaining an initial understanding of interaction tasks and techniques, and for measuring the performance of various techniques in specific interaction scenarios. However, there are some problems associated with using these types of tests alone.

First, while results from informal evaluations can be enlightening, they do not involve any quantitative information about the performance of interaction techniques. Without statistical analysis, key features or problems in a technique may not be seen. Performance may also be dependent on the application or other implementation issues when usability studies are performed.

On the other hand, formal experimentation usually focuses very tightly on specific technique components and aspects of the interaction task. An experiment may give us the information that technique X performs better than technique Y in situation Z, but it is often difficult to generalize to a more meaningful result. Techniques are not tested fully on all relevant aspects of an interaction task, and generally only one or two performance measures are used.

Finally, in most cases, traditional evaluation takes place only once and cannot truly be recreated later. Thus, when new techniques are proposed, it is difficult to compare their performance against those that have already been tested.

Therefore, we propose the use of *testbed evaluation* as the final stage in our analysis of interaction techniques for universal VE interaction tasks. This method addresses the issues discussed above through the creation of testbeds – environments and tasks that involve all of the important aspects of a task, that test each component of a technique, that consider outside influences (factors other than the interaction technique) on performance, and that have multiple performance measures.

As an example, consider a proving ground for automobiles. In this special environment, cars are tested in cornering, braking, acceleration, and other tasks, over multiple types of terrain, and in various weather conditions. Task completion time is not the only performance variable considered. Rather, many quantitative and qualitative results are tabulated, such as accuracy, distance, passenger comfort, and the “feel” of the steering.

The VEPAB project (Lampton et al, 1994) was one research effort aimed at producing a testbed for VEs, including techniques for viewpoint motion control. It included several travel tasks that could be used to compare techniques. However, this testbed was not based on a formal understanding of the tasks or techniques involved.

In this work, we have created a series of testbeds for the universal VE interaction tasks of viewpoint motion control, selection, and manipulation. Together, these testbeds make up VR-SUITE – the Virtual Reality Standard User Interaction Testbed Environment.

The testbeds will allow us to analyze many different ITs in a wide range of situations, and with multiple performance measures. Testbeds are based on the formalized task and technique framework discussed earlier, so that the results are more generalizable. Finally, the environments and tasks are standardized, so that new techniques can be run through the appropriate testbed, given scores, and compared with other techniques that were previously tested.

3.6 Models of Human Performance

Testbed evaluation provides us with a good and general technique for comparing interaction techniques designed for a given task, but this is not the ultimate goal of this research. Rather, we want to be able to design interaction techniques and applications that are more usable and cause users to be more productive. In this light, knowing that a certain technique outperforms another in the tasks required by our application is not good enough, because the best technique may not have been thought of yet! What we really desire, then, is a quantitative model of task performance that lets us determine whether we have reached near-optimal performance, and if not, how we can come closer to it.

If our testbeds were simply representative sets of tasks and environments that seemed intuitively to test techniques fully, it would be difficult or impossible to generalize the results into a performance model, and any model that was created would be quite coarse-grained. However, since the testbeds are grounded in a formal framework that splits tasks, techniques and other factors into fine-grained components, we can create models based on these components which should generalize to produce models that predict the performance of even techniques that were not tested.

We believe there are many benefits of using testbed evaluation combined with formal frameworks to produce models of human performance on the various interaction tasks. However, performance modeling is outside the scope of the current research, and we have left it as future work (chapter 7).

3.7 Application of Results

Testbed evaluation produces a set of results that characterize the performance of an interaction technique for the specified task. Performance is given in terms of multiple performance metrics, with respect to various levels of outside factors. These results become part of a performance database for the interaction task, with more information being added to the database each time a new technique is run through the testbed.

The last step in our methodology is to apply the performance results to VE applications, with the goal of making them more useful and usable. In order to choose interaction techniques for applications appropriately, we must understand the interaction requirements of the application. We cannot simply declare one best technique, because the technique that is best for one application will not be optimal for another application with different requirements. For example, a VE training system will require a travel technique that maximizes the user’s spatial awareness, but this application will not require a travel technique that maximizes point-to-point speed. On the other hand, in a battle planning system, speed of travel may be the most important requirement.

Therefore, applications need to specify their interaction requirements before the correct ITs can be chosen. This specification will be done in terms of the performance metrics which we have already defined as part of our formal framework. Once the requirements are in place, we can use the performance results from testbed evaluation to recommend ITs that meet those requirements. These ITs, having been formally verified, should increase the performance levels (including usability) of the application (hypothesis 3).

3.8 Summary of Methodology

Figure 2.2 summarizes the basic design and evaluation methodology we will use for our research on interaction techniques for immersive virtual environments, including each of the components discussed in the previous sections. It should be noted that this process may be slightly different in individual cases, but our design, evaluation, and application will generally follow a procedure similar to this.

For each universal interaction task, the process begins with informal evaluation techniques: observation, user studies, and/or usability evaluations. These should lead to an understanding of the task and the space of possible techniques, which allows us to create a taxonomy and to categorize existing and proposed ITs, and may also inspire the creation of new techniques. We can also list outside factors influencing performance and performance measures at this time. Once this formal framework is in place, we can perform more formal experiments, involving specific task and technique components and performance measures. These results, along with our design framework, may lead to the design and implementation of novel techniques for the task. Also, experimentation may cause some reworking of the initial taxonomy. When the formal framework is judged complete, we can move to the final analysis step: testbed evaluation. Use of the testbed with a range of techniques and performance measures produces a dataset of results for the given task, which can then be used to make an informed choice of ITs for the target application(s), given their performance requirements.

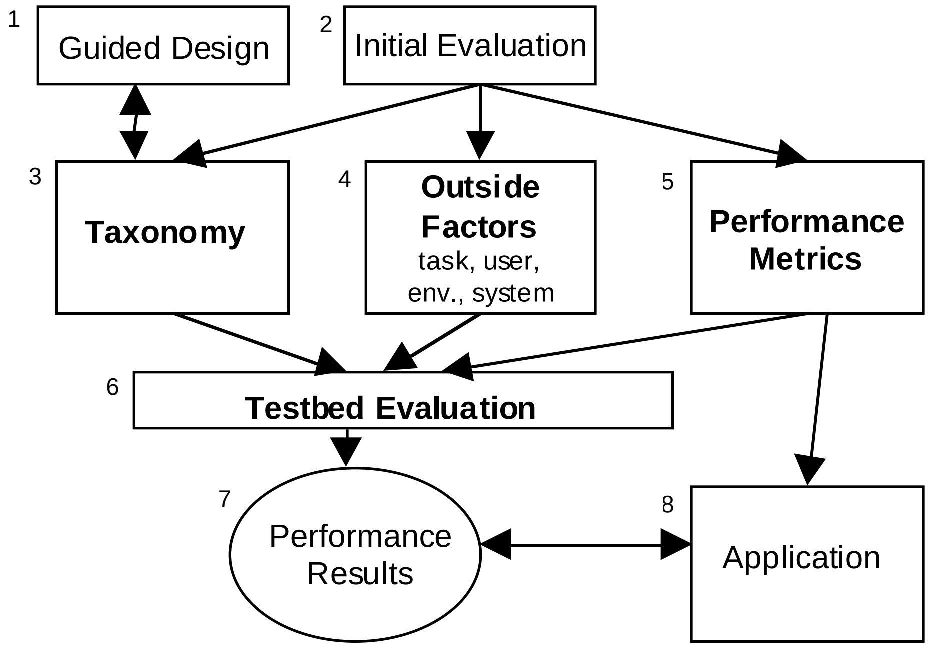

Figure 2.2 Flowchart of Design and Evaluation Methodology