

TABLE OF CONTENTS

| | |
|--|----------|
| CONFIDENTIALITY AGREEMENT | <i>i</i> |
| TABLE OF CONTENTS | 1 |
| LIST OF FIGURES | 2 |
| 1. INTRODUCTION | 3 |
| 2. EXECUTIVE SUMMARY..... | 5 |
| Figure 2-1..... | 5 |
| 2.1 Oz PRODUCTS..... | 6 |
| 2.2 PRODUCT FEATURES | 8 |
| 2.3 TECHNICAL SUMMARY | 10 |
| 2.4 MANUFACTURING | 11 |
| 2.5 MARKETS | 12 |
| 2.6 THE COMPANY | 12 |
| 3. THE EduSpace PRODUCT | 15 |
| 3.1 COMPUTER-BASED TRAINING..... | 15 |
| 3.2 THE EduSpace APPROACH TO COMPUTER-BASED TRAINING. | 17 |
| 3.3 THE EduSpace TRAINING ENVIRONMENT | 20 |
| 3.4 EduSpace COURSEWARE DEVELOPMENT TOOLS | 23 |
| 3.5 THE EduSpace EXPERIENCE..... | 31 |
| 4. MARKETING and SALES..... | 43 |
| 4.1 THE MARKET | 43 |
| 4.2 THE COMPETITION..... | 45 |
| 4.3 Oz COMPETITIVE ADVANTAGES | 48 |
| 4.4 MARKETING STRATEGY..... | 51 |
| 4.5 ECONOMIC ADVANTAGES | 52 |
| 4.6 COMMUNICATION PLAN..... | 58 |
| 4.7 CUSTOMER SERVICE | 59 |
| 5. THE COMPANY | 61 |
| 5.1 Oz CORPORATE DIRECTION..... | 61 |
| 5.2 PRODUCT DEVELOPMENT..... | 62 |
| 5.3 FINANCIAL SUMMARY | 68 |
| 6. EduSpace TECHNICAL DESCRIPTION..... | 73 |
| 6.1 TECHNICAL OVERVIEW..... | 73 |
| 6.2 INTERFACE HARDWARE | 75 |
| 6.3 COMPUTATIONAL HARDWARE..... | 78 |
| 6.4 MATHEMATICAL APPROACH | 83 |
| 6.5 SYSTEMS SOFTWARE..... | 88 |
| 6.6 ENTITY SOFTWARE..... | 91 |
| 6.7 DOMAIN AND INSTRUCTIONAL MODELS..... | 101 |
| 7. GLOSSARY | 109 |
| 8. REFERENCES..... | 115 |
| 9. APPENDIX: Courseware Development Examples | 117 |
| 10. EXTENDED TABLE OF CONTENTS | 123 |

LIST OF FIGURES

| | |
|--|-----|
| Figure 2-1: The EduSpace Advantage..... | 5 |
| Figure 2-2: EduSpace IDW-1 Developer Station..... | 7 |
| Figure 2-3: EduSpace PS-1 Presenter Station | 7 |
| Figure 2-4: Proforma Financial Highlights..... | 13 |
| Figure 3-1: The Designer's Environment | 21 |
| Figure 3-2: The Student's Environment | 23 |
| Figure 3-3: EduSpace Instruction Components | 25 |
| Figure 3-4: EduSpace Development Tools..... | 25 |
| Figure 3-5: Elements of Eduspace Instruction Components..... | 26 |
| Figure 3-6: EduSpace Lesson Organization Display..... | 30 |
| Figure 3-7: Lesson Display During Development..... | 34 |
| Figure 3-8: Lesson Display of SpecGraphs | 37 |
| Figure 3-9: SpecGraphs for a Carburetor..... | 37 |
| Figure 3-10: SpecGraphs for Fuel Flow | 38 |
| Figure 4-1: 1991 U.S. Training Market..... | 43 |
| Figure 4-2: Competitive Product Features..... | 50 |
| Figure 4-3: Cost Comparison..... | 56 |
| Figure 4-4: Oz Unit Sales Forecast | 59 |
| Figure 5-1: Product Development Phases..... | 63 |
| Figure 5-2: Oz Product Cost of Goods | 69 |
| Figure 5-3: Proforma Balance Sheet..... | 70 |
| Figure 5-4: Proforma Statements of Income | 71 |
| Figure 5-5: Proforma Cash Flow Statement..... | 72 |
| Figure 6-1: EduSpace Hardware System Architecture | 79 |
| Figure 6-2: EduSpace Accelerator Board | 81 |
| Figure 6-3: EduSpace Software System Architecture..... | 89 |
| Figure 6-4: Architecture of the Monitor-Coach-Tutor | 108 |

1. INTRODUCTION

Amazingly, leaders in industrial boardrooms, organized labor and government all agree on an important issue: America's workers are ill-trained to meet current employment needs, and are ill-prepared for the rapidly changing requirements of the workplace. Oz... International, Ltd. offers a new concept in computer-based training that can help solve this problem in a cost effective way.

The costs of training and retraining our workforce are huge. Recent studies state the following:

1. American business and industry spend nearly \$300 billion per-year as of 1987 on training. This figure includes close to \$50 billion for the direct cost of formal training, another \$100 billion to deliver informal training, and \$150 billion for trainees' time. (Mincer, forthcoming)
2. By the end of the decade, 75% (125,000,000) of today's workers will need retraining to update current job skills. ("Year 2001: High-Tech Service Report")
3. Workers should expect to change careers five or six times during their lifetimes. ("Year 2001: High-Tech Service Report") Each change requires workers to gain new skills.

Over the past two decades, the U.S. productivity growth has slowed to a crawl. Incomes have stagnated and the wage gap has widened between our nation's educational haves and have nots. For instance, Levy (1988) reports that the annual earnings gap between men in the 25-34 age bracket with 4 years of college and men with 4 years of high school grew from \$3,925 in 1973 to \$9,405 in 1986 (in constant 1987 dollars).

It is necessary to "work smarter" to increase the basic productivity of our nation's work force. But the basic job tasks to be learned are becoming increasingly complex. Until now no one has offered a coherent approach to solving the problem of how to teach complex information to the country's workforce. In short, the problems are:

- Continuing rapid technology change in our workplace. Not only is technology evolving, but also information and technology are crossing disciplines, creating gross a mismatch of skills to tasks.
- Increased complexity of the tasks to be learned. Training must be applicable to tasks involving a wide range of both sensorimotor and cognitive skills.
- Decreased effectiveness of public education. Training must compensate for the inadequate educational level of the general workforce.

- Decreased availability of skilled instructors and increased number of students. Technology and innovation must make up for the high learner/instructor ratio in the corporate classroom.
- Increased severity of economic constraints. Training methodology must achieve increased cost-effectiveness in the workplace.
- Increased need for inter-individual communication. Training must be applicable to team tasks as the work environment shifts to cooperative working groups.
- Increased awareness of the training predicament. In an NBC election poll on September 19, 1992, voters rated jobs and education the number one problem of the country. The formation of coalitions such as "Workforce 2000" acknowledge that we face a serious problem.

In order to remain competitive, our work force needs new tools. Fundamentally, what is required is a *just-in-time training concept*, and a vehicle to facilitate its development and delivery. Oz...International Ltd. has these tools in hand.

2. EXECUTIVE SUMMARY

Oz... International Ltd. is a select team of researchers, educators and industrialists brought together by George Cone as an outgrowth of work with the Human Interface Technology Laboratory (HITL) at the Washington Technology Center on the University of Washington campus.

Oz... International Ltd. is a Washington State Corporation whose goal is to be the world leader in computer-based training development and presentation systems. Oz will attain this goal by being a system integrator which will develop a suite of products for the multi-billion dollar industrial and professional training marketplace.

Oz calls this new training environment EduSpace™. By using a rulebased approach in its instructional development software, Unix-based hardware with multimedia input and display, and interactive virtual reality (VR) environments, EduSpace will reduce the time required to develop training programs. EduSpace will enhance learning by giving the trainee a simulated work experience which approximates the actual job experience. The VR capability in EduSpace will make job-related task simulation affordable to many companies for use as a training tool, in much the same way commercial airlines and the military have used flight simulators to train pilots.

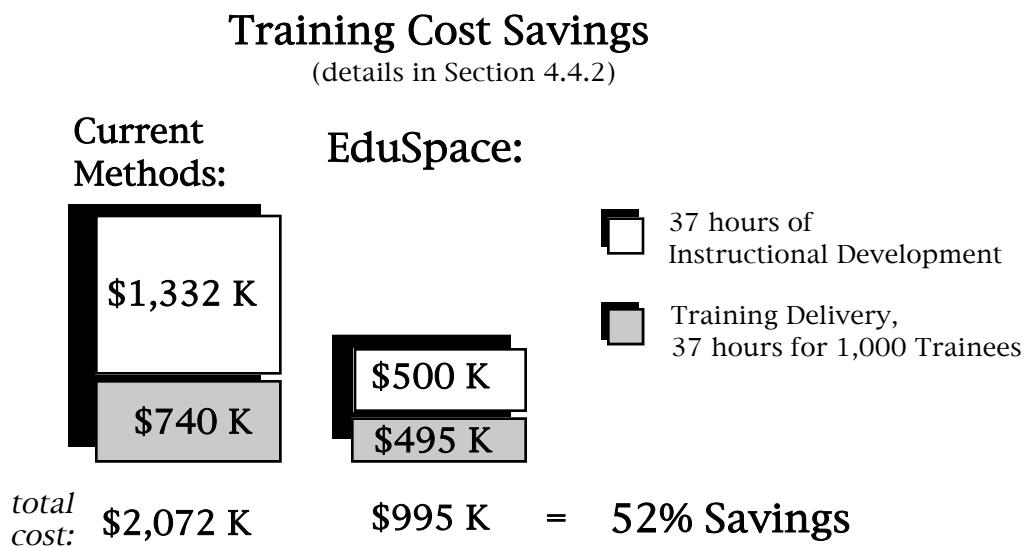


Figure 2-1: The EduSpace Advantage

Figure 2-1 illustrates that Oz EduSpace products will save money and increase performance, providing a strong advantage in building an installed base of customers.

2.1 Oz PRODUCTS

As a systems integrator, Oz will be combining various O.E.M. hardware and software with its own proprietary products. The Oz proprietary products are protected from competition by a combination of patent, licenses, and intellectual property positions.

The initial products are OASIIS™, the Instructional Development Workstation (IDW-1) and the Presenter Station (PS-1). These three products form an integrated computer-based training (CBT) system that aids and empowers the instructional designer while presenting involving and effective educational experiences to trainees.

The EduSpace environmental approach to situational learning represents a paradigm shift in how information is organized and presented. The EduSpace method differs from traditional CBT in two fundamental ways:

- MATERIALS: EduSpace enables efficient collection and management of a rich variety of instructional information
- METHODS: EduSpace provides an interactive training environment that presents information in dynamic response to student behavior.

Oz EduSpace products reduce training costs by providing instructional designers with powerful multimedia and virtual environment information management tools, and by providing students the opportunity to learn efficiently through direct experience. The initial Oz products:

OASIIS (Optical and Speech Interactive Input System)

OASIIS is initially being sold as an O.E.M. product to the CAD companies as a six dimensional input device for 3D applications (replacing the conventional mouse, trackball, etc.). It is the primary interactive device in the Presenter Station and a key component of the Instructional Development Workstation.

The IDW-1 (Instructional Development Workstation)

The Instructional Development Workstation (Figure 2-2) serves as a productivity tool for the instructional designer or the domain expert in developing training courseware. EduSpace, the Oz proprietary software for computer augmented instructional design and display, is embedded in the IDW-1. The IDW-1 is analogous to the engineering workstation but is designed for the instructional training environment. Its capabilities include *capturing and editing of information* (from video camcorder, optical disk, existing CAD data, scanned text and graphics) for the multimedia knowledge database, and *asserting inference and interactivity rules* for the knowledge database elements. Word processing, model making, graphic editors and other existing industrial standard application programs are integrated into the normal operations of the EduSpace development system.

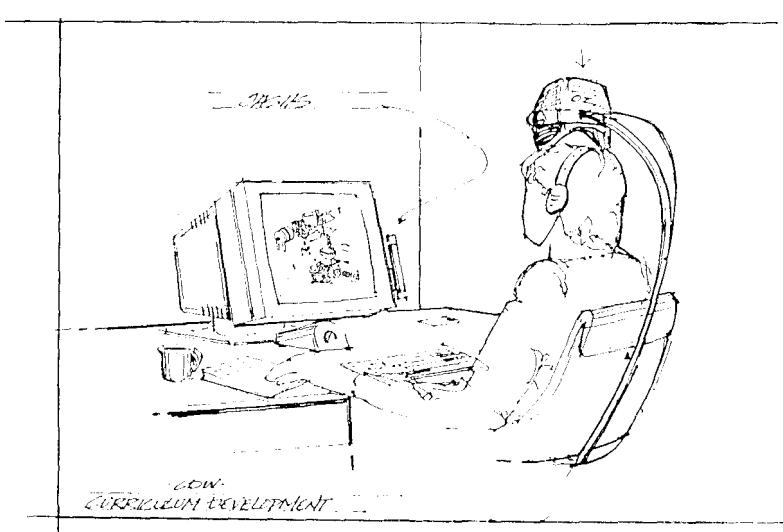


Figure 2-2: EduSpace IDW-1 Developer Station

Instructional designers use supportive tools to develop compelling instructional courseware.

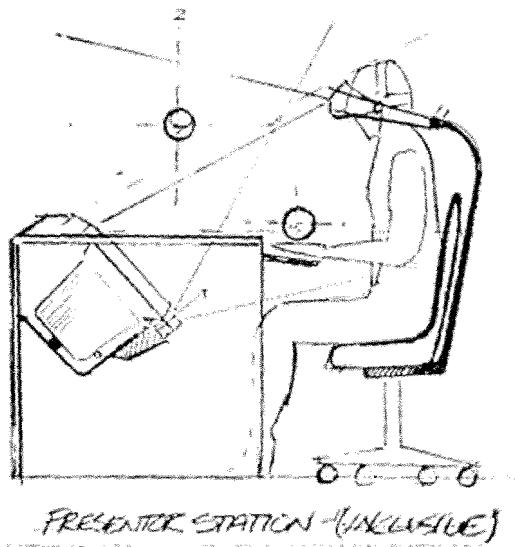


Figure 2-3: EduSpace PS-1 Presenter Station

Trainees move between participation in a virtual world and interactive multimedia experiences while learning new tasks.

The PS-1 (Presenter Station)

The Presenter Station (Figure 2-3) is a training workstation designed for student use. The course material generated on the IDW-1 is presented as courseware by the PS-1.

The overall system design is unique in employing both interactive multimedia (video, text, graphics and audio) and 3-D inclusive environments (VR).

2.2 PRODUCT FEATURES

It is currently anticipated that the first Oz product, OASIIS, will be in beta test within nine months and available to the market in twelve months. The flagship training products, the Instructional Development Workstation and the Presenter Station, will follow within eighteen months. Among the important features of the EduSpace system are the following:

The EduSpace Interface Chair

Information is presented to the user both on a workstation monitor and via a wide field-of-view inclusive display that combines high resolution optics and audio headphones. This *direct experience* audiovisual display is supported by an omni-directional counterbalanced suspension system integrated with an award-winning ergonomic chair. The chair provides the user with a safe, comfortable, controlled and unobtrusive work space while interacting with either inclusive or non-inclusive environments. The Interface Chair will enable the user to focus on the task at hand rather than on the particular interface devices being used.

EduSpace Accelerator Board

The EduSpace Accelerator Board provides the computational power necessary for modeling and interacting with 3-D virtual worlds. It is customized to integrate input from OASIIS and from the Interface Chair. The accelerator contains two specialized CAM chips that produce super-computer performance for the limited domain of EduSpace modeling and interactivity.

EduSpace Development Tools

Oz proprietary software includes five powerful integrated tools sharing a common interface on the IDW-1 platform. Most centrally, EduSpace is designed for *ease of use*. The EduSpace development environment lets software assume much of the responsibility for the most arduous and time-consuming aspects of developing and delivering lessons. The system can be used by instructional design teams or by individual instructional development professionals.

EduSpace tools guide users through interactive specification, modification, and monitoring of lessons. EduSpace tools and their functions include:

- **TaskMaster:** Instructional design and sequencing
- **ProfileBuilder:** Student modeling and tracking

- **MediaMaster:** Digital media acquisition and editing
- **RuleBuilder:** Inference and interaction editing
- **TrainTrack:** Session monitoring & communications interface

These task-specific EduSpace tools are fully integrated to share and cross-reference information, for efficient assembly of course materials. Constrained data entry fields within tools reduce redundancies and omissions in constructing lessons. By offering coordinated tools that address the full scope of the computer-based instructional design process, EduSpace makes a unique contribution to the training industry.

EduSpace Courseware

The ability to reconfigure and reuse lesson components is of primary importance in developing training courseware. Rapid re-purposing of task content is not achievable with existing authoring packages: changes in course content usually require tearing the entire lesson apart, inserting the new changes, deleting unwanted portions and reproducing the updated courseware. Re-purposing is nearly as costly as producing a program in the first place.

In contrast, the EduSpace architecture provides modular lesson components which the instructional designer uses to assemble activities for different training tasks (rather than starting with a blank slate as current CBT authorware does). The modularity of EduSpace components allows instructional designers to collect domain models, inference procedures, instructional methods and strategies, graphical and audio information, and multimedia and virtual environments for reuse in different training courses.

EduSpace treats the lesson presentation as a *virtual environment*. Virtual environments can be two dimensional multimedia displays or three dimensional inclusive displays, or a combination of both. EduSpace uses the same computational metaphor for graphic user interfaces as it does for interactive virtual realities, providing designers with both modeling flexibility and a simple, homogeneous interface.

Most importantly, these lesson components are not merely graphic and simulation primitives, but instructional designs and pedagogical strategies embodied within EduSpace as teaching knowledge. The resulting open system allows the instructional designer to bring in his or her own materials, share materials with other designers, and accumulate libraries of reusable components.

Proprietary Technology

Oz technology is the result of many years of research in the field of human interface with computers, artificial intelligence in training, and computer architecture. Oz integrates the following technologies into its product line:

- OASIIS Position Tracking and Speech Recognition
- EduSpace Accelerator Board

- EduSpace Operating Shell
- EduSpace Instructional Development Tools
- Interactive Multimedia and Virtual Reality Courseware

2.3 TECHNICAL SUMMARY

Oz EduSpace products combine to provide a rapidly configurable workstation environment with powerful development tools for instructional designers, for training instructors, and for training participants. The Oz technology suite is a tightly integrated collection of proprietary technologies for developing and delivering effective training experiences using multimedia and virtual environments.

The *goals* of the EduSpace technical architecture are:

- To create a product that combines modern learning theory with modern computational techniques, a tool with sufficient power and simplicity to substantively improve the training of the American workforce,
- To provide solution paths for known technical difficulties associated with bringing complex and effective training experiences to learners using high technology media,
- To integrate the various levels of computation (hardware, software, modeling, interface) into a sound, tractable product architecture,
- To allow for the utilization of innovative techniques with known performance gains while minimizing risk.

To achieve complex functionality while keeping EduSpace simple to use, Oz has designed a technical architecture which combines high performance hardware and advanced software with an intuitive interface. The result is a system based on a single computational activity at the silicon level which performs an extremely diverse set of functions at the interface level while being easy to use.

The technical plan has been designed to utilize recent advances in hardware and software techniques for computer-based training. The key innovations in the EduSpace product are:

- Intuitive interface tools,
- Qualitative process models,
- Entity-based software,
- Virtual and multimedia operating system,
- Boundary mathematics,
- VE-Chip and MS-Chip, and
- OASIIS and the Interface Chair.

The technical problems associated with computer-based training and with display of complex information are addressed by the EduSpace integrated architecture which

incorporates the above innovations. The entity-based software architecture is the product of many years of research development at the University of Washington and at Stanford University. The EduSpace software and hardware interface (the Development Tools, OASIIS and the Interface Chair) are innovations being developed solely by Oz, and are original as well as unique.

The accelerator chips have been prototyped for a completely separate NASA application, and will be substantively modified for the Oz architecture. Boundary mathematics computing techniques have been under development for over fifteen years and are implemented for parallel theorem proving and for rule base optimization. Oz will be extending these techniques to relational graphs. The EduSpace operating system has been developed over the last three years at the University of Washington. Finally, the qualitative modeling approach is the result of over a decade of research in computer-based training technologies at Stanford University.

2.4 MANUFACTURING

Oz will subcontract and purchase as O.E.M. all the major sub-systems for its products. The company will control all major tooling and sub-systems specifications but avoid the heavy capitol outlay and inflexibility of vertical integration. Final assembly and testing will be performed by the company with very strong quality control measures carried from the smallest component through final test. As a system intergrator, strong corporate partnering, software licensing and corporate agreements will be a way of life.

At the present time Silicon Graphics will be the computer of choice followed by IBM/Apple's new computer to be available in 1994. Serious investigation of the Sun system is underway. The Interface Chair will be procured from HAG, a U.S. company with Norwegian based Corporate Headquarters, under special agreements that include proprietary modifications.

2.5 MARKETS

Oz's initial target core market will be domestic corporations with 10,000 or more employees engaged in manufacturing which spend the bulk of the estimated \$50 billion on formal training, existing large curriculum development corporations, and the Department of Defense, which spent an estimated \$18.3 billion on formal training during its fiscal 1990. Secondary priorities will be companies with between 2,500 to 9,999 employees in manufacturing and the health service industries and foreign companies.

The focus on large corporations is for two reasons: first, corporations have the available dollars to purchase the system; second, corporations will invest in products which can demonstrate savings and increased productivity. The large corporation is more likely to already have significant training budgets and sophisticated professionals in charge of spending. With a large employment base, these companies will have the most to gain from the Oz products and will be the first to understand the savings and increased productivity the products offer. Additionally, the people in charge of training at these

corporations are at the forefront of the latest training techniques and will be able to appreciate the advancements Oz offers to the training industry.

2.6 THE COMPANY

Oz... International Ltd. is located in Seattle, Washington. It has assembled the technology and a management team with the skills and experience necessary to meet its objective *to become the worldwide leader in the training marketplace.*

The key technologies and corporate management, including the Board of Directors, are identified in Section 5.2.2: Management Team.

Due to its unique relationship with the Washington Technology Center, the Human Interface Technology Laboratory, and the HITL Industrial Consortium members, Oz enjoys both a special industrial insight to emerging markets and a strong potential for networking and strategic alliance.

The company has established unique relationships with four other outstanding organizations that maximize the potential of success: The Institute for Research on Learning, Silicon Engines Corporation, Matty Computer and the HAG Corporation.

Projected income, balances and share values are presented in the Proforma Financial Highlights table which follows.

Company Strategy

- Form strong synergistic corporate relationships with educational and industrial partners to acquire a unique technology and market base.
- Develop the components of the Oz technical plan into the EduSpace instructional development system. Emphasize ease of use by providing intuitive interface software and hardware products. Release OASIIIS as the first Oz product.
- Utilize advanced VLSI application specific integrated circuit (ASIC) technologies to achieve product superiority, protect proprietary software, reduce cost and keep competitors on in a catch-up mode.
- Become the leading supplier of quality computer-based industrial training systems and proprietary software by creating unique products through system integration.

PROFORMA FINANCIAL HIGHLIGHTS

| Year End December 31, | 1992 | 1993 | 1994 | 1995 | 1996 |
|--------------------------------------|-------------|---------------|--------------|---------------|---------------|
| Income Statement Data: | | | | | |
| Total Revenues | \$ 0 | \$ 1,420,000 | \$ 4,572,220 | \$ 31,790,750 | \$ 62,957,570 |
| Net Income | (\$378,200) | (\$1,386,000) | \$ 14,108 | \$ 5,705,584 | \$ 8,135,583 |
| % Gross Margin | | 49.6% | 61.5% | 70.6% | 65.8% |
| Operating Income as % of Revenues | | -97.6% | .3% | 17.9% | 12.9% |
| Number of CES* | 8,600,000 | 11,600,000 | 12,100,000 | 12,100,000 | 12,100,000 |
| Net Income per CES | (\$0.04) | (\$0.12) | \$0.00 | \$0.47 | \$ 0.67 |
| Value per Share at 20 Times Earnings | N/A | N/A | \$0.02 | \$9.43 | \$13.45 |
| Balance Sheet Data: | | | | | |
| Cash | \$ 632,312 | \$ 315,200 | \$ 404,693 | \$ 1,120,157 | \$ 2,110,393 |
| Total Assets | 1,132,312 | 1,512,200 | 2,647,748 | 10,767,845 | 21,749,786 |
| Short Term Borrowings | 0 | 0 | 0 | 0 | 0 |
| Stockholders Equity | 1,071,800 | 1,185,800 | 2,199,908 | 7,905,492 | 16,041,075 |

*CES = Common Equivalent Share

Assume 3,600,000 shares @ \$0.167 and 5,000,000 shares @ \$0.20 per share in 1992; 3,000,000 shares @ \$0.50 per share in 1993; 1,000,000 shares @ \$1.00 per share in 1994.

9/17/92

Figure 2-4: Proforma Financial Highlights

3. THE EduSpace PRODUCT

To introduce the EduSpace product, the current state of computer-based training is discussed, and then the comparative advantages of the EduSpace approach are delineated. The five integrated EduSpace software development tools are described in Section 3.3, and features of the graphical user interface are illustrated in Section 3.4. Finally, Section 3.5 exemplifies how EduSpace products will be used by each of three types of users: instructional designers, training instructors, and students.

3.1 COMPUTER-BASED TRAINING

Computer-based training (CBT) has several advantages over traditional forms of training. CBT has the potential to provide flexible learning experiences in greater frequency and intensity than are permitted by traditional linear media (text and charts). Using CBT, we can compress time and conditions and we can simulate situations. We can allow learners to manipulate variables in low risk situations and to control their learning experiences to meet their personal learning styles and needs. And these things can be done in a timely and affordable way (Gery, 1987).

However, an examination of existing training development and delivery methods reveals that the potential of CBT has not been fulfilled with current systems. The following comparison between current CBT methods and the EduSpace experience illustrates the value of Oz products.

3.1.1 Current Training Courseware Development

The capability to create practical interactive learning experiences has existed, in varying degrees, for nearly two decades. Sophisticated special-purpose military CBT systems (such as SIMNET) and aircraft industry flight simulators have proved to be highly effective training systems. But their multi-million dollar price tag makes them generally inaccessible to those charged with meeting the increasingly diverse and complex training needs of large organizations .

Affordable tools for computer-based instructional design and delivery have, to date, been fragmented and inadequate. Of the 74 commercial CBT authoring programs available today, not a single one is considered easy to use, particularly by the infrequent user.. (Patton, 1991). Current CBT production is time consuming, costly, and resource intensive. For example, at the lowest levels of interactivity (Level 1), an Interactive Video CBT (IVCBT) lesson can easily require more than 1,000 hours to construct for every single hour of net running time.

CBT courseware is categorized into levels of sophistication, from Level 1 (text-based drills) through Level 4 (fully interactive multimedia environments). As increasing levels of interactivity are embedded in programs (from Level 1 to Level 3), the

development time escalates to more than 2,500 hours of development for a 1-hour IVCBT lesson. The military routinely pays contractor costs of \$3,000 per minute for new IVCBT development time, or \$180,000 per course hour. If the contract calls for a full Level 4 Interactivity of integrated multimedia, the effort quickly surpasses 5,000 hours for one hour of new CBT courseware. Thus, ten hour-long courses at this level typically run over \$1-million in initial development costs (Patton, 1991). Level 4 courseware now exists only in multi-million dollar military and aircraft simulation systems.

The complex process of creating CBT courseware loses its value if whole courses are thrown out when some parts of it become outmoded. As most industrial CBT developers have found, course content is constantly changing, usually immediately after the new program is implemented. For CBT development to be a cost-effective effort, courseware must be flexible enough to "re-purpose", which means that parts of lessons can be edited or replaced easily so that the courseware as a whole is still useful. Rapid re-purposing of usable existing task content is not achievable with existing authoring packages. CBT developers faced with the need for constant updates, and constrained by budget and people limitations, are usually unable to complete courseware revisions until a year or more after the program enters the field.

3.1.2 Current Training Delivery

CBT courseware resulting from current programs are frequently unrealistic and just plain boring, continuing to deliver information in the form of electronic page-turners that require only the lowest level of rote memorization.

The boredom in CBT learning can be magnified by graphic frames based on cartoons and stick figures. For a generation nurtured on slick broadcast video media, today's CBT graphic frames are an embarrassment. Compounding the poor graphic design is a lack of meaningful interactivity between the user and the machine. First, the machine presents a bit of content. The learner reacts; the machine presents some more data, then the learner reacts again, and so on until the learner is exhausted. The format becomes one of..."its your turn, now its my turn."

However, this is *not* how humans interact in real life, and certainly not how expert teachers and learners interact, one on one. The outcome is as expected. The CBT lesson quickly becomes a boring task, which the learner wants to quickly exit .

3.2 THE EduSpace APPROACH TO COMPUTER-BASED TRAINING

Oz provides integrated solutions for current problems in designing and delivering computer-based training. EduSpace's workstation environment offers powerful software functionality and a variety of interactive interface technologies (hypertext, high-resolution graphics, video multimedia and VR) to enable the rapid development

and delivery of compelling situated training courses. EduSpace is *ergocentric*, meaning that the system and the curriculum it helps to construct are designed around the needs and behaviors of the person using it.

It is important to stress that the sophisticated computational environment is only half of the Eduspace advantage. Effective learning is *not* guaranteed by "high-tech" approaches. For applications of educational technology to augment learning, the learning process must be correctly understood and accounted for. Our most current understanding of the way people learn and perform is embedded in the design of EduSpace, for the benefit of both those who develop instruction and those who learn with it. This framework does not impose particular teaching methods, but ensures that EduSpace has the capability to develop compelling and effective learning experiences.

3.2.1 EduSpace Courseware Development and Delivery

EduSpace software tools combine to provide instructional designers with the means to capture, edit and integrate a variety of information forms to use as training materials. The interface provides a modular, self-organizing structure for lesson assembly. Lesson components are arranged so that the designer can easily access and organize the materials and interactions needed to construct courseware.

Courseware consists of guided interactive activities, which can be either individual or group experiences. Training activities for each aspect of the task to be mastered are created by giving *behaviors* to multimedia (text, pictures, graphics, animation, video, voice and sound) and VR elements, so that *the information itself guides and responds to students* as they explore content areas.

VR provides a particularly useful environment for learning. The application of this inclusive computer interface for training is *not* "a solution looking for a problem." The original purpose of VR was to create a more natural way to access and understand complex information. The principles of perception and performance have shaped the evolution of the technology, and research in America and abroad continue to add to our understanding of how to adapt VR more closely to human needs and behaviors. VR is powerful because it is an entirely new way to use computers to access information.

In a virtual training world, we can view and manipulate objects and systems in a natural way. We can reach out and pick up an object up. We can move the object. We can see how the object reacts when we bring it together with another object. We can hear what is going on behind us. We can turn around and look. Thus, motor skills to be learned in a virtual world are almost identical to the skills we use in the real world. By situating real work conditions in virtual training, the information to be learned is presented in a context appropriate to the skills needed in the actual work setting.

The attention of trainees is engaged and focused during hands-on VR training exercises and simulations (M. Bricken 1992). EduSpace combines interactive multimedia and virtual environments in such a way that knowledge can be quickly and actively internalized by the learner, on the spot, in real time. There is no lag between the

formulation of knowledge, learning how to apply this knowledge, and the immediate output in the form of performance. This is a powerful capability in the training environment because the virtual experience allows the learner to move with a purpose, allowing the coordination of cognitive and psychomotor channels. The result: maximized learning leading to mastery performance.

3.2.2 Educational Characteristics of EduSpace

Technology alone does not necessarily improve student performance, but performance increases up to 50% have been demonstrated when interactive visual media is used in combination with sound learning principles (Fletcher, 1991; Mattoon, 1992). EduSpace combines state-of-the-art technology with verified research results in educational theory. Three key educational characteristics contribute to the training effectiveness of Eduspace: situated learning, social learning, and individualized instruction.

Situated Learning

Situated learning views the student as a "situated agent" (Lave and Wenger, 1991). This means we build our knowledge of the world by being an active part of it. Situated learning thus becomes a prescription for instructional activity: people gain knowledge by performance on tasks. According to the theory of situated learning, knowledge is imparted by doing something in a real situation, such as the work setting. Thus, *activity-based learning* leads directly to physical job performance, which can be measured, evaluated, and enhanced.

Researchers are now beginning to pinpoint how the learning and application of knowledge takes place. It has been discovered that the physical components of the brain, at the level of neuronal groups composed of hundreds and thousands of neurons, are always new — not predetermined and causally interacting in the sense of most machines we know — but coming into being during the activity itself, through a process of reactivation, competitive selection, and composition (Edelman, 1987; Smoliar, 1989). In this manner, groups of neurons form new adaptive clusters. As these new networks form, they become specific and unique to the situated experience. This ability allows the brain, and subsequently our muscles, to formulate new patterns of performance, the goal of training.

The power of situated learning theory comes from the fact that it explains how learning is bound to sensorimotor coordination, and why knowing and doing are inseparable. Our ability to speak, to reason, to make sense of things are learned as ways of physically interacting with our environment. Thus, knowledge becomes our capability to adaptively interact with our environments (Clancey, 1992). Thought of in this framework, learning is a *proactive* process; activity-based instruction creates an environment in which the student can incorporate knowledge and cognitive skills as a part of their physical skill-base.

Social Learning

Research in collaborative learning abounds with evidence of its educational value: "The learner tends to be more productive in a group situation than working in isolation. Ongoing discussion involves the active participation of students in the teaching-learning process; attention is then directed toward the learning activity" (Belkin 1977).

People "solve practical tasks with the help of their speech, as well as with their eyes and hands...Human learning presupposes a specific social nature and a social process by which people grow into the intellectual life of those around them" (Vygotsky 1978).

Peer co-discovery and reinforcement are strong forces in the learning process (Johnson, 1983). The social aspect of learning is especially important when trainees will be working together on tasks or projects.

Eduspace provides the capability for individual, one-on-one and group training experiences. Presenter Stations can be networked for communication and cooperative problem-solving in shared student environments. Social VR allows several trainees to interact simultaneously in the same audio-visual environment, sharing control naturally while conversing with each other and the instructor.

Individualized Instruction

Each student approaches new information differently, and forms understanding through unique processes (W. Bricken, 1987a). The Eduspace CBT system accommodates student individuality by constructing a dynamic model of the student as the student moves through the program.

This model is used by the program to present the student with alternative approaches to the subject matter that fit with what the student knows and with the student's cognitive learning style.

EduSpace's optional guidance styles and dynamic interactivity allow students to approach and explore new information in the way that is most natural for them. This provides students with the opportunity to construct their own understanding of the subject matter in the most effective way.

3.3 THE EduSpace TRAINING ENVIRONMENT

EduSpace is designed with three kinds of end-users in mind: *instructional designers* (or design teams, including domain experts who develop training courses), *training instructors*, and *learners*.

EduSpace's ergocentric tools and displays are responsive to behaviors and goals of the particular person using the system. Responsive systems are embedded in the interface to assist the instructional designer in rapid development of training sessions, to aid the trainer in monitoring and enriching the training process, and to guide and correct students while they are engaged in particular activities.

The interface is task specific for each of the user groups. Since different software tools and displays are needed by designers, by instructors and by students, the interface displays a different functional configuration for each group.

3.3.1 The Instructional Designer's Interface

The IDW-1 provides instructional designers with a sophisticated and supportive courseware design environment. Designers can work with text, high-resolution 2-D graphics, MIDI (sound and voice) and video elements on the workstation monitor, interacting with the information using standard input devices (keyboard and mouse). They can use OASIIS to interact with 3-D data on screen, or in combination with the ergonomic Chair-mounted inclusive display to interact with the 3-D information inside virtual environments.

EduSpace integrated software tools guide designers through interactive specification and modification of lesson components. These tools enable courseware developers to capture, edit and sequence elements of media, inference and information behavior to create interactive training activities.

The dynamically nesting lesson format automates the drudgery of organizing and keeping track of the complex information required to perform task analysis and knowledge acquisition functions and then embody performance objectives into staged interactive learning experiences. The designer can see in one graphical display the relationships between lesson components, as well as the relationship of each component to the overall structure of the lesson. This representation also allows direct access to nested components.

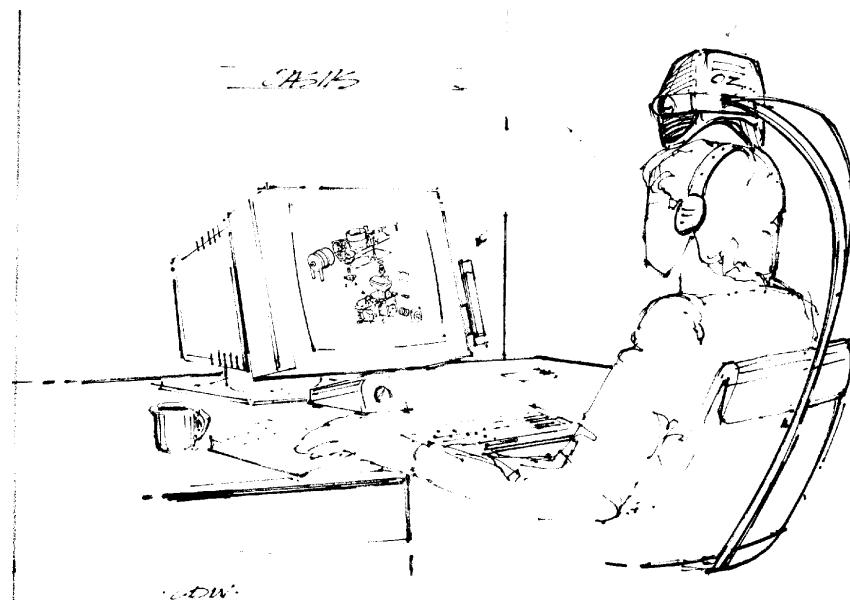


Figure 3-1: The Designer's Environment

An important feature of the EduSpace interface is that it allows *non-sequential modular development*. Lessons are pre-organized into related instructional components. Development of lessons can be initiated or edited from within any component. One instructional designer might start developing his lesson with the task analysis component, another might begin with behavioral objectives, and a third might start with knowledge acquisition. Moving between instructional components during the lesson construction process is facilitated by the dynamic display.

Task and student modeling are supported by a qualitative and process modeling approach, which expresses behaviors (both dynamic student behavior and the behavior of instructional information) in terms of propositional statements and relational graphs that specify the attributes and behaviors of *entities*. (See Section 6 for technical descriptions of qualitative process modeling and entity-based programming.)

Entities are the primary organizational structure in the EduSpace software architecture. Entities are collections of computational resources that accomplish a specific task in an interaction with the learner. They are used to specify objects in multimedia and virtual environments, and they are used to specify activity sequences in instructional designs.

The EduSpace interface reflects the Oz system's entity based software architecture. The structure of lessons and of activities is operationally consistent with the task structure of system entities, allowing each activity within a lesson to behave as an independent software entity with its own set of attributes and rules. This feature allows designers to reorder and substitute activities in a lesson and lessons in a course, providing design flexibility, easy course updates, and lesson reusability. This modular approach to instructional design is further supported by the use of open media, inference and interaction libraries.

3.3.2 The Training Instructor's Interface

The IDW-1 Workstation can also serve as a training monitor station in networked EduSpace CBT systems, from which the trainer can observe and participate in individual or group training courses. The session monitor interface arrays small replicas of each student's display on the instructor's screen, allowing the trainer to concurrently observe the progress of each student.

The instructor can initiate and coordinate *parallel communications* to enable conversations with and between individuals or groups. During a training session, the instructor can branch on-line to relevant information, ask learners to repeat material, and present alternative explanations when the initial one doesn't work. Learners can query the instructor for clarification, repetition, additional information, or anything else they think might help them understand.

The instructor can link students together for cooperative work, and he can enter virtual environments with students using his inclusive display, for one-on-one or group teaching.

Administrative software support automates many of the trainer's scheduling, tracking and certification chores. As well as keeping track of class statistics and performance

records, it registers the progress of students with graphical displays that enable immediate evaluation of the success of particular lessons, of individual trainees or of training sessions.

3.3.3 The Student's Interface

The EduSpace Presenter Station presents the courseware designed on the IDW-1 in a diversity of delivery formats ranging from screen-based text with 2-D and 3-D graphics and video, to multisensory inclusion in a virtual environment.

The way that lesson activities are presented is another unique aspect of EduSpace. *Parallel activities* are provided for students to explore in the way that is most effective for them. Different activity options and sequences are initiated in direct response to each student's choices and behaviors. One student may move directly into a hands-on VR experience with the lesson's subject matter, while another chooses to listen to an expert explain that subject, and a third joins a small on-line group working with interactive animations on the same topic.

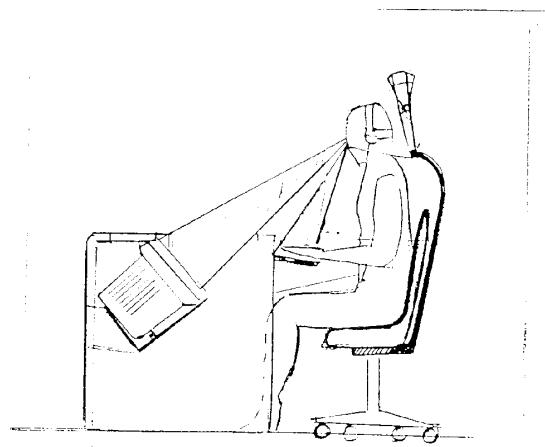


Figure 3-2: The Student's Environment

3.4 EduSpace COURSEWARE DEVELOPMENT TOOLS

The benefits of well designed interactive CBT are clearly conceived and demonstrably effective. But the process of planning, editing, programming, and testing interactive CBT courseware is a precise and demanding task. The very nature of the computer medium requires specificity in articulating design, scripts, displays and logic. Designing instructional activities requires different kinds of information (text, voice, graphics, video) to be retrieved, sorted, linked and organized. The appendix contains two examples of today's CBT development process as an illustration of the complexity of the task.

While EduSpace cannot reduce the scope of the job, it can make it easier in three ways:

- Developing effective interactive CBT presently requires a large team of experts. The instructional designer currently works with a subject matter expert, a writer, an editor, a data entry specialist, an authoring system specialist or programmer, a media expert, a graphics designer, an interactive CBT specialist, a production administrator and a CBT administrator (Gery, 1987). *EduSpace combines and automates many of the activities of expert teams to reduce the number of specialized roles needed in the courseware development process.*
- A minimum of nine different kinds of software productivity applications are required by the instructional design team: project organization tools, outline processors, design tools, flowcharting software, authoring systems, programming or authoring languages, graphics software, and recordkeeping systems. *EduSpace offers integrated development tools that share data structures and a common interface, to reduce development time, while also allowing the import of standard file formats from external applications.*
- The current approach to CBT design lacks a coherent organizational structure. *EduSpace courseware is organized into instructional components which embody the full scope of the CBT design task, to reduce production errors and increase the quality of courseware.* (Figure 3-3).

The EduSpace interface combines and coordinates *instruction components* using rule-based tools to assist the process of designing and delivering training, as illustrated in Figure 3-4.

The *elements* of these instruction components are the kinds of information that EduSpace coordinates and tracks for the instructional designer, the instructor, and the student. These elements are listed in Figure 3-5.

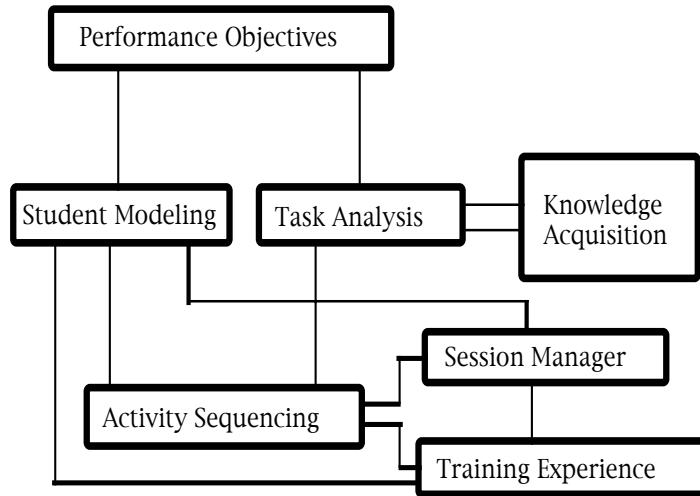


Figure 3-3: EduSpace Instruction Components

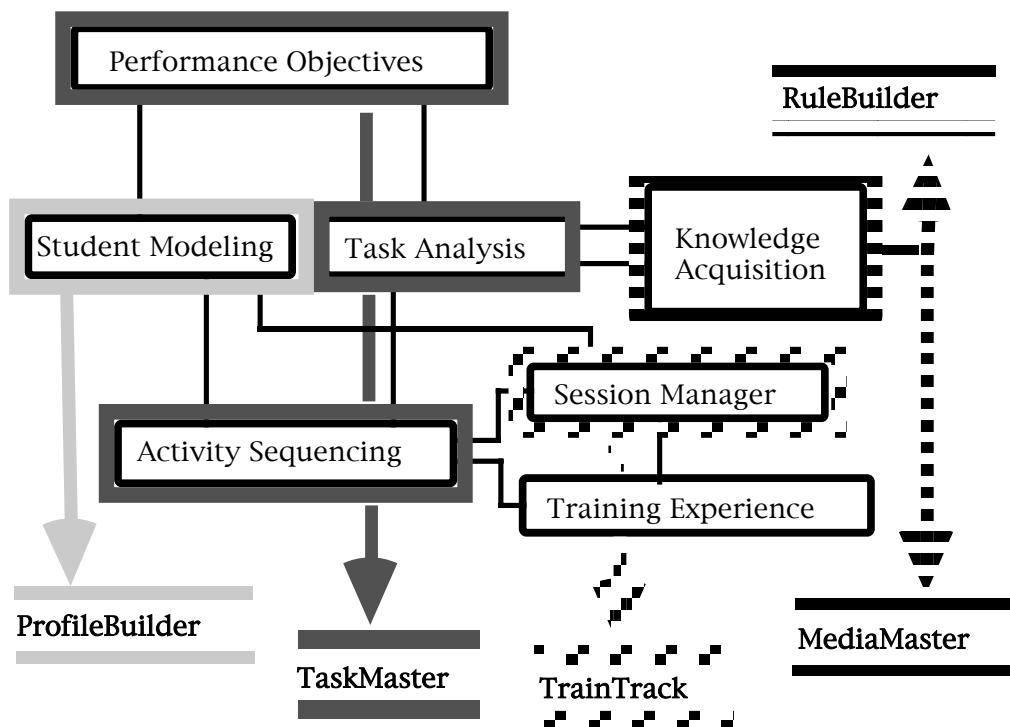


Figure 3-4: EduSpace Development Tools

Performance Objectives:

- *Operational Skills*
do: operation/sequence
- *Cognitive Skills*
know: fact/concept
relate: objects/processes
synthesize: solve problem
construct: create solution

Student Modeling:

- *Target student*
- *Individual students*
interaction patterns
test results
sensory strengths
cognitive style
virtual body map/files

Task Analysis:

- *Elements*
- *Steps*
- *Sequences*
- *Constraints*
- *Rules*
- *Heuristics*

Activity Sequencing:

- *Methods*
description
example
demonstration
interaction
construction
- *Materials*
knowledge elements
media elements
- *Capabilities*
see
hear
touch
enter
move
select
manipulate
edit
construct
- *Strategy [Monitor-Coach-Tutor]*
guidance
narrate
notify
suggest
critique
error remediation
allow
constrain
notify
correct
consequent result
testing
methods
choice
fill-in
calculate
demonstrate
results
record/tabulate
branch & loop

Knowledge Acquisition:

- *Knowledge Base*
object data
relationship rules
context constraints
process dynamics
performance heuristics
- *Media Library*
acquire/digitize guide
import standard text
& graphics formats
import & edit:
hypertext
chart/graphic
photos
2-D & 3-D CAD
animation
video
virtual reality (VR)
MIDI sound & voice
instrument data

Session Manager:

- *Lesson Information*
course/lesson outline
activity guide
- *Session Information*
system configuration
Oasiis
inclusive display
MIDI
network links & resources
monitoring windows
participation options
- *Student Information*
administration
student data
group profile
job background
learning styles

Figure 3-5: Elements of Eduspace Instruction Components
3.4.1 The Five EduSpace Software Tools

Proprietary EduSpace software tools include **TaskMaster**, **ProfileBuilder**, **MediaMaster**, **RuleBuilder**, and **TrainTrack**. Each tool is related to one or more instruction components. Each EduSpace tool is discussed in terms of the components it incorporates and its functionality for different users:

TaskMaster

- Eduspace Instruction Components: Performance Objectives, Task Analysis, Activity Sequencing

TaskMaster is the primary tool for organizing and configuring EduSpace courses. It interactively guides the translation of performance objectives into discrete tasks that have particular activity steps and sequences. Each lesson is described using a dialogue-driven or graphical specification process. A lesson reaches closure only when all aspects of the activity (method, materials, capabilities & pedagogical rules) are determined.

MediaMaster

- Eduspace Instruction Component: Knowledge Acquisition

MediaMaster has two functions. It provides digital information acquisition tools and guides for capturing and importing graphical elements (CAD data, video, animation, etc.) It also contains media construction tools for editing and arranging multimedia and for creating and configuring virtual environments.

RuleBuilder

- Eduspace Instruction Component: Activity Sequencing

RuleBuilder allows the entry and editing of knowledge base rules and cases. Rules can be specified in two ways: with propositional statements or with relational attribute graphs. RuleBuilder contains the entity editor, which encapsulates rules and media elements into activities, and activities into lesson modules.

ProfileBuilder

- Eduspace Instruction Components: Student Modeling and Activity Sequencing

ProfileBuilder collects and manages different kinds of individual student information, including interface configuration files, interactivity preferences, dynamic behavior during training, test results, and training history. This information has two purposes.

The first purpose is to provide the EduSpace system with a model of the student to serve as a basis for the dynamic behavior of the *Monitor-Coach-Tutor* expert system module. The embedded Monitor-Coach-Tutor provides flexible interaction strategies within lessons.

The second purpose is to provide the instructor with information about student performance and progress. Student records and real time performance information are transferred into the TrainTrack tool in the form of a smart (cross-referenced data fields with automatic updates) spreadsheet for recording and tracking student information and progress.

TrainTrack

- Eduspace Instruction Components: Session Manager, Student Model

The process of monitoring and participating in student activities is simplified with this separate interface for session management. Course Information and Student Profiles are used to configure classes. *World Windows* show simultaneous views of student perspectives, and *Instructor Flags* are intervention notices signaling need for personal attention. The instructor can address individuals or sub-groups of students on-line, and connect students together for cooperative work. She can move into individual or shared virtual environments for one-on-one or group teaching. This tool also tracks student progress, performance and test results for the instructor.

3.4.2 The EduSpace Graphical User Interface

The EduSpace Graphical User Interface (GUI) weaves instruction components and development tools into a self-organizing visual information display. The EduSpace GUI represents complex data hierarchies from an intrinsic perspective to allow a coherent view of the organization and relationship of courseware components and elements. This representation allows both interactive viewing and rearrangement of components, and immediate access to development tools.

The EduSpace GUI metaphor: Using EduSpace tools to assemble and sequence lessons can be seen as participating in the growth of an information tree. The tree is self-organizing: it knows what form to grow in while maintaining instruction component relationships. The designer gives substance to the tree (content information). As branches (new components) in the form become useful, the tree grows out in front of the designer, indicating what kind of information to add, until each branch is whole and the tree is mature.

The tree metaphor describes both EduSpace software functionality and of its GUI. Each lesson is a separate software entity in which instruction components and elements are hierarchically configured. The graphical form of the display is functionally representative, serving both as the lesson description and as the lesson entity specification.

The self-organizing graphical components can be viewed in 2-D and 3-D forms. The graphical form of these rings is arbitrary, and can appear as nested squares, cubes, circles, spheres or any other boundary shape.

In the 2-D *Lesson Display*, the lesson component hierarchy, or tree, is viewed from an "inside" perspective, like looking at the cross-section of a tree trunk. Components are displayed as nested rings of the tree. Rings inside a larger ring indicate other branches of the tree. Two ways of representing the same lesson display of the hierarchical relationship of EduSpace lesson components are depicted in Figure 3-6.

By representing the information hierarchy intrinsically, as it would be seen from the inside of a tree (elements of a single entity) rather than the outside (as separate branches), the designer can see in one graphical display all of the relationships between lesson components as well as the relationship of each component to the overall structure of the lesson.

This representation also allows direct access to nested components. The outer ring, or bark layer, of the component tree (labeled *LESSON*) is the top level of each lesson, and contains all other components. Using the buttons to the right of the display (some of which trigger pop-up menus choices), the designer can *open* and *close* rings to view relationships between lesson components. Each component can be opened independently.

By moving *in* and *out* of rings, the designer changes levels visually (the selected ring and the components nested within it expand to fill the screen). Pertinent icons for RuleBuilder, TrainTrack, ProfileBuilder, or MediaMaster tools are displayed. When a component is opened, SpecBoxes or SpecGraphs appear.

SpecBoxes are text based specification dialogue boxes. They are used to create activity modules by specifying the attributes and behaviors of entities. The designer creates these activity entities by linking together selected elements from the EduSpace Knowledge Base and from the EduSpace Media Library. New elements can be added to the knowledge base and media library using MediaMaster and RuleBuilder.

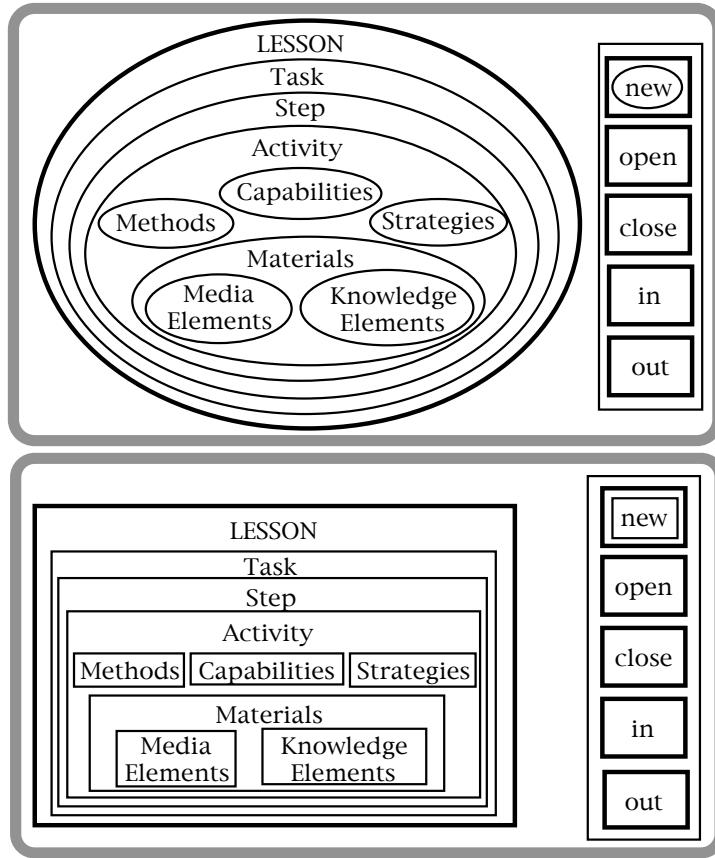


Figure 3-6: EduSpace Lesson Organization Display

Every TaskMaster lesson component has its own kind of SpecBox. SpecBoxes are constrained for categories of input and appropriately formatted for error-resistant entry of the different elements of information that are required to develop a lesson. Each SpecBox has a communications line for interactive dialogue between the designer and the system.

SpecBoxes are dynamically linked so they can share elements of information, and all component SpecBoxes are linked with the EduSpace knowledge base. New entries are automatically entered in and updated across SpecBoxes and in the knowledge base.

SpecGraphs allow the *graphical* generation of rules, relations and attributes of entities. Inside of the inclusive environment, OASIIS provides interactive reconfiguration of entities (such as saying, "Scale down to 3 and turn red", or by holding the corners of a cube and pulling to stretch it to a larger size).

The designer can add instructional components using the *new* button to select from a pop-up menu of Instruction Component Types. Added components automatically nest in the appropriate location within the ringed configuration, which resizes to contain them. While components are constrained to particular nestings as a consistency-maintenance device, these levels can be changed by expert users. The organizational

structure of the interface allows more efficient management and retrieval of lesson components, and optional methods of activity specification.

3.5 THE EduSpace EXPERIENCE

This section describes the EduSpace interaction process for each kind of end-user: the instructional designer, the training instructor and the training participant.

3.5.1 The Instructional Designer's Experience

In this example, a single instructional designer, who is not an engine maintenance specialist, has the assignment is to create a series of lessons that will train engine maintenance workers. His industrial employers have provided the following resources: a 3-D CAD database of the engines their trainees will be maintaining, a list of general performance requirements, the manuals currently used by the company for training, the names of company maintenance experts...and an EduSpace IDW-1 system.

Knowledge Acquisition

The designer's first job is to enter training-specific information into the knowledge base: he digitizes illustrative pages from the manual and adds them, along with the CAD data and new technical vocabulary, into the EduSpace knowledge base. He goes to the site and videotapes observations of experts on the job, later transferring them to videodisk. Those experts can also advise him on what the "tricky" areas are for novices and what job skills are most valuable, providing insights into lesson emphasis.

When the knowledge acquisition process is complete, the designer has a collection of teaching materials and a general understanding of the training domain. With subject matter texts nearby for reference, he is ready to begin transforming the information from books, manuals and the minds of experts into memorable training experiences.

Starting a New Lesson

He slides into the EduSpace Station, brings the head-mounted display within easy reach, and adjusts the ergonomically designed chair to a comfortable fit. He selects the OASIS interactivity options (hand-tracking, speech input, object frames, wand) that he will use during this session, and turns his attention to the monitor. He opens the TaskMaster tool, and chooses *New Course* from the top level file menu, bringing up the *New Course SpecBox*.

The New Course SpecBox elicits the course name and identification, description, prerequisites, and a description of the target student for whom the course is intended. The designer enters the name of his course, "Engine Maintenance" and specific identification information, then completes the form-based specifications.

In another field of the New Course SpecBox, the designer can enter a sequential list of the lessons in the course. From this list he can dynamically rearrange lesson order, or he can open a lesson. He names Lesson 1 "Carburetor Maintenance" and opens it. The *Lesson Display* opens.

Defining Lesson Objectives

The designer chooses to move in to *LESSON*. He sees a bounded field that contains one unlabeled task component and the Objectives SpecBox icon. He opens the Objectives SpecBox and is prompted to specify the performance objectives of the lesson, stating exactly what skills the student will be able to perform after completing training. This formal specification of objectives allows principled lesson design and training progress assessment.

The designer selects activity elements (behavior descriptors, objects and relationships) from the EduSpace knowledge base. If any of the words that the designer uses are not recognized by the knowledge base, the RuleBuilder icon will appear. By opening this tool, the designer can add new vocabulary elements, objects and relationships to the knowledge base.

He enters performance elements into one of several natural-language relational statement formats. He can also construct new relational formats. An example of a simple relationship:

| FORMAT | <i>behavior:</i> | <i>object:</i> | <i>relation:</i> | <i>object:</i> |
|--------------------|------------------|----------------|------------------|------------------------------------|
| <i>Objective 1</i> | | | disassemble | the parts of the carburetor |
| <i>Objective 2</i> | | | replace | the broken parts of the carburator |
| <i>Objective 3</i> | | | reassemble | the parts of the carburetor |

Performance standards are also specified in the Objectives SpecBox. Speed, accuracy and general proficiency requirements can be stated for each objective.

Specifying Training Tasks

For each performance objective, a *Task* component is automatically created, labeled and represented as a ring nested within the *LESSON* boundary. The three objectives specified by the designer for Lesson 1 appear as components in the display.

The designer goes in to Task 1, and opens the Task Analysis SpecBox. The performance elements that he specified at the *LESSON* level are already entered in the box, to serve as a starting point for the task analysis procedure. The next process for the designer is to specify the steps required to perform the task, using the same relational format as before:

behavior: *object:* *relation:* *object:*

Task 1: objective disassemble the parts of the carburetor

- Step 1: know the parts of the carburetor*
- Step 2: know the parts which join other parts*
- Step 3: know the tools to use for each part*
- Step 4: use the tools to disassemble the carburetor*

Configuring Particular Activities

The elements of the lesson, the steps and their sequence have now been specified. An Activity component is automatically created for each Step. The designer moves in to Step 1.

The Step boundary expands and displays an Activity SpecBox and an unlabeled Activity component. The designer opens the Activity SpecBox, then names four different activities that will be interactively sequenced through the behavior of each learner in achieving the goals of Step 1.

Figure 3-7 illustrates the way that the designer's Lesson Display would look if each component were open at the top level.

He moves in to Activity 1, which he has called "learn names: ". It contains the four components of an activity: Methods, Materials, Capabilities, and Rules. He goes in to the Methods component. As one option in the set of activities to help students "name carburetor parts", he decides to show students a diagram of the carburetor with the parts labeled with their names. He wants to let the students turn off the name, enter the name themselves, and have the label respond to correct and incorrect names. He wants to let them test their memory of all the parts, and finally to pass a formal vocabulary test.

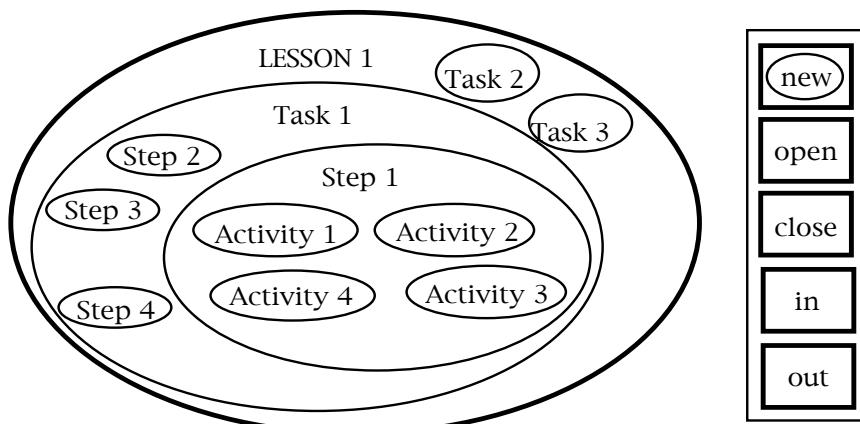


Figure 3-7: Lesson Display During Development

Organizing Media Elements

While some media elements are being constructed by corporate graphics designers for use in the courseware, he has learned to configure some of the graphics himself. He envisions this particular activity as a multimedia element, and knows he has a graphics screen to configure. He decides to do that first, goes to the top-level MediaMaster icon and launches it. A graphics window opens to display several base-functionality options, including video editing, VR construction, screen construction, and animation editing tools. He selects the dynamic screen construction tool, using it to create a new media element defined as a screen with text and graphics fields.

He opens the Media Library and imports a labeled carburetor diagram from the digitized manual pages. He arranges fields around the labels so the part names can be made to disappear and new names can be entered in. He also imports to a second media element file the 3-D CAD drawing, for use in an animation for Activity 2.

Specifying Teaching Methods

The designer opens theMethods SpecBox to choose the teaching methods for this activity: description and interaction. When he selects *description*, a dialogue box opens asking,

"Describe what?"

He enters: parts of motor.

The box responds: *"Lesson topic: parts of carburetor. Replace 'motor' with 'carburetor'?"*

The designer sees his error and OK's the change. When he specifies interaction, he is asked,

"Interact with ?"

He specifies: parts names.

"Interact how?"

He enters: turn off part names; enter part names.

Assigning Interactivity Options

He closes Methods and opens Capabilities. The Capabilities SpecBox already has three capabilities checked and specified, from the information shared with the Methods SpecBox:

see: carburetor part names

select: part names

enter: part names

But, as he views his other options, the designer decides that hearing the names as well as seeing them would be an appropriate strategy, so he also checks,

hear:

A dialogue box opens: "Hear what?"
 The designer responds, hear: part names.

A MediaMaster icon appears & prompts him with a blinking
 "?"

He opens the MediaMaster tool, and the communication line holds the following message:

"No such entity: hear parts names. Add?"

The designer can use the MIDI tools in MediaMaster to speak and record each name and associate it with the correct part, or he can import an existing voice file. He decides to wait for this step until he is configuring the Media SpecBox. The "hear" capability puts itself on "off-waiting" status; if the media element "hear part names" is added to the Media Library, it will turn on; if no module appears, it will remain off. He closes MediaMaster and continues to configure the Capabilities SpecBox, where he is asked to specify an I/O device for each capability.

Building Interaction Rules

The designer is now ready to specify the Strategy for Activity 1. Going in to Strategy opens the Strategy SpecBox which requests the interactive rules and procedures for the monitor-coach-tutor module: entity rules (eg., How does this display respond to student behavior?) and teaching strategy (eg., When a student has asked for coaching, how will they be notified?). He begins by building new entity rules using RuleBuilder, which allows entry and testing of behaviors not already in the knowledge base.

An important feature of the entity rule building process is the Test Window that appears during the editing process, which allows entity behaviors to be viewed as the edits are made.

Using the object and interaction elements that he previously entered in the SpecBox, he chooses rule formats that will turn off part names when they are selected, check new entries for correctness, and flash the name three times if correct.

Then he specifies the coaching error notification:

FORMAT: *If [],
 Then []*

If the entry does not match the name,
 Then scale the name to 0.

He sees the name in a small field on the Test Window, and selects it. The name turns off, and he enters an incorrect name in the field. Nothing happens. He reads his rule again, then changes it:

If the entry does not match the name,
Then scale the entry to 0.

When he tests the behavior again, the incorrect word shrinks away into nothing. He enters the correct name into the field and it triggers the rule he specified earlier: the name flashes three times.

Building Behaviors with Graphs

A Behavior Graph both represents and defines a relationship between any two entity attributes. By manipulating the data in the graph, the dynamics of the behavior are defined. This graphical form of entity specification is an alternative to the propositional statement format.

The designer now decides that he will model some of the basic dynamic processes of the carburetor. To illustrate the flow of gas and air, the designer creates five new materials elements. He grabs a copy of the float chamber, the needle valve, the choke valve, and the throttle valve from the converted CAD data. He places each one in a different materials element. He also labels the fifth materials element "fuel". He opens the RuleBuilder inside of each element, and sees an unspecified SpecGraph.

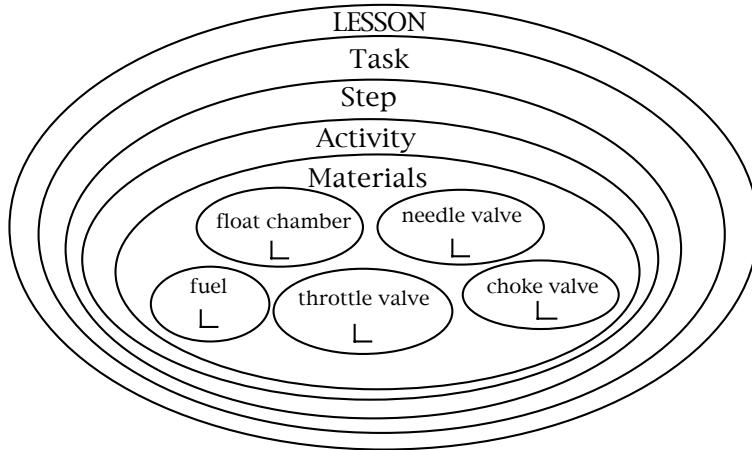


Figure 3-8: Lesson Display of SpecGraphs

The designer labels the horizontal axis by selecting the entity and the attribute which will instigate the behavior. He also labels the vertical axis with an attribute of the element. This attribute will display the effects of the behavior. At the system level, a SpecGraph implements a series of conditional match and substitute operations.

Now it is time for the designer to construct a virtual carburetor. To do so, he defines the element behaviors with the following SpecGraphs:

The SpecGraphs in Figure 3-9 implement the stasis mechanism of the float chamber and needle valve, which adjust to maintain the fuel level under constant drain.

The SpecGraphs in Figure 3-10 implement the creation and movement of fuel mixture as a result of air flow. The measure of air flow is a variable parameter which the designer can reuse in his next course, "throttle mechanics". In this lesson, it will form the base for the dynamic disassembly required in Task 1.

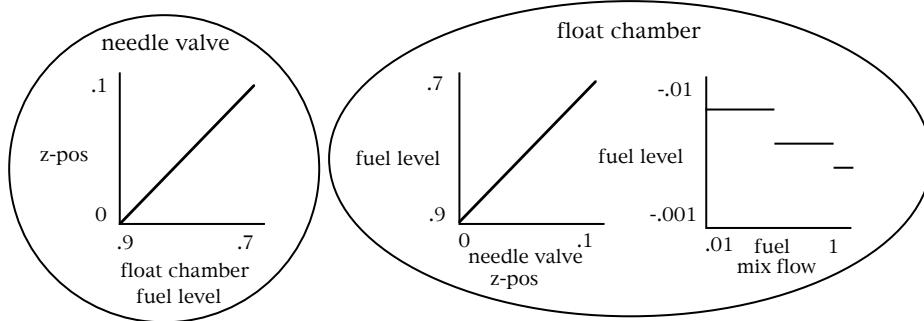


Figure 3-9: SpecGraphs for a Carburetor

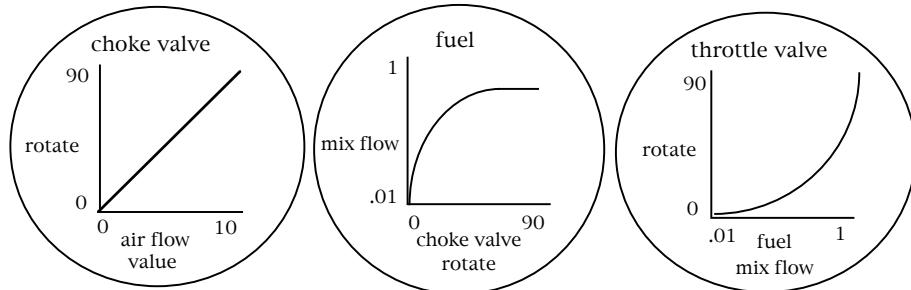


Figure 3-10: SpecGraphs for Fuel Flow

Finishing A Lesson

The designer continues to specify the interaction process and entity behaviors of Activity 1, then moves to Activity 2. He falls into the rhythm of the process over the next few days, moving in and out of component levels to flesh out the form of each lesson, guided by the TaskMaster tools.

He uses the MediaMaster and RuleBuilder to construct the context and object behaviors for the final carburator assembly and functional simulation exam, importing frames

from the site videos as texture maps of the actual working environment. He can use these again in future training courses for the company.

When the SpecBoxes in each component of the lesson have been configured and tested, and the lessons are arranged in the order of presentation, the course is ready for a final run-through. He takes the course himself, and he finds that he learns more than he ever wanted to know about carburetors. Now the course is complete. Or, so he thinks.

Two days before the first training session is to begin, an engine parts update notice arrives on his desk. He turns to EduSpace, accesses the company's CAD data base and downloads the new parts specifications into the EduSpace knowledge base. He adds new relationship descriptions to associate with the new graphics, and does a *Universal Replace* on all references to the old parts. He edits the part names in the diagram and in the flow animation, and then tests the new course iteration for continuity. He has finished the course update, just in time.

3.5.2 The Trainer's Experience

The trainer's EduSpace Developer Station is networked with 15 Presenter Stations in the training room and 5 more Presenter stations at a site across town. Her 20 students are scheduled to log on in an hour, and she begins to prepare for the session.

Setting Up the Session

She opens the TrainTrack tool and selects *Lesson Information*. She scans the Course Outline provided by the instructional designer and finds that while she has taught this subject matter before, there are several new activities in these lessons. She opens the *Lesson Display* and moves directly to the new activities to review them.

Then the trainer opens *Session Information* and proceeds to follow the designer's configuration specifications. She selects *System Communications* and initiates its SpecBox, where she turns on Presenter Station monitors, mouse and keyboard (for screen-based interactions), selects OASIIS wand and frame functions (for 3-D interactions), and turns on the boom-mounted displays and earphones.

She next sets up her *Monitoring Window*. She places twenty two-inch square *WorldWindows* in a grid on the lower three-quarters of his monitoring screen. *WorldWindows* will display the active screen of each student during the session, and allow her to observe the progress of the whole class, or to expand and view a particular group of student screens. The screens of remote students are distinctively clustered.

She locates her on-line *Student Communications Window* at the top right corner, and her *Systems Readouts* on the left. She can communicate with any or all of the students by

selecting their screens and typing in the communications window, or speaking into the microphone. She can participate in an individual student's virtual world or in a group world with her own boom-mounted display and OASIIS tools.

She opens *Student Information*, transfers the file of registered students into the Administration SpecBox. She opens the Student Profile information, and copies it into ProfileBuilder. When she compares the group profile to the Target Student description, she notices that two of the enrolled students have not completed the course prerequisites. She loads the "Maintenance Safety" Lesson on their stations; they'll have to pass that before they can join the others in the "Engine Maintenance" Course.

She also notes that two recently transferred employees have been working in engine maintenance for years, and just need experience with the specific type of engine they will be maintaining for this company. She sets their courseware to start at Lesson 6. Students begin to log on to their stations; their screens blink to life in the trainer's WorldWindows. The trainer thinks that Activity 1 (learn part names) looks a little dull. She decides give the Lesson 1 students some time to practice the parts names activity individually and then group them into on-line teams for some friendly competition in naming parts.

Monitoring the Session

The session begins. She sees three *Instructor Flags* already; she selects the screens of the trainees who need help, sees what their problem is and points them toward a solution path. She gets a System Readout indicating that there is a lag in the remote-site communications; she opens a talk window to their system administrator and he deals with it. Two more Instructor Flags alert her to trainees who are stuck at the same point in Activity 2; she links them and encourages them to work on a solution together.

Another Flag; one of the advanced students is ready to demonstrate his new expertise with carburetors for Mastery Certification. The instructor opens this student's WorldWindow, moves her inclusive display into place, and enters the student's virtual world to see how well he accomplishes his task. After the performance demonstration, the instructor returns to the Monitoring Window, reviews the WorldWindow display to check the progress of the Lesson 1 students, and starts to organize their group naming exercise. And so it goes...

Evaluating the Session

When the session ends that afternoon, the trainer checks the *Cumulative Accomplishment* display, where a 3-D representation of the Lesson Trees represents student achievements and frustrations as graphical information. The branches (lesson components) show a green leaf on each tip that a student has completed successfully, a wilted brown leaf for failed attempts, and a red apple for Mastery Certification. She sees that her grove is thriving, although one activity has a disproportionate number of brown leaves, and she makes a note to monitor this activity more closely.

She powers down the EduSpace System for the day and is about to shut down the IDW-1 Station, when one of the instructional designers comes up to see if she wants to test out his new underwater training environment. They eventually leave when people start to arrive for the evening's EduSpace international corporate training seminar.

3.5.3 The Participant's Experience

"...so then my group set a speed record in getting the answers, and I went on to try different things with this animation of how the gas and the air mix -- and I could hear how the engine would sound with different amounts of gas. Then I got the answers right about how that works and I decided to look at this video of our maintenance pro Sam doing a demo on valve adjustment. He has some good tricks.

"No, that was just the morning. Then we had a VR part -- VR, getting inside computer worlds...through those headsets. Four other people in my class were in there, and we could hand tools to each other and talk about how to put this to that and why some things didn't work. We did it great when we were together -- but when I went to fix it by myself, I forgot this one thing. So, the instructor talked to me in my headphones, clear from the other building, and gave me some ideas to try and they worked, and then I passed that test, too.

"So, I finished two lessons today. They say when you're done with the course there's this wild mastery exam, like the Star Tours ride... Well, you get to go in VR and put your motor that you've fixed right into the car and then be the test driver. Sometimes the whole class goes along and if it works you pass that part, but if the thing fails, you crash with everyone watching, or, well, I don't know...sure, I'll tell you about it..."

"Yeah, you're right, things really have changed...a flipchart and a manual, uh-huh...well, I'd probably sleep through that...OK, talk to you next week, Mom...Bye."

4. MARKETING and SALES

In this section, we discuss the training market and the current competition. We then present the Oz competitive advantages and the market strategy.

4.1 THE MARKET

The formal corporate training market has steadily grown as industry and society realize the questionable and outdated education received by most Americans from our school systems. Our companies are being out-produced by foreign competition that can bring new and innovative products to market faster. There are many reasons for this, but the one basic cause is our inability to respond to the training required for the complex and ever-changing technology that drives these new products. In response, corporate training budgets have grown. In 1991, domestic companies with ten thousand or more employees spent \$3.2 billion net for training (not including the wages of those being trained). In aggregate, *Training* magazine has estimated that domestic companies spent \$43.2 billion in 1991 for formal training. This number includes \$8.7 billion spent on outside vendors for hardware, off-the-shelf and custom training products, and other outside services. (See Figure 4-1.) These outside expenditures are the targeted service market for Oz.

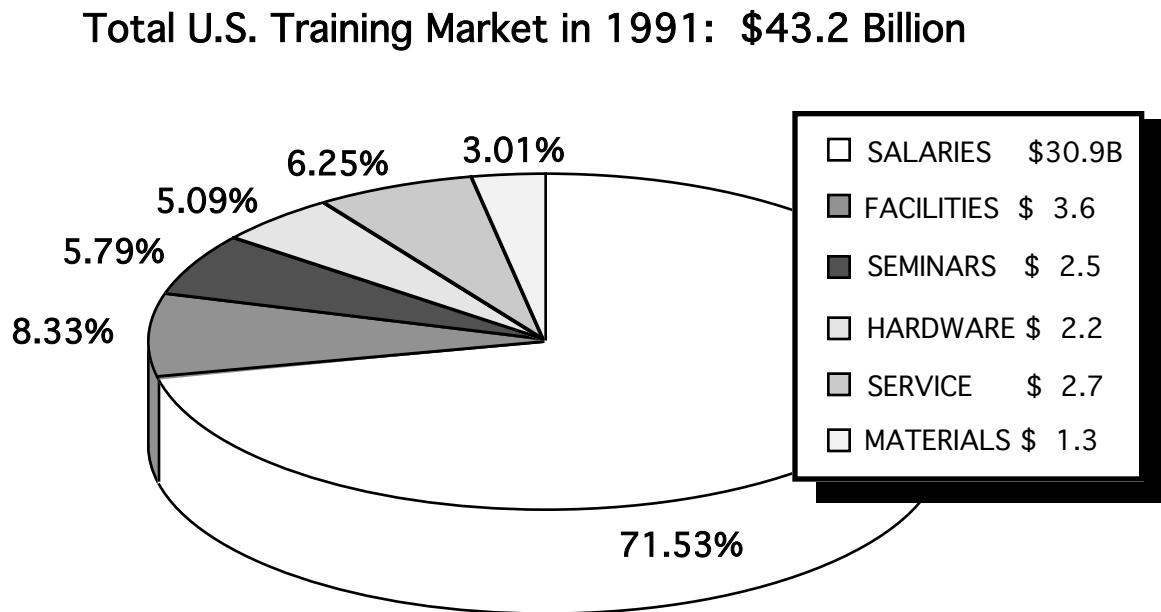


Figure 4-1: 1991 U.S. Training Market

In addition, the U.S. Department of Defense expenditure for training for the period ending fiscal 1990 was \$18.3 billion. Oz will be aiming at this market through established suppliers as an O.E.M. initially and will reevaluate DOD sales distribution at a later date.

While specific figures for foreign training expenditures are not currently available, Oz intends to become an exporter of its training products. The VR capability is especially transportable to foreign markets as there is no need to translate "actual" work experiences into a foreign language. Foreign revenues in maturing software application companies compose between 25% to over 50% of their total revenues. Clearly, foreign source revenues offer great potential to Oz.

The cost of training per employee varies in terms of the source and quality of training, but the value of training is rarely questioned. For example, a recent survey in *Training* magazine showed that a third of domestic corporate management felt training increased in value during the last recession, while only 6% of management felt it decreased. Another survey showed that during the recession in 1991, 40% of human resource executives received increases in their training budgets, while only 16% received decreases. This same survey also revealed that as domestic quality control concerns have grown, production workers in domestic corporations of 100 or more employees now receive an average of 37 hours of training a year, second only to that of salespeople at 39 hours per-year average. *The training market place is durable under all economic conditions* and will receive additional national attention as domestic emphasis on quality improvement in production expands and product life cycles shorten.

It is important to note that Oz will be entering the market with an evolutionary product for the industry. Therefore, trying to predict the actual size of the Oz market is difficult prior to the introduction of the product. We do expect good market acceptance based on early market surveys. Perhaps the best inference of market potential comes from the simulation market. FlightSafety International, a public company engaged in providing flight, marine, and power plant training, simulation services and software, reported \$268 million in revenues in its last fiscal year. The ability of the Oz product to service a much broader range of corporate activities including manufacturing, gives Oz an even larger potential revenue base than the simulation marketplace.

Jostens Learning Corporation (a division of Jostens, Inc.), a supplier of integrated learning systems for the education market, has shown an average growth of over 30% per annum since 1987 going from a reported \$39 million in sales to \$160 million reported in 1991. It is apparent that the training marketplace is growing substantively.

4.2 THE COMPETITION

4.2.1 Computer-based Training Vendors

These are the vendors who supply the training market with part of the estimated \$8.7 billion in outside products and services. "Outside" vendors is an appropriate term in that most large companies (10,000 or more employees) have internal training departments. In fact, the bulk of training dollars (an estimated \$30.9 billion) goes for the salaries and fringes of these internal training staffs. As a consequence, the first line of potential "competitors" are the very internal training staffs that Oz targets as clients. However, it should be noted that in 1991 it is estimated that 82% of these companies used computers in training and 61% were also doing computer-based training. Because one of Oz's missions is to empower trainers to be more productive, there exists a natural market for Oz to sell into these internal training staffs. This is especially true for the 61% of companies that already do computer-based training, as the Oz product can be positioned as an extension of their already existing computer-based training efforts (See Section 4.4.1: Economic Advantages).

Outside training and educational computer vendors can be classified into three broad classes by revenues. The first of these categories we will call "M&P" suppliers for "mom and pop" suppliers. The term M&P is not meant to be an indication of the professionalism of these suppliers, but simply means their revenues are under \$5 million per annum. In fact, most of these suppliers have revenues under \$1 million per annum and fewer than six employees.

These suppliers are the creators of the off-the-shelf and, in some cases, customized software packages used in training and education. There are dozens of these M&P vendors, but their products have a distinct flavor to them. They are almost all serving vertical markets. By far, their largest market is software used to train users of well-known software applications such as WordPerfect, Lotus 1-2-3, etc. Other vertical markets such as medical training and school or training management packages are also prevalent. Some of these companies are also the developers of *Authorware* products. *Authorware* products are software programs currently used by trainers to develop training sessions.

While taking their bite out of the market, the M&P suppliers cannot be considered head-on competitors with Oz. First, they are not system integrators. Second, they have concentrated on using traditional software development methods which limit the interactivity of their products in training. None of the M&P vendors appears to have the technical breadth necessary to develop a VR product.

The second category contains companies with revenues between \$5 million and \$100 million. Many of these companies are more in the educational markets as opposed to the training markets. As a result, we will refer to them as "MidEd" companies.

MidEd companies are also often divisions of or subsidiaries of other companies. Many are the divisional companies of the large publishing companies. Some of these, in turn, are now owned by other companies; i.e., Simon & Schuster is a Paramount Communications company and Harcourt Brace Jovanovich, Inc. is a General Cinema company. Among these divisions are Houghton Mifflin Software Division, Academic Press, Inc., American School Publishers, and Crwth Computer Courseware. Some MidEd companies are the subsidiaries of other companies. Among the subsidiaries is

Jostens Learning Corp., a unit of Jostens, Inc. There are other stand alone companies in the MidEd category such as Broderbund Software and Educational Development Center, Inc. which is a non-profit organization.

Most MidEd companies have focused more on the educational markets to date. However, there is nothing stopping them from entering into the software side of the training business in a more aggressive way. The one exception to this statement are the interactive video suppliers such as Industrial Training Corp. ("ITC"). ITC is a true multimedia training supplier. The company sells preprogrammed interactive video disks and the software required to run them on a PC. The PCs needed to run these programs must be equipped with special video boards, often require touch screens, and must be attached to specific video players. The programs themselves are generic in nature and little to no customization is possible. ITC has developed a large library of these generic interactive disks that includes such titles as *Interpersonal Skills*, *Pipe Fitting*, and *Electric Motors*.

Once again, most of the MidEd companies are not system integrators, nor are they necessarily even software developers as they may purchase software from other development companies. In order to solve the development problem, some of these companies, such as Jostens Learning, have formed or bought into software developers. The divisional companies of the large publishers are clearly electronic extensions of the publishing business and not hotbeds of research and development. These divisions are generally lost in the financial consolidations with the parent companies and do not supply enough revenues to warrant the research and development dollars necessary to build a VR system. Similarly, the other MidEd companies are more interested in fulfilling the traditional software publisher's role. They generally distribute programs on a royalty basis for the original developer, although they do some internal development as well. Again, their revenue bases do not allow for large R&D projects.

Finally, there are the computer hardware vendors. These vendors include all the well known names such as IBM, NEC, Commodore, and Apple. In addition to computer hardware, much of the outside hardware expenditures go for non-computer equipment such as overhead projectors, copiers, LCD panels, white boards, video players and recorders, CD players, and data/video projectors. The suppliers of this type of equipment include such well known names as Sony, Mitsubishi and Panasonic. Some of these companies, such as IBM, will bundle software from outside vendors with the computer or multimedia equipment. In fact, many of the computer systems sold are now positioned as "multimedia solutions".

These hardware companies may represent the source of any true long-term competition to Oz. They have the financial muscle to take on large research and development projects. They have the incentive to develop software which will drive hardware sales. In fact, the concept of a multimedia solution to training is much closer to the Oz concept than anything else on the market.

The only other commercial possibility for competition now known could come from companies like Microsoft. There are several reasons as to why these companies are not a near term threat. One is the current lack of computing power in the PC world. Several more generations of PCs will have to be created to reach the computing power

necessary to drive an inclusive VR system on a PC. Next, the training market is a vertical market to the large software publishers. It would be a market more currently addressed by VARs and other developers using their development tools for existing applications.

4.2.2 Virtual Reality Vendors

The current state of the virtual reality marketplace consists more of speculation than actual dollars. The popular press has over-blown the near-term potential of virtual reality beyond present capability. However, there does exist a growing body of virtual reality companies. The current marketplace is best described as technology driven and not market (solution) driven. It has been estimated by *Success* magazine that the best known of the virtual reality companies, VPL Research, Inc., has sold an estimated 1,200 systems over the life of the company. Most of these systems were sold to research facilities, i.e., they were sold for research and development and not to find a solution to a market-driven problem. These systems ranged in price between \$10,000 and \$250,000. The research market is limited, but continues to grow.

In addition to turnkey virtual reality systems, VPL Research along with other companies such as Ascension Technology Corp., EXOS Inc., Fake Space Labs, LEEP Systems Inc., Logitech, Polhemus Inc., SimGraphics Engineering Corp., Shooting Star Technologies, and Virtual Technologies make input/output devices and support equipment for virtual reality systems. Included in this equipment are head-mount displays, booms, data gloves, full body suits, and tracking devices. The 6-D tracking and pointing devices manufactured by Logitech, Polhemus, Shooting Star Technologies, and Ascension are direct competitors of Oz's OASIIIS product (along with the more traditional input devices such as the keyboard and mouse). Oz initially will rely on the equipment produced by some of these manufacturers in the integration of its products.

Perhaps the best known commercial success of a "virtual reality product" was Mattel's Power Glove for the Nintendo Home Entertainment system. Mattel sold \$40 million of the Power Gloves during the first year they were on the market. More than 500,000 of the Power Gloves shipped within weeks of the product's release. Today, most of those gloves are gathering dust in closets. Mattel failed to follow through with the game software necessary to take advantage of the 3-D input device. Nevertheless, from the point of view of the product as a toy, its market life and sales volume made it a success.

The above observation of the Power Glove points out that one market solution for virtual reality is an entertainment product. Most of the work in this area is currently being done at the arcade level. Horizon Entertainment (a division of Edison Brothers, a \$1.4 billion NYSE specialty retailer) distributes a virtual reality game called "Dactyl Nightmare". W. Industries (out of the UK) manufactures several virtual reality arcade games and has joined with Horizon Entertainment and Sphere Software to form CyberStudio which is to design the software for additional virtual reality arcade games. Lucas Arts and Hughes Aircraft have teamed up to work on virtual reality theme-park rides. In addition, major Hollywood companies such as Disney and MCA are rumored to be working on using virtual reality in theme-park-type rides.

Outside of entertainment, the lure of virtual reality remains strong. Companies such as American Express, Nynex, Microsoft, Intel, Digital Equipment Corporation, Sun Microsystems, Silicon Graphics, IBM, U.S. West Telecommunications, AT&T, Ford, Fujitsu, Hughes Aircraft, Boeing Corporation, John Fluke Manufacturing, Sharp, and EDS have engaged or are engaged with virtual reality research.

Companies with virtual reality software in the marketplace include , Bechtel Software Inc., Sense8 Corporation, Virtus, Perceptronics Inc., Reflectone Inc., VREAM Inc., Xtensity Inc., and Straylight Corporation. Autodesk has distributed the alpha version of it's Cyberspace software to beta test sites, and expects to enter the market in 1993.

Since most observers of virtual reality see the training market as a primary application area, it may be expected to see the manufacturers of large military simulators such as Evans & Sutherland, GE Aerospace, and Hughes Training Systems attempt a low-cost virtual reality training product in order to take advantage of the price efficiencies offered by workstation virtual reality and to open up new markets to combat the falling defense market. We also turned up several proposals by large companies for the Department of Defense that involved some form of VR to be utilized in a training environment.

4.3 Oz COMPETITIVE ADVANTAGES

Oz has many advantages over all of the existing virtual reality software companies and those who wish to join this marketplace. The Oz competitive advantages in the training market can only be assessed by the strength of the Oz technical and management plan, since no comparable advanced technology training products exist. The Oz advantages include:

- an exceptionally strong managerial and technical team
- a tightly integrated suite of products
- deep understanding of the architecture of modern computer-based training
- unique yet proven technical innovations
- powerful strategic alliances
- momentum and timing

One key advantage is the dedicated professional team that Oz has assembled to create the EduSpace products. Oz key personnel have rich and diversified managerial experience. But in addition, active partners and stockholders in Oz include nationally recognized leaders in computer-based training, in interface design, and in virtual environment technologies. Only a small cadre of computer scientists have experience in and understanding of virtual reality interface and programming techniques. An equally small number of educators have deep understanding of computer-based instruction. The Oz team leads the industry in the technical and managerial capabilities

necessary to bring EduSpace to the marketplace. No other companies in software or in training have comparable technical expertise.

Another key advantage is the Oz integrated suite of products. Oz has identified outstanding problems in the computer training market and has crafted a high performance product to address those needs. All large companies struggle with information which must be assembled and conveyed to new employees. The EduSpace product is able to provide *just-in-time situated learning* by incorporating exclusive innovations in computer-based training. The EduSpace Station is a seamless integration of the hardware, software, modeling, and interface tools needed to convey copious information to novices.

Oz has identified the necessary components of a computer-based training system for achieving learning. OASIIS provides accurate bodily position and speech information while the Interface Chair provides a comfortable, seat-of-the-pants interface to high quality multidimensional multisensory computer generated virtual environments. The EduSpace tools facilitate the intuitive design of compelling courseware, ease of student tracking and interaction, and memorable student experiences. Entity-based programming implemented in boundary mathematics provides a foundation for algebraic computation. The EduSpace Accelerator Board furnishes the computational power necessary to present sensually rich instructional material in realtime. Compared to existing competitors (Figure 4-2), Oz offers the only integrated solution.

Oz has proprietary exclusive licenses to key technologies needed for multimedia and virtual environment training. These technologies derive from work at Oz, at the University of Washington, at Stanford, and at the labs of strategic industrial partners (see Section 5.2.3: Corporate Affiliates and Strategic Relationships).

In particular, the EduSpace Accelerator Board is positioned to provide the performance necessary for complex multimedia and virtual world models and interactivity. The VE-Chip provides unique functionality for very efficient collision detection and spatial interactivity. The MS-Chip provides super-computer computational performance for customized EduSpace data.

| COMPANY | Interactive Software | Interactive Video | Rulebased Software | Multimedia Software | Hardware | Inclusive Virtual Reality |
|---------------------------|----------------------|-------------------|--------------------|---------------------|----------|---------------------------|
| Oz... International | Yes | Yes | Yes | Yes | Yes | Yes |
| Allen Communications | Yes | | | Yes | | |
| IBM | Yes | Yes | | Yes | Yes | |
| Industrial Training Corp. | | Yes | | | | |
| Jostens | Yes | | | Yes | | |
| VPL Research | Yes | | | | Yes | Yes |
| Autodesk | Yes | | | | | Yes |
| Sense8 Division, Inc | Yes | | | | Yes | Yes |

Figure 4-2: Competitive Product Features

Expressed in terms of cost benefits, the MS-Chip provides (roughly) 1000 function evaluations per second *for every dollar of retail cost*. Since multiple MS-chips can be easily connected (with logic driven by hardware or by entity-based software), the EduSpace accelerator offers scalable computing complexity for virtual environments. The cost-performance of the MS-Chip is roughly three orders of magnitude better in price-performance than the nearest comparable, the Cray I.

Oz is in a uniquely supportive environment to solve many technology and personnel acquisition problems that competitors must face. Oz has a close professional relationship with the Washington Technology Center, the University of Washington, and Stanford, allowing it access to outstanding experienced professionals. The HITL Industrial Consortium provides excellent connections to interested clients. As well as Oz and a dozen others, consortium members include Boeing, Ford, Microsoft, American Express, Fujitsu, Sharp, Digital Equipment, and Sun. The advanced development of transferable technologies at HITL has provided Oz with EduSpace product features several years ahead of competitors. These advanced prototypes serve as an established foundation for Oz software product components. The students employed at HITL to develop these software prototypes are available for hire by Oz immediately after graduation.

Oz has momentum and timing. Oz has been incorporated for over a year, and has carefully assembled the knowledge, technologies, personnel, and business plan to assure competitive success. The country agrees that it must address its educational dilemma immediately, and industry will insist upon it, since industry is paying a tremendous price for on-the-job training. Although high technology has not penetrated classrooms, recent spectacular successes of military training by simulation will drive a thriving research market over the next few years. EduSpace is targeted for marketing within a window that forecasts explosive growth in the need for training simulators. *Oz is in position to define the training market.*

4.4 MARKETING STRATEGY

Oz's initial target core markets will be:

- domestic corporations with 10,000 or more employees engaged in manufacturing,
- existing large training courseware development corporations, and
- the Department of Defense.

Secondary priorities will be companies with between 2,500 to 9,999 employees in manufacturing, the health service industries, educational institutions, and foreign companies.

Large Domestic Corporations

Because EduSpace is an extension of previous computer-based technologies, its market place will be found where technology-based training methods have already taken root. In 1991 it was estimated that 82% of companies with 10,000 or more employees use computers in training, and 90% of these companies use video in training. Since 61% of these companies are also developing computer-based training internally, they are the best place to begin a marketing effort. The "port of entry" will be the trainers and training developers who see the advantage of having an Oz system for themselves and will help sell the overall concept.

Courseware Development Companies

Sales to corporations and publishers that develop commercial training courseware are also expected to be significant. At first glance, sales to large training developers may appear contradictory as these organizations offer internal competition for the outside training dollar. Nevertheless, these developers will already have made a large commitment to computer-based training programs and/or inter-active video training programs and will understand the power of the Oz system. Their creation of programming materials using EduSpace will stimulate sales of Presenter Stations and make it possible for the more timid buyer to experiment with fully-completed training programs before making the purchase decision.

Department of Defense

The Department of Defense has invested substantial resources in simulators for military training. The *1990 Military Manpower Training Report* estimated that the cost of formal training would exceed \$18.3 billion for that year. The new flexibility offered by the Oz products should be a big attraction to the Department of Defense. Again, the Department of Defense is an organization that has an active investment and existing

understanding of the power of high-tech training aids which make it another fertile marketing ground.

Additionally, Oz will address vertical market segments which include high risk industries, high tech industries and assembly line industries. In high risk industries, effective training is critical to life, such as building a rocket for NASA or producing life-support systems for medical centers. In high tech, product cycles are short, and therefore re-training cycles must be short in order to get the product to market in a timely fashion. In assembly line industries, retraining is ongoing because new methods are constantly being incorporated. Oz products clearly address these needs.

Finally, as the reputation and success of EduSpace spreads, Oz will begin marketing efforts aimed at the smaller companies, educational institutions, and foreign territories.

4.5 ECONOMIC ADVANTAGES

Because the initial investment of a complete suite of Oz products is relatively expensive, Oz will stress the potential economic savings by the use of the products. The bulk of the dollars for formalized training in industry, an estimated \$30.9 billion in 1991, was for training *and production* staff salaries. Also, the estimated wages of employees while in formal and informal training has been put at \$150 billion per annum! Without a doubt, a set of enhanced training tools, such as EduSpace which can increase training effectiveness while reducing cost, will be a highly sought after training solution.

The economic advantages to the Oz products can be placed into three broad categories:

- *Salary savings* of the design team, classroom instructor, and trainee,
- *Reduced instructional setup and reconfiguration costs*, and
- *Increased profits* by speeding up time to floor and product through put.

4.5.1 Salary Savings

While there are no actual existing measures of increased effectiveness for the Oz products (since the products are still in the development stage), there are strong inferences for the potential value of the Oz products to be found in studies on the use of interactive videodisc technology in training and education. Interactive videodisc technology could be called another form of computer-based training, since a computer is used to direct the operations of a videodisc player. Interactive video will also be available as a separate feature of EduSpace and will be integrated into the system through existing multimedia software packages. These existing packages are already beginning to show productivity improvements. As a result, the following discussion on increased effectiveness will apply directly to the Oz products.

The training world has divided video interactivity into three levels described below:

Level-1 Interactivity

The trainee utilizes a TV, videodisc player, and remote control unit. The trainee can only do chapter searches and picture stop operations on the video disc. Typical costs for development of one hour of non-broadcast video quality courseware run between \$20,000 and \$40,000.

Level-2 Interactivity

The videodisc player at this level comes with a built-in microprocessor with limited programability. The videodisc player can be programmed to perform simple playback sequences and respond to viewer choices during the program. Typical costs for developing one hour of level-2 courseware run between \$50,000 and \$100,000.

Level-3 Interactivity

At this level the videodisc player is controlled by an external microcomputer, and the trainee is able to interactively control the videodisc presentation through the external microcomputer. The trainee intermixes analog data from the videodisc player and digital data displayed on the computer's monitor. The typical cost for one hour of level-3 courseware is between \$100,000 and \$200,000 with an average of \$180,000 per hour.

At level-3, a single hour of training material will translate into 4 to 6 hours of actual training time. This expansion comes about because of the interactivity between the videodisc contents and the lesson software on the computer. The software directs the student through the lesson by asking questions, giving examinations on the material, and branching off to new areas of information as required by the student's progress. The student is free to browse the video information while learning.

EduSpace will permit easy creation of a level-3 program. Much of the cost of the level-3 program currently comes from the necessity of hiring or employing camera, sound, and edit crews for the production of the actual video footage. New software developments will allow EduSpace to integrate images from a variety of sources, including video footage shot directly by the instructional designer and product designs taken directly from an existing CAD database in digital form. These innovations will allow the instructional designer to skip the production crew cost, if desired, and realize enormous savings.

EduSpace will also take training interactivity to a new level. The introduction of virtual reality will create a completely inclusive interactivity, something only simulators can do today. This new level is already being referred to as "level-4" by instructional technologists. The use of virtual reality will also reduce any required video time in the

lesson as the VR environment will replace the video as a transmitter the information to be learned.

In an article by J.D. Fletcher in "Multimedia Review" (taken from a study prepared for the Department of Defense), the following findings were noted on the use of level-3 interactive videodisc training:

- The average amount of student time saved in the use of interactive videodisc training was 31%.
- Interactive videodisc instruction was equally effective for both knowledge and performance outcomes increasing comprehension and performance for the average trainee by 28%.
- Overall interactive videodisc instruction was more effective than conventional instruction increasing the average trainee's comprehension of the material by 38%.

Based on these findings, there is no reason not to believe that the inclusive virtual reality component of EduSpace will result in *additional* savings and increased comprehension by the average trainee.

What the above study does not address is instructional preparation time. There currently exist several well known authoring software programs for developing instructional lessons *which incorporate video*. None of these systems contain a rule based instructional guide built into the software as will EduSpace. These rules, along with the unique EduSpace graphical user interface, allow the computer to structure the lesson plan for the designer, creating development time efficiencies which are estimated to approach 50%.

From these observations, EduSpace will create a potential savings for each customer. Figure 4-3 presents a cost comparison based on the following assumptions:

- The instructional designer will not rely on professional crews to shoot the video portion (if any) of the knowledge base.
- The average hour cost of interactive video lesson today is \$180,000 which breaks into 5 hours of class time. Approximately one half of this cost is in the video production.
- The use of virtual reality will eliminate 30 minutes of required video time from the lesson.
- Factory workers receive an average of 37 hours of training per annum (*Training* magazine). In level-3 interactivity, this would require the creation of 7.4 hours of interactive video programming.
- The use of VR will increase comprehension of the lesson material by 33%, reducing the required training time by a like amount.

- We assume one thousand workers will be trained on a minimum configuration of one Instructional Development Workstation at a cost of \$150,000 and 20 Presenter Stations at a cost of \$15,000 each. The EduSpace system will be amortized over 5 years.
- The average salary of an industrial worker with annual fringes is \$37,500 per annum. The average salary of a classroom instructor with annual fringes is \$46,049 per annum.

Oz...International, Ltd. Program Development and Delivery Cost Analysis

| Cost Comparison: | Today's Cost | EduSpace Cost |
|---|--------------------|-------------------|
| Video Production & Instructional Design | \$1,332,000 | \$499,500 |
| Classroom Instructor Salary | 46,049 | 30,853 |
| Worker Salary for Training Time | 693,750 | 464,813 |
| Total Program Development & Delivery Cost | <u>\$2,071,799</u> | <u>\$995,165</u> |
| Total Savings | | \$1,076,634 |
| Per Cent Savings | | <u><u>52%</u></u> |

3 Year Cash Flow Analysis:

| | |
|--|-------------------|
| Initial Investment for 1 IDW-1 and 20 PS-1 | (\$450,000) |
| Year One savings (less equipment outlay) | \$626,634 |
| Year Two savings | \$244,134 |
| Year Three Savings | \$244,134 |
| Internal Rate of Return | <u><u>85%</u></u> |

9/17/92

Figure 4-3: Cost Comparison

4.5.2 Setup and Reconfiguration Costs

It is easy to recognize that tremendous savings have been achieved over the years by training pilots on \$15 million simulators rather than letting them fly the actual aircraft which cost many times more to built and operate than the simulators. (Crashes are also costless on a simulator compared to the actual thing.) This same type of recognition will

be given to Oz products which will reduce the cost of simulation itself through virtual reality.

For example, EduSpace is more flexible than the traditional simulator, since EduSpace can be more easily reconfigured for different training tasks *and* more easily reconfigured for different types of fidelity (physical, functional, cognitive) and different skill levels required by the individual trainee within the simulated task. Also, EduSpace is most intimately tied to the learner and not the physical simulation of the system being trained making it more suitable for enhancing cognitive skills. Finally, EduSpace holds the lesson digitally and any changes in the underlying processes being taught in the lesson can easily be implemented without having to worry about rebuilding an entire analog videodisc. These features will add cost savings which are industry specific and tied to the complexities of the simulation required to accomplish the training task.

4.5.3 Time to Floor

Another extremely important but industry specific savings will be found in the reduced time required to train workers and the increased effectiveness of the training. In the modern electronics industry where a product may have an 18 to 24 month life cycle, time literally is money and marketshare. By cutting back on the number of hours required to develop a training lesson and the time required to deliver that information to the worker, EduSpace will allow a manufacturers to get their products to market more quickly. Expected increases in cognitive retention should also result in less reworking. Together, these efficiencies will create the opportunity for the manufacturer to be first in the market place with a better product. As most profits are realized by capturing market share early, the manufacturer can reap enormous rewards and stay ahead of competitors.

4.5.4 Distribution

Distribution of the Oz products will be by direct sales to customers facilitated by sales representatives. A major effort will be made to hire sales people who come from both the training and engineering environments. Trainers will have had the actual experience of developing courseware and understand the trainer's needs and political position within the company. Engineers will be used in conjunction with manufacturing sales under a team concept because they bring actual production experience and the language of engineering production to the sales process. Engineers will be especially able to aid in the close of the sale by giving assurance to the "informal" decision makers. That is, the engineers will aid in convincing the production department of the power of EduSpace in training the production workers.

Sales representatives will be engaged on an independent contractor basis. These representatives will be the geographical eyes and ears of Oz and will target potential customers and individuals to contact within these companies for their respective territories. Once an appropriate contact is made, Oz sales people will take over the

selling effort. By engaging these sales reps, Oz will not have to spend overhead on regional sales offices.

4.5.5 Product Support

Oz will remain the customer contract for all hardware and software support for all of the Oz products. Because the software is proprietary, it is obvious that Ozwill have to develop the internal support teams necessary to give customers the backup required to trouble shoot any software problems. However, hardware support may be in appearance as well as actual. Oz will take the direct responsibility for maintaining its Instructional Development Workstations. The IDWs are the heart of the Oz product line and require the fullest attentions of Oz personnel to maintain the best customer relations. However, the Presenter Stations are not as critcal. While maintaining the customer contact, Oz will have several options for PS-1 maintenance, including calling the CPU manufacturer if the machine is still under warranty, subbing out the maintenance work to the original manufacturer, or doing the maintenance with Oz personnel.

4.6 COMMUNICATION PLAN

Oz's marketing department will be communicating with upper management and those in charge of corporate training through a comprehensive marketing communications policy. The objectives of the Oz communications policy are to:

1. Build an understanding of Oz's products at the top corporate level and among those in the training and development field.
2. Position Oz as a major player and emerging world leader in the area of training and development.
3. Set the overall corporate image in support of sales, and corporate identity in the public sector as well as Corporate America.
4. Develop support materials for
 - Corporate identity, including press kits, corporate brochure, letter head and signage.
 - Sales support materials, including sales literature, inquiry response packages, user manuals and technical manuals.
5. Through advertising reach out to top corporate management, trainers and decision makers.
6. Conduct a strong public relations program utilizing:
 - Articles in trade and business journals as well as the news media.
 - Service on high profile committees in organizations such as ASTD.

- Provide literature, video presentations and newsletters on Oz's positive impact on the corporate classroom.
- Secure speaking engagements for key Oz employees which provide great exposure for the company and its products.

4.7 CUSTOMER SERVICE

Oz realizes that service is equal to product in importance to customers, especially those which Oz will be targeting. On-site installation, training and support, and initial system training will be standard for Oz systems. In addition to thorough yet easy to comprehend user and maintenance manuals, Oz will maintain an 800 number for customer service during regular customer working hours, plus 24-hour emergency numbers which will allow Oz to respond to any crisis.

User groups will be formed and supported by regular newsletters, exchange of training ideas and software support. In addition new products will be tested through the user group members providing focus group attention to the new products while keeping our current customer base abreast of new techniques and technology.

Top flight response will be Oz's goal in keeping the products up and running in the field. The maintenance contract program and built-in profit margins allow for these high standards in customer service.

Oz UNIT SALES FORECAST

| Year End December 31, | 1992 | 1993 | 1994 | 1995 | 1996 |
|------------------------------|----------|--------------|--------------|---------------|---------------|
| Product, Unit Sales: | | | | | |
| OASIIS | 0 | 120 | 400 | 750 | 1000 |
| IDW-1 | 0 | 0 | 18 | 130 | 250 |
| Presentor Station | 0 | 0 | 15 | 500 | 1800 |
| Product, Sales Price: | | | | | |
| OASIIS | \$ 1,000 | \$ 1,000 | \$ 900 | \$ 850 | \$ 798 |
| IDW-1 | 0 | 0 | 149,750 | 149,750 | 139,750 |
| Presentor Station | 0 | 0 | 14,980 | 14,980 | 11,984 |
| Revenues: | | | | | |
| Product | \$ 0 | \$ 120,000 | \$ 3,280,200 | \$ 27,595,000 | \$ 57,306,700 |
| Contracts | | 1,300,000 | 1,000,000 | 1,500,000 | 0 |
| Service and support | 0 | 0 | 292,020 | 2,695,750 | 5,650,870 |
| Total Revenues | \$ 0 | \$ 1,420,000 | \$ 4,572,220 | \$ 31,790,750 | \$ 62,957,570 |

9/17/92

Figure 4-4: Oz Unit Sales Forecast

5. THE COMPANY

5.1 Oz CORPORATE DIRECTION

Oz...International Ltd. is a Washington Corporation incorporated in 1991 by George W. Cone. To date the company has received funding from private sources and from its founder.

Goals

Oz will use its primary intellectual properties (described in Section 6: EduSpace Technical Description),

OASIIS
The Interface Chair
EduSpace Accelerator Board
EduSpace Operating Shell
EduSpace Development Tools and Interface

in conjunction with Boundary Mathematics to establish itself as a major, international company, providing advanced computer-based training systems and software products to the education and training market.

Strategy

Oz will achieve its goals and objectives by implementing the following strategies:

- Provide, in a timely fashion, high quality products and service that not only increase the productivity of instructional designers but also assist the designer to deliver courseware of the highest quality. In the final analysis, the success of Oz must rest upon the successful job performance of those trained on the system.
- Build a strong management team that can provide the leadership for rapid growth in market, product development, and fiscal responsibility.
- Aim sales and distribution at the U.S market, initially building a solid distribution and service organization dedicated to the industrial and medical training market.
- Protect the company against changing markets and economic conditions by developing software products while procuring hardware items through O.E.M. relationships. Support product development with subcontracts for custom systems integration.

- Strive to be and stay a market driven company delivering quality products to the field of training and education.

5.2 PRODUCT DEVELOPMENT

Oz has defined the overall design concepts and software requirements for the EduSpace system. The basic license for the Boundary Mathematics has been obtained and a letter of understanding with the Washington Technology Center consummated. Both Silicon Engines and Matty Computer have entered into a letter of understanding with Oz to give exclusive rights for the use of their patents and proprietary technology in the education and training markets. Oz is in negotiation with HAG International, a major chair manufacturer, for an exclusive value added resaler (VAR) agreement to our market as well as the CAD market. Silicon Graphics has been contacted and initial agreement obtained for a VAR contract utilizing their computer line as the initial engine for the Instructional Design Workstation.

Oz is also currently in discussions with Fake Space Inc. and with Crystal River Inc. for hardware components for the Interface Chair's audio and video output. Fake Space manufactures the BOOM, developed for virtual environment display at NASA. Crystal River manufactures the Convolvotron, the leading three dimensional sound synthesizer.

Oz believes the OASIIS technology has application as a stand alone product to the CAD market as well as being a primary input subsystem to the EduSpace instructional workstations. Currently Oz is developing design specifications for OASIIS, and pursuing appropriate patent protection. It is anticipated that OASIIS will be a key revenue generator to the corporation both as a OEM product and in the form of license royalties.

Upon completion of funding, Oz will immediately begin product development. Oz intends to commence delivery of the first product, OASIIS, in approximately 12 months, followed with the Instructional Developers Workstation and Presentor Workstation in 18 months. The initial systems will contain the key software for a major product introduction. Oz plans to develop a series of software packages as well as distribute other synergistic software that will be sold as added features to the product. This software strategy is intended to keep the average sales price fairly high with solid profit margins.

Oz is committed to spend 5% - 7% of its revenue on product development. It will continue to be involved with the Washington Technology Center and the Human Interface Technology Lab, as a prime source of future product innovation. The product introduction schedule and a resource schedule are presented below.

5.2.1 Project Milestones

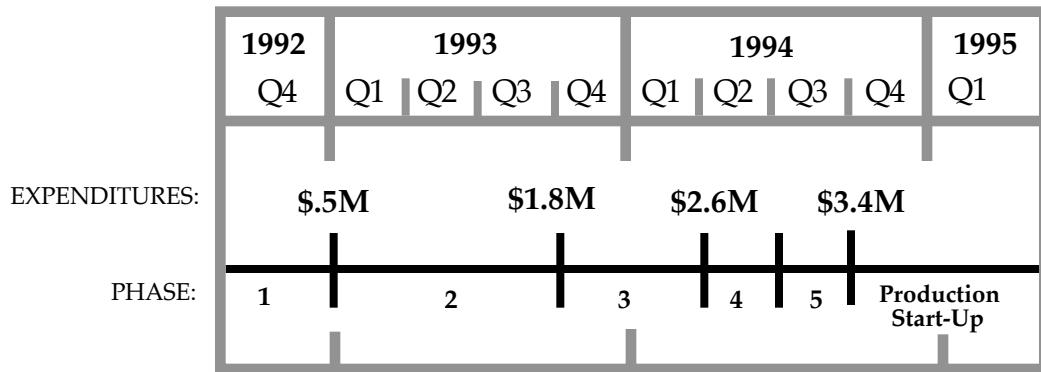


Figure 5-1: Product Development Phases

- Phase 1:** Generate hardware and software system specifications
Write Oz Business Plan
Acquire funding
- Phase 2:** Develop hardware and software for all sub-systems
OASIIS
Interface Chair
EduSpace Accelerator Board
EduSpace Courseware Development Tools
- Phase 3:** ASIC acquisition
System integration
Test and debug
- Phase 4:** Alpha Test
Bug corrections
Complete customer maintenance program
- Phase 5:** Customer beta test
Production start-up

5.2.2 Management Team

The company has set ambitious goals. In order to achieve them it has identified a team with the requisite technical, educational, marketing, manufacturing and financial skills, which also can provide solid leadership and management.

William Bricken - Technical Consultant

Dr. Bricken is the Principal Scientist at the Human Interface Technology Laboratory, where he is designing and implementing the Virtual Environment Operating System and the interactive tools of the VR environment. His prior positions include Director of Autodesk Research Lab, which developed the Cyberspace CAD prototype of virtual reality, and Principal Research Scientist at Advanced Decisions Systems, where he pioneered high-performance inference engines, visual programming systems, and instructable interfaces. Dr. Bricken holds a multidisciplinary Ph.D. in Research Methodology, Educational Psychology and Computer Science from Stanford, and degrees in Statistics (MS Stanford), Education (DipEd, Monash University, Australia), and Social Psychology (BA, UCLA). Before entering industry, Dr. Bricken was a Assistant Professor of Education at University of Hawaii and at Monash University. He specialized in General Methods of Teaching.

Dr. Bricken is an internationally recognized expert in architectures for virtual environments, is the inventor of Boundary Mathematics and void-based computation, and is responsible for developing over \$3 million of infrastructure at University of Washington's HITL.

Others that cannot be disclosed at this time but who are available under confidentiality agreement.

Board of Directors:

To be included.

5.2.3 Corporate Affiliates and Strategic Relationships

The Company has created a business philosophy that encompasses corporate affiliates and partnerships and other strategic alliances. Following is a brief outline of our current and anticipated partners and their relationship to Oz.

William Bricken, Ph.D., Human Interface Technology Laboratory

5.2.4 Technology Agreements

Available upon request and under non-disclosure.

5.3 FINANCIAL SUMMARY

The company has assembled corporate partners, technology and has identified a strong management team that, combined with a large attractive market, will provide dramatic growth and substantial profitability.

5.3.1 The Corporate Structure

Oz... International Ltd. is a Washington State Corporation founded September 4, 1991. The board of directors is made-up of five voting members and two technical non-voting members. At the current time there are 10,000,000 shares of common stock and 5,000,000 shares of preferred stock authorized.

5.3.2 Financial Projections

The following Oz... International Ltd. financial projections cover a period of five years from 1992 through 1996. The financials are supported by the sales forecast and pricing policy stated in Section 4: Marketing and Sales, and by the following assumptions:

Revenue projections were developed by the projected availability of product to the U.S. market, a cautious production buildup plan, and the sales history of workstations into the electronics industry, in particular Mentor Graphics.

Service and Support net sales are assumed to be 10% of Product sales net of OASIIS OEM sales. The cost of goods sold is assumed to be 60% of Service and Support net sales.

Product Cost of Goods are based on initial engineering estimates and vendor quotations. The software development is expensed and not included in the cost of goods.

Operation expenses were projected in detail for 1992 and 1993 as needed to complete product development, build a corporate infrastructure and launch the product. Thereafter all estimates are based on ratios on net sales and are derived from comparable industry models. Selling, General and Administrative expenses are assumed to be 30% of net sales. Depreciation and amortization are provided for on a five year life, straight line basis.

Interest: No calculations are made for either interest earned or interest expensed.

Taxes are assumed to be a straight 34% with utilization of losses carried forward.

Accounts receivable are assumed to be 20% of sales and payables are aged 60 days.

Inventories are assumed to be 5% of net sales.

Software development cost are assumed expensed in the year incurred. Although FASB 86 calls for the capitalization of software development costs at certain points, such capitalization would increase net income and is not reflected in these projections.

Cash proceeds are from equity financing staged throughout the first five years.

Oz PRODUCT COST OF GOODS

| Year End December 31, | 1992 | 1993 | 1994 | 1995 | 1996 |
|---|--------|-----------|--------------|--------------|---------------|
| Product Cost times Sales Volume: | | | | | |
| OASIIIS | \$ 0 | \$ 66,000 | \$ 220,000 | \$ 285,000 | \$ 380,000 |
| IDW-1 | 0 | 0 | 841,050 | 4,431,700 | 8,522,500 |
| PS-1 | | | 152,850 | 2,557,500 | 9,207,000 |
| Total | \$ 0 | \$ 66,000 | \$ 1,213,900 | \$ 7,274,200 | \$ 18,109,500 |
| Unit Cost Standard per Year | | | | | |
| OASIIIS | \$ 550 | \$ 550 | \$ 550 | \$ 380 | \$ 380 |
| IDW-1 | 0 | 0 | \$46,725 | \$34,090 | \$34,090 |
| PS-1 | 0 | 0 | \$10,190 | \$ 5,115 | \$ 5,115 |

9/17/92

Figure 5-2: Oz Product Cost of Goods

PROFORMA BALANCE SHEET

| Year End December 31, | 1992 | 1993 | 1994 | 1995 | 1996 |
|---|---------------------|---------------------|---------------------|----------------------|----------------------|
| ASSETS | | | | | |
| Current assets: | | | | | |
| Cash | \$ 632,312 | \$ 315,200 | \$ 404,693 | \$ 1,120,157 | \$ 2,110,393 |
| Accounts receivable | 0 | 284,000 | 914,444 | 6,358,150 | 12,591,514 |
| Inventories | 0 | 71,000 | 228,611 | 1,589,538 | 3,147,879 |
| Total Current Assets | <u>632,312</u> | <u>670,200</u> | <u>1,547,748</u> | <u>9,067,845</u> | <u>17,849,786</u> |
| Equipment and property | 50,000 | 482,000 | 830,000 | 1,520,000 | 3,810,000 |
| Other assets | 450,000 | 360,000 | 270,000 | 180,000 | 90,000 |
| Total assets | <u>\$ 1,132,312</u> | <u>\$ 1,512,200</u> | <u>\$ 2,647,748</u> | <u>\$ 10,767,845</u> | <u>\$ 21,749,786</u> |
| LIABILITIES AND STOCKHOLDERS' EQUITY | | | | | |
| Current liabilities: | | | | | |
| Accounts payable | \$ 60,512 | \$ 334,400 | \$ 447,840 | \$ 2,352,934 | \$ 4,660,945 |
| Notes payable | 0 | 0 | 0 | 0 | 0 |
| Income taxes payable | | | | 509,419 | 1,047,765 |
| Total current liabilities | <u>60,512</u> | <u>334,400</u> | <u>447,840</u> | <u>2,862,353</u> | <u>5,708,880</u> |
| Stockholders' equity: | | | | | |
| Common stock | 450,000 | 450,000 | 1,450,000 | 1,450,000 | 1,450,000 |
| Convertible Preferred A & B | 1,000,000 | 2,500,000 | 2,500,000 | 2,500,000 | 2,500,000 |
| Retained earnings | (378,200) | (1,764,200) | (1,750,092) | 3,955,492 | 12,091,075 |
| Total stockholders' equity | <u>1,071,800</u> | <u>1,185,800</u> | <u>2,199,908</u> | <u>7,905,492</u> | <u>16,041,075</u> |
| Total liabilities and stockholders' equity | <u>\$ 1,132,312</u> | <u>\$ 1,520,200</u> | <u>\$ 2,647,748</u> | <u>\$ 10,767,845</u> | <u>\$ 21,749,786</u> |
| 10/20/92 | | | | | |

Figure 5-3: Proforma Balance Sheet

PROFORMA STATEMENTS OF INCOME

| Year End December 31, | 1992 | 1993 | 1994 | 1995 | 1996 |
|-----------------------------------|-------------|---------------|--------------|---------------|---------------|
| Revenues: | | | | | |
| Product Contracts | \$ 0 | \$ 120,000 | \$ 3,280,200 | \$ 27,595,000 | \$ 57,306,700 |
| | 0 | 1,300,000 | 1,000,000 | 1,500,000 | 0 |
| Service and support | 0 | 0 | 292,020 | 2,695,750 | 5,650,870 |
| Total Revenues | 0 | 1,420,000 | 4,572,220 | 31,790,750 | 62,957,570 |
| Cost of revenues: | | | | | |
| Product Contracts | 0 | 66,000 | 1,213,900 | 7,274,200 | 18,109,500 |
| | 0 | 650,000 | 370,000 | 450,000 | 0 |
| Service and support | 0 | 0 | 175,212 | 1,617,450 | 3,390,522 |
| Total cost of revenues | 0 | 716,000 | 1,759,112 | 9,341,650 | 21,500,022 |
| Gross Margin | 0 | 704,000 | 2,813,108 | 22,449,100 | 41,457,548 |
| Operating expense: | | | | | |
| Sales, administration and general | 211,700 | 1,040,000 | 1,374,000 | 9,937,225 | 19,687,271 |
| Research and engineering | 166,500 | 1,050,000 | 1,425,000 | 4,768,613 | 9,443,636 |
| Total operating expenses | 378,200 | 2,090,000 | 2,799,000 | 14,705,838 | 29,130,907 |
| Operating income | (378,200) | (1,386,000) | 14,108 | 7,743,263 | 12,326,642 |
| Other income (expense) | | | | | |
| Income before income taxes | (378,200) | (1,386,000) | 14,108 | 7,743,263 | 12,326,642 |
| Provision for taxes | 0 | 0 | 0 | 2,037,678 | 4,191,058 |
| Net income | (\$378,200) | (\$1,386,000) | \$ 14,108 | \$ 5,705,585 | \$ 8,135,583 |
| | | | | | 10/20/92 |

Figure 5-4: Proforma Statements of Income

PROFORMA CASH FLOW STATEMENT

| Year end December 31, | 1992 | 1993 | 1994 | 1995 | 1996 |
|------------------------------------|-------------------|-------------------|-------------------|---------------------|---------------------|
| Cash beginning of period | | \$ 632,312 | \$ 315,200 | \$ 404,693 | \$ 1,120,159 |
| Cash from operations: | | | | | |
| Net income | (\$ 378,200) | (\$ 1,386,000) | \$ 14,108 | \$ 5,705,584 | \$ 8,135,583 |
| Depreciation & Amortization | 0 | 150,000 | 250,000 | 400,000 | 800,000 |
| Accounts receivable | | -284,000 | -630,444 | -5,443,706 | -6,233,364 |
| Inventories | 0 | -71,000 | -157,611 | -1,360,927 | -1,558,341 |
| Current liabilities | 60,512 | 273,888 | 113,440 | 2,414,513 | 2,846,356 |
| Net cash from operations | -317,688 | -1,317,112 | -410,507 | 1,715,464 | 3,990,234 |
| Cash used for investment: | | | | | |
| Purchase of property and equipment | -50,000 | -500,000 | -500,000 | -1,000,000 | -3,000,000 |
| Cash from financing: | | | | | |
| Notes payable | 0 | 0 | 0 | 0 | 0 |
| Convertible Preferred stock | 1,000,000 | 1,500,000 | | | |
| Common stock | | | 1,000,000 | | |
| Net cash from financing | 1,000,000 | 1,500,000 | 1,000,000 | 0 | 0 |
| Cash end of period | \$ 632,312 | \$ 315,200 | \$ 404,693 | \$ 1,120,157 | \$ 2,110,393 |

9/17/92

Figure 5-5: Proforma Cash Flow Statement

6. EduSpace TECHNICAL DESCRIPTION

Experience with human-computer interaction clearly demonstrates a massive gap between the mathematical functioning of the computer and the intuitive disposition of people while learning. *Computer technology has only recently become advanced enough to solve the problems it creates with its own interface.* EduSpace integrates reliable mathematical models with an intuitive behavioral interface, bridging the gap between unfriendly computing and non-computational learning.

Experience with multimedia and virtual environment interfaces has confirmed knowledge from educational psychology about how people learn. *Since humans develop in a spatial environment, our knowledge skills are anchored to interactions in spatial environments.* EduSpace provides spatial tracking, three dimensional modeling, and inclusive display. To provide the performance necessary for spatial modeling, EduSpace includes advanced computational techniques and the customized VE-Chip for spatial computation.

Experience with artificial intelligence techniques in computer-aided instruction has shown that *people form models of processes while learning.* EduSpace provides direct interaction with models, a formal representation for task and student modeling, and a unified programming model based on behavioral software entities. In order to support interaction with realistic computer models in space, EduSpace includes an integrated operating shell and the customized MS-Chip for symbolic computation.

6.1 TECHNICAL OVERVIEW

The EduSpace technical architecture seeks both focus and generality. We have designed the lower levels of the Eduspace system to be homogeneous, powerful, and generally transparent to the user. The interface is intuitive, powerful, and task specific.

The following components of the technical architecture define the Instructional Development Workstation. Each component is described briefly below and in detail in the sections which follow.

Interface Hardware: The physical components of the EduSpace workstation include the Interface Chair, which combines ergonomic comfort with high functionality, and the OASIIS spatial position and speech recognition system, which combines high efficiency with ease of use.

Computational Hardware: The EduSpace Accelerator Board supports complex training tasks and simulations with distributed process management and customized drivers for Oz interface hardware. The board houses the EduSpace VE-Chip and MS-Chip for super-computer performance.

Mathematical Approach: Two mathematical techniques provide a formal basis for the design of both the EduSpace accelerator chips and the algorithms used for task

modeling. Boundary Logic introduces proven techniques for simplifying rulebased inference. Algebraic Match-and-Substitute provides a general computational strategy for implementing constraint-based reasoning.

Systems Software: Systems software tools coordinate hardware resources with the computational algorithms that provide experiences to the participant. The EduSpace Operating Shell manages resources, data, communication, entities, and models.

Entity Software: Like objects in object-oriented programming, entities serve as the primary organizational principle for programming EduSpace. Entities act as behavioral systems, providing a natural metaphor for responsive, situational computation.

Domain Models: The strength of EduSpace training capabilities lies in its intuitive yet realistic modeling of tasks and of the process of training. Qualitative Task Modeling and Interactive Process Modeling provide powerful techniques for structuring, measuring and understanding learning experiences.

Interface Software: The EduSpace Development Tools are described in Section 3: The EduSpace Product. The interface provides these tools:

- **TaskMaster** for developing task oriented learning experiences,
- **RuleBuilder** for knowledge acquisition and task modeling,
- **MediaMaster** for multimedia and virtual environment management,
- **ProfileBuilder** for student modeling and assessment of progress, and
- **TrainTrack** for managing the EduSpace session.

OASIIS and the Interface Chair provide hardware support for transducing participant behavior, for recognizing position and voice, and for displaying three dimensional and multimedia environments. These hardware interface tools are supported the Eduspace Accelerator Board and its embedded VE-Chip and MS-Chip parallel accelerators.

The software interface is designed for ease of use and is customized to the training task. Task and student modeling are supported by a qualitative process modeling approach which expresses processes in terms of relational graphs. These graphs are properties of entities, the organizing programming metaphor of Eduspace. Entity activity is coordinated and maintained by formal mathematical methods, so that behavioral specifications are expressed in algebraic structures which simplify using match-and-substitute computation. Algebraic computation is integrated with control using Boundary Logic, which also improves the performance of software algorithms. The resulting algebraic logic is supported by the Eduspace Accelerator Board and its parallel chips.

Together, the silicon accelerators, mathematical techniques, systems software, the entity abstraction, domain modeling, intuitive interface, and customized interaction hardware devices form a suite which promises to deliver high quality instructional design and performance to the workplace, moving the training of American employees into the 21st century.

6.2 INTERFACE HARDWARE

The Issue: Learning is situated, and is affected by the comfort, integration, and ease of the learning environment. Traditional CBT fails to recognize situated learning.

EduSpace integrates inclusive display and sensing hardware into an ergonomic chair-based interface system. Comfortable, non-intrusive, interface hardware situates learning within the virtual environment.

The physical hardware which presents EduSpace learning experiences to the trainee is designed with ease of learning in mind. Instructional designers and trainees alike will experience EduSpace from the comfort of the Interface Chair. To provide interaction with the digital environment, EduSpace includes the OASIIS position sensor and speech input system.

6.2.1 The Interface Chair

Environmental distraction decreases both job performance and learning. The *ergocentric design* of the Interface Chair enables the instructional developer as well as the student to focus on their task without distraction or discomfort.

The Interface Chair combines award-winning dynamic ergonomic design with functional utility. It features:

- Safe, comfortable, controlled and unobtrusive seating.
- Fully integrated display and tracking interface devices.
- A high resolution inclusive display mounted on an omni-directional counterbalanced support system for easy transfer between monitor based and inclusive viewing.
- Wireless radio frequency communication between peripheral devices and the central computer, eliminating wires, tethers, and harnesses.
- A self-contained power supply providing 16 hours of continuous use.
- A spatial sound system.
- The OASIIS position and speech recognition system.

6.2.2 OASIIS

The primary interface hardware to EduSpace is the Optical And Speech Interactive Interface System (OASIIS™). OASIIS is a combination image and speech pattern

recognition tool that recognizes bodily movements and voice commands. The movements and commands generated by an EduSpace participant are converted into digital input for controlling computer images and providing operational instructions to EduSpace courseware components. OASIIS provides movement, orientation and command data to software entities managing audio, video and modeling.

When used with CAD products, OASIIS replaces the mouse and the digital tablet, allowing six dimensional spatial and volumetric input. OASIIS operates with a diversity of *physical tools* that provide tactile and spatial ergonomic feedback to the learner.

OASIIS has four major subsystems, each of which is unique in design. As a system they provide interaction with computer generated 2D/3D inclusive and multimedia environments. The OASIIS subsystems include:

- *Spatial position and movement* recognition, providing six degree of freedom (position and orientation, translation and rotation) information to the viewing model,
- *Speech* recognition, providing a 64 word vocabulary for spoken command control of the display,
- *The Wand* software for indicating, grabbing, and moving entities in multimedia and virtual environments, and
- *White object* recognition and integration.

OASIIS can track hand movement and gestures for both hands, providing two hand manipulation of spatial information. OASIIS provides a *white object* capability: when the participant is holding a physical object such as a pen or a flashlight, a virtual image of the object, with appropriate functionality (the virtual pen writes, the virtual flashlight illuminates), appears in the display.

White objects provide true physical feedback to participants in virtual environments. Physically, a white object can be a common actual object, such as a pen, a pistol, a potato, or a piano keyboard. The physical object may be augmented with tiny optical markers for ease of spatial identification. In virtual space, the white object is a *model* of the physical object. Recognition of the physical object is assisted by the software model of its geometry and functionality.

Integration of the physical and the virtual is provided by the participant. There is no attempt to solve the difficult problem of *registration* of virtual outlines on top of physical objects in physical space. Only the *feel* of the white object is transferred.

OASIIS speech recognition provides a unique, built-in capability to control environmental behavior through voice commands. Currently, speech recognition requires a relatively expensive add-on board. Since the DSP processing capabilities of

OASIIS can provide both image and sound recognition, OASIIS provides both without incurring extra cost.

The OASIIS Wand serves as a generic pointing, grabbing and steering device. The physical Wand is a hand held actual object which supplies position and orientation information to the model. The sensor information inhabits a virtual object held by a virtual hand.

Functionality is assigned to the Wand by making it sensitive to commands generated through speech input, gestures, or the keyboard. These commands are inserted into the behavioral dispositions of the Wand as logical rules. The general form of a response rule is "if recognize-input then generate-associated-output". Some functions of the OASIIS Wand include:

- *Ray on/off*: A ray emanates from the end of the virtual rod, colinear with it.
- *Identify*: The first object which the ray penetrates returns its name.
- *Connect*: Construct a communications port between the rod and the identified object.
- *Jack*: Teleport the viewpoint (the ray vector) of the participant to the identified point on the object.
- *Grasp*: Attach the end of the ray to the identified object. When the Wand is moved, the object stays attached. When the Wand is rotated, the object rotates.
- *Normal*: Rotate the identified object so that the intersecting ray is normal to the object's surface.
- *Sight*: Jack into the Wand. The viewpoint of the participant issuing the command is linked to the ray vector.
- *Move faster/slower*: Move the viewpoint of the participant along the ray vector.
- *Mouse*: emulate mouse movement and button commands.

6.2.3 Hardware Architecture

The physical architecture of the Instructional Development Workstation and of the Presenter Station is presented in Figure 6-1. Presenter Stations and Development Workstations share a common FDDI or Ethernet connection. Both also include the EduSpace Accelerator Board.

The host computer provides input capabilities for external devices such as:

- Keyboard
- CRT
- Printer
- External memory
- Optical Scanner
- CD ROM
- Camcorder and VCR

In addition to coordinating standard external devices, the host computer provides FDDI and ethernet connections to remote, distributed computing resources. Networked resources may include other EduSpace Stations, external application programs, standard workstations installed on the accessible network, and links to data sources.

6.3 COMPUTATIONAL HARDWARE

The Issue: Complex multimedia and virtual environment displays require tremendous computational resources for realtime interactivity.

EduSpace supports super-computer performance with a cost effective proprietary accelerator board with customized content-addressable memory chips.

EduSpace software and the accompanying hardware accelerators function in standard UNIX environments, yet provide super-computer performance at PC prices. To achieve the several orders of magnitude performance improvement necessary for realtime digital interactivity, Oz will use sophisticated but proven innovations in both hardware and software. Performance is provided by a pair of highly customized, very-reduced-instruction-set computer chips which implement massive parallelism.

Figure 6-1: EduSpace Hardware System Architecture

6.3.1 EduSpace Accelerator Board

The EduSpace Accelerator Board (EAB) provides customized input/output and computational support for Oz hardware interface products, and houses the proprietary VE-Chip and MS-Chip accelerator chips which provide the bulk of performance improvement.

The EAB architecture in Figure 6-2 is designed to effectively implement a simple and extremely efficient computational model. It optimizes the use of computational resources provided by the host operating system and by the VE-Chip and MS-Chip accelerator chips. The combination of specialized hardware and an innovative mathematical approach provides truly superior performance.

The EAB provides a customized input controller for the OASIIIS speech and position recognition system and for the Interface Chair. Behaviors monitored by Oz peripherals will be processed directly by the VE-Chip, so that all behavioral measurements generated by OASIIIS and by the Interface Chair will be subjected to negligible transmission and computation delay. The EAB design also includes expansion space for future virtual environment tools.

The input processor in the EAB connects to the EduSpace accelerator chips and to the host platform native CPU via a reconfigurable VME bus. It also supports 3D video-out with two stereo pair video-out channels, and 3D audio-out with two audio channels.

Figure 6-2: EduSpace Accelerator Board

6.3.2 VE-Chip and MS-Chip Computational Engines

The accelerator engines on the EAB provide massively parallel processing of symbolic and spatial information. The engine itself consists of a controller chip and several proprietary dynamic content addressable memory (CAM) chips. The configurable controller chip will contain local ROM and programmable logic arrays. It will coordinate data flow through the CAM accelerator chips.

Oz is developing two customized CAM chips. Both are *dynamic* in that *every CAM cell on the chip has specialized processing capability*, allowing massively parallel computation at the bit level.

The MS-Chip is specialized for parallel match-and-substitute computation (see Section 6.4: Mathematical Approach). It achieves performance on the order of *trillions of comparison operations per second* by testing equality of bits in parallel over the entire cell array of the chip. In one bus cycle, all fields in all words in the MS-Chip can be compared for equality to an input data vector.

The MS-Chip performs parallel substitutions within a few clock cycles. A typical mathematical function requires about 20 substitutions for complete evaluation, using the match-and-substitute formal mechanism of lambda calculus. Running at 40 MHz, this means that the MS-Chip can perform 2 million function expansions per second. This performance is equivalent to a Cray I. Simpler processes, such as determining logical consequences or identifying spatial collisions, require fewer matches, resulting in greater speed for these common calculations.

The VE-Chip is specialized for parallel spatial comparisons. It provides search, manipulation, and interaction with all spatially distributed data in discrete n-dimensional space in $2n$ bus cycles. The VE-Chip is used for spatial tracking, for collision detection, for filtering, for implementation of spatial attributes, and for image matching.

The VE-Chip and the MS-Chip chips are gangable and cascadable through the VME bus, permitting scalable increases in system speed for complex training environments. EduSpace Accelerator Boards support resource sharing between distributed workstations containing other EABs. Internally, the EAB supports efficient allocation of resources across active entities in shared virtual and multimedia environments.

Performance can be enhanced in any EduSpace Station by increasing the number of MS-Chip and VE-Chip components on the accelerator board. These chips are scalable, providing approximately 1000 function evaluations per second *per dollar investment*.

Note that the VE-Chip and MS-Chip accelerate computations and not display, which is the responsibility of a graphics accelerator. Oz intends to use commercially available graphics accelerators from Silicon Graphics for visual output, and customized DSP chips for three dimensional audio output. We do not yet know the extent of improvement or the average size of graphical models that can be processed by a

combined graphics and computational accelerator system within the maximal lag time for sensory fusion (approximately 1/20 of a second).

6.4 MATHEMATICAL APPROACH

The Issue: Algebraic computation is slow on stack-based computers, yet provides highly desirable computational features. The traditional strategies for representing algebraic logic are inefficient.

EduSpace combines algebraic logic with functional programming to create an efficient match-and-substitute computational strategy that can be implemented on a customized massively parallel chip. EduSpace employs an innovative mathematical approach to efficiently structure computation.

Reliability in computer software is achieved by principled implementations of circuitry, software, and interface based on formal mathematical models. The technical architecture of EduSpace is rooted firmly in mathematics. Although our techniques are generally innovative, they are established and proven within specific mathematical communities.

6.4.1 Boundary Logic

Oz has an exclusive license to Boundary Logic, a coordinated suite of mathematical techniques which compile logical and algebraic data structures into highly efficient microcode for controlling the EduSpace Accelerator Board.

The computational model of interaction and training within EduSpace is based in logical rules and algebraic equations. Boundary logic uses two representational innovations to simplify the storage and the reduction of rules and equations.

A boundary consists of a closed edge and the space contained by the edge. In boundary logic, *containment* and *empty space* are used to express logical concepts and connectives.

- Boundary logic shrinks the use of memory by assigning meaning to *empty space*, to the absence of a representation.
- Boundary logic manages parallel deduction by using *containment* (the nesting of boundaries) to express logical and temporal dependencies, permitting principled parallel simplification of complex expressions.

EduSpace parallel deduction uses both a rewrite strategy (destructive in-line substitution) and a graph reduction strategy (substitution of pointers to permanent data structures). Rewriting is space conserving, since simplification substitutes a smaller

expression for a more complex one. The cost is a loss of previous state, since rewriting is destructive. Graph reduction manipulates an abstract representation of the database, conducting inference and logic control without effecting the underlying data structures.

Importantly, the vast majority of boundary logic substitution operations are *erasures rather than rearrangements*. Erasures are obvious reduction steps to take, since they remove irrelevant structure. Erasures are easily implemented, and have the pleasant property of freeing memory. The primary conceptual innovation that permits extensive simplification by erasure is the use of empty space (i.e. of non-representation) as a semantic element in the theory of representation.

Void-based representation provides a startling innovation for logic, control structure and deduction. The traditionally accepted *minimal basis* for elementary logic is two concepts, {if-then, false}. All other logical constants and connectives can be expressed in terms of this minimal basis. Minimal logical representations are not usually used in traditional microcodes, because they tend to explode the size of the representation.

In boundary logic, the void can be assigned to the concept false, and a container can be assigned the meaning of if-then (i.e., what is inside implies what is outside). The resulting minimal representational basis for boundary logic is one concept, {container <void>}. That is, the notation of logic is reduced while the expressability stays the same. To state this mathematically, there is a *many-to-one map* from traditional logic to boundary logic. The advantage for memory management is that empty space takes up no memory. The advantage for computation is that, for the most part, simplification takes place by erasure, by substitution of the void.

The use of the void both for representation and for simplification makes EduSpace computation far more efficient than traditional systems which use duplication and rearrangement as the main computational strategy. It is always possible to generate computations that are complex and time-consuming. Oz, of course, cannot solve this problem. Our mathematical techniques do, however, exhibit provable asymptotic improvement. For a broad range of problems, boundary logic has been shown to be on average 500% more efficient than resolution, as implemented by the Boyer-Moore theorem prover (W. Bricken, 1987b).

The techniques of boundary logic have been developed for other areas of mathematics, in particular for integers and for algebra. Oz has an exclusive option to license these technologies for future development of extremely efficient general purpose processors.

6.4.2 Match-and-Substitute

All operations generated by EduSpace will be implemented by a single computational algorithm: *match-and-substitute*. Elements which have a complex syntactic description (such as the sum $2x+3x$) are identified by syntactic pattern matching. Then a simpler expression (the result $5x$ in the example) is substituted for the complex one. This methodology is commonly called *algebra* (substitution of equals for equals); the implementation of EduSpace relies on techniques taught in high school algebra classes.

Mathematically, match-and-substitute has been shown to be Turing equivalent, which means that anything a computer can do, it can do with a match-and-substitute strategy. The symbolic processing program **Mathematica**, developed by Stephen Wolfram, uses match-and-substitute to perform all mathematics commonly found in academic graduate school programs. Term rewrite theory is a version of match-and-substitute well studied by the logic programming community. Substitution-based languages and machine architectures are actively pursued by the functional programming community, most notably by David Turner in England.

Algebra (that is, match-and-substitute) provides a particularly appealing mathematical model for a computational engine, for several good reasons:

- As a computational technique, algebra is common and well understood.
- Equations from any standard textbook can be programmed directly by copying. For example, to implement a concept of force, entities are first assigned mass and acceleration properties. Then the force an entity exerts in the direction of its acceleration is identified simply by asserting the equation

$$\text{Force} = \text{Mass} * \text{Acceleration}$$

into the database of the space within which the entity resides.

- In contrast to logical deduction and data-driven programming, equations are bi-directional and constraint-based. The computational process does not need to wait for the values of every variable to be known in order to make progress. In the above example, when Mass = 5, by simple pattern matching and substitution the Force equation is constrained to

$$\text{Force} = 5 * \text{Acceleration}$$

The partial information automatically generates a constraint on the relationship between Force and Acceleration for the particular entity with Mass = 5.

Algebra is not a common model for computational processing because of a historical split between logicians and algebraists during the decades that current computer architectures were being designed. It is known that programming with pure logic is mathematically identical to programming with algebra. Many research groups worldwide are working to develop algebraic programming. Recent research in the computational logic community has focused on programming issues of parallel logics (eg Prolog III). This exploration reunites computer logic with algebra, generating new techniques such as constraint-based reasoning.

Boundary logic is the Oz proprietary technology for implementing *algebraic logic*. All control structures and logical connectives of traditional programs can be converted to sets of algebraic equations using boundary logic techniques.

The EduSpace database therefore consists of large collections of equations. Facts about the physical world are naturally expressed using equations (eg $F=ma$). Control structures and logical deduction are also expressed as equations, in boundary logic notation. Since equations are simplified and solved through match-and-substitute, the entire database of EduSpace requires only one action, that of repeated substitutions.

This technique is generally considered impractical for the basis of a computational engine because it is impractical for conventional register-based CPU architectures. An Oz partner (Silicon Engines) has developed a proprietary chip technology, the MS-Chip, capable of performing match-and-substitute extremely rapidly. The speed-up is achieved by addressing the database elements in parallel, making, in effect, all substitutions for a particular variable at the same time. The standard technique of searching a database for matching items to be substituted is improved by eliminating sequential search for particular matches.

What remains is the control structure that decides which substitutions to make when. An effective ordering for substitutions in complex databases can often be identified in advance. We have developed proprietary boundary logic algorithms to structure algebraic databases for efficient processing. These structuring rules themselves can be expressed as boundary logic equations, so that the entire database simplification and control mechanism is represented concurrently within the database.

The strongest point of our match-and-substitute architecture is that it is completely homogeneous across all types of calculations. *Computation of functions, triggering of rules, control decisions, and database compilation all take place opportunistically in parallel.* This means that the MS-Chip operates within a paradigm of very reduced-instruction-set-computing (VRISC). Type checking, instruction parsing, program branching, and other low level operational details are subsumed by the single computational process. Circuit design is consequently simpler and cheaper.

The combination of rapid cycling of parallel substitutions and self-similarity of data and control permits dynamic evaluation of substitution strategies. This feature is very important for interactive virtual worlds, since the movements and behavior of the participant control the responses of the entities in the world in realtime.

6.4.3 Dynamic Parallel Database

Although the EduSpace MS-Chip performs parallel substitution during a single computational thread, it must also cope with multiple data transactions from multiple active entities and participants. To manage the coordination of interprocess communication, we have developed a variant of the Linda parallel database model developed by Gelernter at Yale. In Linda-like languages, communication and computation are treated as independent processes, relieving the programmer from having to choreograph interaction between multiple processes. Linda supports local,

asynchronous parallel processes, a highly desirable quality for complex, simultaneous, multimedia presentation systems.

Structurally, the EduSpace database consists of a collection of fragments of information, labeled with unique syntactic identifiers. Collections of related data (such as all of the current properties of Cube-3, for example) can be rapidly assembled by invoking a parallel pattern match on the syntactic label which identifies the sought after relation (in the example, matching all fragments with the label "Cube-3" creates the complete entity known as Cube-3). The EduSpace approach of fragmented data structures permits dynamic, interactive construction of arbitrary entity collections through realtime pattern-matching. Requesting "all-blue-things" creates a transient complex entity consisting of all the things in the current environment that are blue. This entity is implemented by a dynamic thread of things with the attribute "color=blue" in the data structure.

In object-oriented programming, object attributes and inheritance hierarchies commonly must be constructed by the programmer in advance. Efficiency in object-oriented systems requires compiling objects. This means that the programmer must know in advance all the objects in the environment and all their interactions. In effect, the programmer must be a god. Virtual worlds and multimedia environments are simply too complex for such monolithic programming. Although object-oriented approaches provide modularity and conceptual organization, in large scale applications they result in complex property and method variants, generating hundreds of object classes and forming a complex inheritance web. In many cases, a principled inheritance hierarchy is not available, forcing the programmer to limit the conceptualization of the world. In other cases, the computational interaction between objects is context dependent, requiring attribute structures which have not been preprogrammed. In the EduSpace implementation of entities, attributes can be generated interactively. Structures across entities can be identified in realtime based on arbitrary patterns, such as partial similarities, unbound attribute values (i.e. abstract objects), and ranges of attribute values. Programming EduSpace entities can feel exactly like programming objects, but the difference in the underlying implementation permits dynamic, realtime specification.

In summary, our mathematical computational model is parallel algebraic pattern-matching and substitution on partitioned databases which support multiple concurrent interactions. Computational efficiency and protection of software innovation are addressed by customized, proprietary chip technologies.

6.5 SYSTEMS SOFTWARE

The Issue: Complex multimedia and virtual environments require parallel computing resources and specialized operating system management techniques.

EduSpace provides a parallel distributed operating system which partitions interprocess communication from computation

The EduSpace Operating Shell (EOS) manages the complex interaction between the many components of an EduSpace Station. These components include:

- Data management
- Communications management
- Process management
- Entity management
- Software tools management

EduSpace is intended to function on a host computer running a version of realtime UNIX, such as SGI's Lynx and CMU's Mach. The EduSpace software architecture is presented in Figure 6-3. This section describes the software operating shell, the following section describes the construction, modeling and editing of entities.

6.5.1 Programmability

C is the native programming language of UNIX systems. Oz will provide a C to EduSpace Accelerator Board microcode "realtime compiler" in the form of a collection of match-and-substitute equations which convert the functional specifications of the C language into either equational specifications for the EAB (in the case of recognizable transformations) or to calls to the host UNIX operating system (in the case of remote calls or unparsable transformations).

Algebraic data which define constraints on behavior in the EduSpace environment will be supported by an algebraic specification language based on match-and-substitute computation. The EAB will also support languages currently incorporated into the HITL VEOS. These include LISP, C, UM (an graphic entity interaction specification tool), GL (a graphics language from Silicon Graphics), MIDI, X window protocol, and DXF graphics file format protocols.

Figure 6-3: EduSpace Software System Architecture

6.5.2 Data, Communications and Process Management

The Virtual Environment Operating System (VEOS) is a C implementation of a match-and-substitute operating system developed at HITL. It provides a tightly integrated computing model for data, processes, and communication. VEOS was designed from scratch to provide a comprehensive and unified management facility for generation of, interaction with, and maintenance of complex multimedia and virtual environments. The initial version of VEOS was stable in February 1991, and since then VEOS has seen two complete iterative revisions.

The VEOS kernel consists of four tightly integrated components.

- TALK manages interprocess communications.
- SHELL manages entity initialization and linkages.
- NANCY manages the distributed parallel database.
- FERN manages entity processes.

The VEOS 2.2 (release date: 10/92) also incorporates several higher level programming languages in its interface. HITL is currently using VEOS in several world construction projects. As a prototype, VEOS trades flexibility for efficiency. It is particularly effective in providing an interactive rapid prototyping capability for testing different world and entity structures. VEOS is, however, a research prototype and not a supported product.

6.5.3 EduSpace Operating Shell

The EduSpace Operating Shell (EOS) is modeled on the successful Virtual Environment Operating System (VEOS) developed by Dr. William Bricken and his students at HITL over the last three years (W. Bricken 1990). Oz has negotiated an option for an exclusive license to use the VEOS for commercial training systems, and has a long term memorandum of agreement to support and develop future HITL software prototypes which use VEOS.

The Oz exclusive license to VEOS 2.2 is for training and educational applications of the software technology incorporated in the VEOS architectural design, its implementation details and its supporting software tools.

Oz will use the VEOS architecture to guide the design of the EduSpace Accelerator Board and the EduSpace Operating Shell . The resulting integrated architecture will provide an very-reduced-instruction-set implementation with the advantages of realtime interactivity and the efficiency of compiled code. The EAB will provide the speed-up necessary to convert VEOS from a prototype to a product.

In adapting VEOS, EduSpace will use NANCY-like data storage and retrieval models to define the MS-Chip data structures. SHELL-like facilities will coordinate the data sets for entities active in each display environment. TALK-like input/output ports will coordinate inter-processor communication, and FERN-like management algorithms will provide overall coherence to each virtual entity in the system, whether it is a display entity, a resource entity, or an abstract entity.

A central advantage of EOS is that the insertion of a declaration into the database during runtime will change the display environment immediately. EOS will provide *realtime interaction with virtual and multimedia worlds*, permitting very rapid prototyping of instructional designs and construction of virtual entities while an EduSpace application is running.

6.6 ENTITY SOFTWARE

The Issue: Existing software tools for training are dated, non-modular, overly complex and lacking inclusion of the student.

EduSpace uses behavior-based entities as an organizing principle for modeling virtual environments and for the representation of self. Entities provide rapid, interactive instructional design through modularity and intuitive behavior.

The EduSpace interface (as presented in Section 3: The EduSpace Product) is built upon a foundation of entity-based software models. The EduSpace software architecture is diagrammed in Figure 6-3. An entity is a collection of resources which exhibits behavior. The entity-based model of programming has a long history, growing from formal modeling of complex systems, object-oriented programming, and concurrent autonomous processing. Agents, Actors, and Guides all have similarities to entities. Entity-based programming was developed at HITL specifically for managing the complex interactional processes of participatory multimedia and virtual reality.

Some types of entities, and how to construct and edit them, are described below.

6.6.1 Entities

Entities are the primary organizational structure for the software tool layer of the EduSpace architecture. Like the match-and-substitute computational approach of the silicon layer, entities provide a uniform, singular metaphor and design philosophy for the organization of software resources and displays.

An entity is a coupled collection of resources that can accomplish a specific task while maintaining a behavioral interface with human participants.

The EduSpace Operating Shell itself is the prototype entity. That is to say, the functional core of EOS provides management of data, processes, and communications while interacting with users. Every entity in an EduSpace environment inherits the capabilities of the operating shell. The implementation of entity functionality can be

achieved by time-sharing a single EOS entity-process on a single EOS board or by distributing EOS entity-processes over distributed, concurrent computational resources.

EOS also provides the direct link to the processing capabilities of the EduSpace Accelerator Board. EOS-based entities therefore interact with their environment at VE-Chip and MS-Chip processing rates.

In virtual environments, entities can have 3-D display properties. Each entity in a virtual world can also have behavioral properties, so that they interact dynamically with the environment. In screen-based multimedia, windows are entities which exhibit display capabilities such as video and sound. Multimedia entities can also exhibit intelligent behavior, such as knowing how to resize in any display environment, or even filtering display content which is irrelevant to the task being learned by a participant.

The self-similarity of system and software architecture provides several distinct advantages:

- The world designer does not need to distinguish between software display entities and hardware computational resources. A simple linking action is all that is needed to configure the computational environment. Invisible to the designer, EOS automatically coordinates between software and hardware. The integration is seamless because both share a common language, computational approach, communication protocol, and database storage mechanism.
- New devices, CPUs, and software capabilities can be added modularity and optionally. New resources can interact remotely with an EOS database partition, or they can run a EOS process locally, seamlessly integrating into a network of distributed processes.
- Software resources not native to EOS, such as a statistical analysis package or a text editor, can be easily incorporated into an entity, giving each entity the capabilities of any software tool within the entire computational environment.
- The participant interacts with a virtual world as the inhabitant of a virtual body. Since the virtual body is an entity, it permits each participant access to the capabilities of the computational environment. Since the virtual body resembles both participants and entities, it serves as a conformable, intuitive mediator between the physical and the virtual.

Most importantly, the underlying operation of the EduSpace system can be completely invisible to the end user. Yet using a simple rulebased interface, the participant can actively modify almost any system performance or display characteristic. As greater design control is desired, deeper layers of EOS functionality can be accessed, providing novice through expert levels of control and complexity.

6.6.2 Internal Processes of Entities

The internal process of an entity (and of the EOS system) consists of a sense-process-act loop. This loop is called a *behavioral cycle*.

The analogy to human behavior is not strong. An entity *senses* by gathering relevant data from the database. Relevance is the key. Each entity has a "perceptual" filter on the database which limits the amount and the type of data that the entity must process on each behavioral cycle. *Filters* are patterns which must be matched for a data fragment to be relevant to the entity. For example, an entity configured with a sound play-back capability can sense sound bits stored in the database. If it lacks a sound capability, it will ignore all data labeled as sound. Since filters are expressed in terms of matching, they are implemented in the MS-Chip. That is, all information relevant to a particular filter is matched in parallel, in a few clock cycles.

The processes internal to an entity are controlled by two separate processing loops. The *React* loop reads sensed data and immediately responds by posting modified data to the common database. This cycle handles all realtime interactions and all reactions which do not require additional computation or local storage. The *Persist* loop runs on resources local to the specific entity, and is not responsive in realtime. Persist loop computations typically require local memory, function evaluation, and inference over local data. Persist functions can copy data from the shared database and perform local computations in order to gather information, but there is no time constraint on returning results. By decoupling local computation from environmental reactivity, entities can react in a timely manner while complex responses can still be evaluated whenever computational resources permit.

6.6.3 Software Tools

EduSpace provides several layers of software tools, including:

- software hooks to operating system resources for programming and reconfiguring the system itself (used by the **TrainTrack** tool),
- generic tools for configuring and interacting with multimedia and virtual environments (used by the **MediaMaster**),
- tools for building training environments and designing learning activities (used by the **TaskMaster** and the **RuleBuilder**),
- knowledge acquisition tools (used by the **MediaMaster**), and
- pedagogical tools for coaching and tutoring the student based on the process model of the student (used by the **ProfileBuilder**).

Figure 6-3 lists the development tools and languages that will be added incrementally over time to the EduSpace software infrastructure. These tools will expand the capabilities of the instructional designer by adding new functionality to the user

interface. In this way, backward compatibility with earlier EduSpace software releases can be maintained. Some of the capabilities of these development tools are described below.

6.6.4 Space Management

Space is an important part of every environment. In the simulated environments of EduSpace multimedia and virtual reality applications, the Space entity manages all the spatial interactions between the other entities and participants.

The VE-Chip on the EAB serves as the Space manager for EduSpace. This chip greatly improves the performance of virtual environments with spatial characteristics such as gravity, fluid flow, object collision, and window placement.

The Space Entity determines the underlying metric of the display environment. In virtual worlds for example, the fabric of space may not be continuous. A participant may move from one position to another by instantly transferring perspective rather than by moving through distance. Space entities also manage location information and collisions, since space provides the concept of proximity and the global coordinate system.

Other properties managed by space include linearity, opaqueness, granularity, force fields such as gravity, and wind.

6.6.5 The Virtual Body

As an interface becomes more interactive and natural, it incorporates more of the physical behavior of the human participant. In a multimedia environment, display interactivity is constrained to the monitor. Multimedia increases the display bandwidth by adding video, sound and other sensory modalities. In a virtual environment, the user becomes a participant, the participant's actions form a broad bandwidth of behavioral communication between person and machine.

Since the participant is included within the virtual environment, the representation of self is fundamental to virtual interface design. The Virtual Body is the primary reference point, the interface between the person and the virtual environment. It provides direct access to computational graphic objects, it is the channel of direct action and control. Monitoring the Virtual Body provides the computational system with a complete record of actions taken by the participant.

The Virtual Body is a component of the student model. It is used for:

- attaching arbitrary hardware input devices to arbitrary representations of components of our body. Usually the linkage will emphasize naturalness.
- maintaining an accurate physiological model of the participant.

- making psychometric measurements of behavior in a virtual environment.
- calibrating the EduSpace environment to the physiology of particular participant.
- maintaining coherence between the participant's model of physical activity and the virtual representation of that activity.

The Virtual Body includes a *physiological model* of the participant. The physiological model serves as the calibration tool for customizing interface hardware performance to individual physiological differences (recognizing the flexibility of a particular participant's hands for example). This model provides a modeling tool which assists the OASIIS position recognition system when image recognition is ambiguous. It also provides vital physical feedback to the participant by maintain the image of the physical body in a virtual environment.

The Virtual Body is a unique tool with several advantages for interaction with inclusive environments. These include:

- The sensors measuring participant activity are designed to be transparent. Natural physical movement directly affects the computation; there is no apparent interface. The Virtual Body software maintains the illusion of direct interaction.
- Mapping between physical action and computational effect is flexible and dynamic. A spoken word, for example can change the computational effect of shifting one's gaze from "Identify that object" to "Transport me to that object."
- Physical actions in a virtual environment furnish psychometric data on performance, resource expenditure (load), and cognitive model. Since action can be taken literally (there is no symbolic transcription filtering the meaning of behavior), performance in a virtual environment mimics performance in reality. As long as the representation of the task is valid, the participant's behavior directly indicates that participant's ability to perform.
- Humans have immense variety of physical form. For accurate interaction between the physical and the virtual, it is necessary for software development tools to be responsive to individual differences. A calibrated virtual body magnifies the sense of presence in a virtual world, and enhances the pleasure of virtual interactions.

6.6.6 Multiple Participants

The EduSpace Operating Shell supports multiple participants within the same computational environment. Since every participant has a virtual body, every participant is represented within the virtual environment as an entity. The same mechanisms which manage program entities can also manage participant entities.

Virtual environments, whether two dimensional multimedia environments or three dimensional inclusive environments, lose their appeal without interaction with other humans. Multiple participants are necessary for computer-based learning, to include both teachers and peers.

What are the mathematical tools which coordinate actions of multiple participants in virtual worlds? The problem is similar to that faced in the design of software architectures for intelligent distributed agents (Genesereth and Nilsson). Entities differ from agents in these ways:

- Entities can be fully connected while maintaining individual perspective (merged without resolving inconsistencies). Sharing information between automated entities is a matter of convenience.
- Entities inhabited by participants do not require automation of knowledge. Some representation problems in world models can be interactively deferred to humans. Sharing information between participants is a matter of negotiation and choice.
- The environment is computational. Some difficulties with modelling real world phenomena can be finessed in a virtual world. The environment, itself an entity, can be fully responsive.

Coordination between participants depends upon mutually consistent models of shared environments. Network-based research on intelligent agents has been developed for adversarial environments. Virtual worlds, in contrast, permit complete communication with environmental entities while maintaining individual perspectives. Unlike objective reality, virtual spaces accommodate multiple concurrent perspectives, each associated with a different participant. One participant can perceive a green desk, for example, while another can perceive the same desk to be brown. A third may even not perceive the desk at all.

Inconsistencies across participants can be negotiated or managed. Negotiation can be sensory (sharing viewpoints), data based (sharing memories), or rule based (sharing dispositions). Maintenance of contradiction in virtual worlds requires merging inconsistent perspectives and information without disabling action or inference.

Oz is developing an approach to inconsistency that uses a three-valued logic with an imaginary boolean value. The imaginary logical value is analogous to the imaginary numerical value i . It does not interact with two-valued deduction, permits a weaker form of inference in the presence of inconsistency, and allows lazy resolution of contradictions. This approach is equivalent to a hypothetical worlds approach, but splits at the variable level (bottom-up) rather than at the model level (top-down).

6.6.7 Tools for Creating Entities

As well as tools for the design of training experiences, EduSpace provides capabilities for editing the elements of those experiences. The majority of the specialized tools which provide geometric shape, sound coordination, animation, device communication, etc. are commercially available software which can be easily integrated into EduSpace functionality. EduSpace will provide import capabilities for standard data formats. Oz will also develop proprietary software when needed.

The need for inclusive tools has not yet arisen in the software marketplace because working in three dimensions is not realistic on a two dimensional screen. Recent advances in inclusive VR display equipment, and particular advances made at HITL in how to use VR equipment effectively for training, suggest that construction and interaction in three dimensions is preferable for many learning tasks, and mandatory for motor learning. To have an ambulance driving trainee demonstrate competence in high speed vehicle operation in an inclusive simulator is far better than just giving the trainee a multiple choice test. In particular, inclusive editors provide the ability to test construction skills by simulating that construction in a three dimensional virtual environment.

All interface tools, particularly the RuleBuilder, the TaskMaster, and the ProfileBuilder provide the capability of editing entities. The entity library will provide templates for standard simple entities and their hierarchical relationships. The functionality required for editing entities is provided at the interface by SpecBoxes, which are coordinated by the EduSpace courseware tools.

Changing the Static Form of an Entity

The static components of entities are described and changed using primitives for creating three dimensional spatial models. EduSpace supports output standards such as DXF, PICT, Postscript, NTSC for video, and MIDI for sound.

Entities can be statically modeled in any modeling system which supports standard file formats. Objects designed for a three dimensional environment require 3D specifications, but 2D objects can also be incorporated into three dimensional virtual spaces, and three dimensional scenes can be projected into two dimensions for display on multimedia screens.

Since we expect that the specification of different tasks will require different construction styles, we are developing strategic partnerships with producers of hybrid geometric software libraries. EduSpace geometry editing styles will include:

- the standard boundary and constructive solid geometry approaches,
- a generalized sweep approach most suitable for gesture based construction,
- algebraic specification, and

- enumeration and decomposition.

Enumeration and decomposition are simple construction techniques that have never succeeded by themselves in the CAD marketplace. They are still important to include in an intuitive environment because they are intuitively easy to use. The *enumeration* of regular space-filling solids, such as voxels or cubes, is tedious without algorithmic assistance. With appropriate inclusive tools, such as a three dimensional generalization of current two dimensional drawing tools, enumeration becomes a sculpting tool with arbitrary computational chisels.

The *decomposition* of a space into arbitrary sections can be made easy with a wand that acts like a spatial knife, carving cells by drawing planes with broad gestures through space.

Algebraic specification of curved lines and planes is particularly convenient in EduSpace, since the implementation is based on algebraic manipulation. Using equations to express spatial constraints can be computationally complex, but the MS-Chip promises dramatic performance improvements.

SpecBoxes for editing static forms will also provide capabilities for assigning and changing attributes of entities, such as color, scale and decomposability. When these attributes change over time, they appear as dynamic activity.

Changing the Dynamics of an Entity

Entity dynamics is achieved by associating behavior functions with sensory input and with processes internal to the entity. EduSpace will provide an extensive array of designed behavioral functions. The pedagogical style for a particular lesson, for example, could be selected from the following

- ignore errors
- store errors without notice
- notify errors immediately
- notify repeating errors
- notify errors and display correction.

Each style would be automatically applied across the lesson, while the style of notification could be different for each different trainee.

The EduSpace courseware development tools provide templates for designing and writing the logical structures that trigger behaviors. The SpecBox interface can be as simple as "Copy my actions", which would trigger a rule that recorded the actions of the participant for later playback and display.

Editing dynamic characteristics of entities is accomplished by the rulebased interaction tool for creating entity behaviors, RuleBuilder. Behaviors can be algorithmic, reactive, responsive, inferential, coordinated or autonomous. Entities exhibit persistent behavior

by running algorithms that do not interact with the environment. This is the general case with most programmed simulations.

The RuleBuilder tool permits associating any entity attribute with an arbitrary looping function, causing that attribute to vary over time. Smooth functions create continuous behavior; chaotic functions create complex behavior. Branching behavior is provided by logical rules which use conditional functions.

Reactive entities have rules that trigger when specific events are posted to the environmental database. The trigger is expressed as a perceptual capability of the entity. Entities which can perceive and react to a hand gesture, for example, can perceive (can read the database of) the participant's hand and respond to it.

RuleBuilder, the dynamics editor, provides simple tools for asserting perceptual limitations and preferences for constructed entities, for asserting reactive rules in if-then-else format, and for prioritizing the distribution of computational resources available to the entity for behavioral activities.

Responsive entities also react to their environment, but they have memory resources to store previous experiences. These entities can exhibit complex delayed responses and critical incidence behavior. A screw being inserted into the wrong hole by a novice carburetor repair trainee may wait until the third repetition of the error before responding with a correction. RuleBuilder provides simple accumulation commands for specifying memory dependent behavior.

Inferential entities have a small inference engine which they can apply to their accumulated database of experiences and internal rules. Although the search is time consuming, inferential entities can find interesting, useful deductions which can generate unique behavior. Inference can also be used for goal directed activity such as planning and information gathering. RuleBuilder makes coupling inference with data straightforward by providing mathematical relational structures, such as automated rules of symmetry, transitivity, and distribution.

Coordinated entities share a common time stamp and synchronized clocks. Synchronization is a simple command for RuleBuilder. Coordination rules not only provide flocking behavior in entities, they can be applied to multiple participants, assuring, for instance, that classes of trainees all complete similar training experiences. The concept of coordination can be generalized to the existence of a measure relation between arbitrary attributes of multiple entities. RuleBuilder, for example, permits the linking of "all Blue entities" so that their behaviors might exhibit a group constraint.

Autonomous entities provide exploratory tools for emergent behaviors. RuleBuilder provides the capability of observing multitudes of similar entities as their behavior unfolds over time. The designer, rather than having to specify all behaviors concretely, can just watch for behaviors of interest, then specify the particular entity for further cloning and observation. These *genetic techniques* are computationally expensive, and we do not know the size and variation of populations that can be supported in realtime. Preliminary calculation suggests the EduSpace system is capable of supporting hundreds of genetic entities concurrently.

The most complex entity is one that is inhabited by a human participant. In *inhabited entities*, dynamic behavior is slaved to physical transducers attached to the participant. These signals are standardized to the participant's physiological body model. The physiological model is then mapped onto a virtual body, which can be an arbitrary representation. A participant teleoperating a robot claw, for example, may prefer his virtual hand also to be represented by a claw, even though the physiological model is of a hand.

Changing Spatial Entities

Space entities manage metrics, coordinate systems, location information, collisions, linearity, opaqueness, granularity, force fields, and winds. They determine the properties of the local environment. The VE-Chip is specialized to provide the computational resources needed by space entities.

Editing spaces is a substantially unfamiliar activity. However many spatial interactions require nothing more than specifying an equation which applies to a particular domain. For a simple (but not realistic) simulation of gravity, each behavioral cycle a spatial rule could subtract a constant amount from the height of every entity not in contact with the ground. It is a small step to then add more realism by providing acceleration. Each entity is made responsive to the time it is falling. As that time accumulates, an increasing amount is subtracted from its height above ground.

Of course, gravity and most field effects will be provided in EduSpace pre-constructed physics modules and accessible through a selection activity. The Space Editor then provides manipulation of parameters such as selecting the gravitational constant of the environment.

The Space Editor includes controls for determining the metric and granularity of the enclosing space. It is preferable to use an integer metric, for example, rather than a continuous real metric when distance comparison is to be very fast.

Editing Images

Images are edited by the MediaMaster tool, which organizes video and still imagery into the display environment. Images provide complexity in a display environment, but they are expressed by arrays of values for display rather than as collections of objects with behaviors. Since image processing and display is relatively advanced in the commercial marketplace, Oz is seeking a strategic multimedia partner to provide image editing and animation tools.

Image data can be imported without change from external sources such as a VCR. Images can also be created by projecting a three dimensional scene onto a two dimensional plane, in effect abandoning the additional spatial information. Finally, images can be composed from both internal and external sources to create unique multimedia experiences.

6.7 DOMAIN AND INSTRUCTIONAL MODELS

The Issue: Current instructional lesson plans are at best branching and rigid. They lack the formal structure which assures that a lesson simulation acts like the real world; they contain only rudimentary task and student models which can guide the interaction between student, teacher, and task.

EduSpace includes task and instructional modeling tools for both objects and processes. The task model provides realism to the simulated task being taught. The student model provides record keeping and a basis for the management of instructional dialogues. The situation-specific model provides both student and computer with a reliable common ground for interaction.

Qualitative domain models and formal process models provide structure for the EduSpace experience. The rulebase of EduSpace maintains a dynamic model of the task being learned and of the student's interaction with the lesson. The domain modeling techniques which will be integrated into the EduSpace product have been developed by Dr. William J. Clancey and his students at Stanford University and at the Institute for Research on Learning over the past eighteen years.

Two techniques, the numerical and the symbolic, dominate computational approaches. Numerical methods rely on the measurement of attributes. Symbolic methods are more general, relying on descriptions of relationships between attributes and situations. Rule-based logical programming is a symbolic technique, as are relational networks. By constructing formal models of events and situations, symbolic computation can be reliably used to develop models of instructional tasks. This approach is called *qualitative domain modeling*. When models are of events which occur over time, the approach is called *process modeling*. (Clancey, 1990).

Qualitative and process models are implemented within EduSpace entities. The modeling approach of this section defines a methodology for structuring entity behaviors. Each entity is seen as an encapsulated model of some object or process in the training environment, in the model of teaching, or in the interaction between the task model and the student. Entities also define the database partitions which modularize calls on computational resources and optimize pattern-based information retrieval. A *qualitative process model is therefore a formally structured entity*.

The key ideas in this section follow:

- A rule base, contains a general model of some system, which we call the *domain model*. Eduspace maintains domain models of the training task, the student, and the interaction between the task and the student.
- An inference procedure constructs a *situation-specific model* of the particular events and processes which occur as the student interacts with the task model.

- The domain model, the situation specific model and the inference procedure can all be represented as *relational networks*. The EduSpace Accelerator Board computes the changes in relational networks between and within entities.

The central architectural distinction is between:

the *general domain model* (how entities in the task domain normally appear and behave),

the *situation-specific model* (the particular entities in the world now being presented, manipulated, and reasoned about), and

the *domain database* (the particular entities in the world that are available to be presented, manipulated, or reasoned about).

The instructional designer first builds a general domain model of the skills to be taught. EduSpace provides specific interaction tools and a library of designed domains and entities to ease the entry and organization of the task related knowledge. While the student interacts with the elements of the task model, a situation-specific model is built by making modifications to the domain database.

In general, situation specific models enable the tracking of subtasks which have been started but are incomplete. The training software can then deliberately compare pedagogical alternatives, seamlessly integrating older lines of inquiry with new methods.

In terms of the implementation levels of the EduSpace architecture, situation specific models are maintained and manipulated within the match-and-substitute paradigm running on the EduSpace Accelerator Board. The relational network which describes the structure of each active world environment is processed in parallel by the accelerator. At the interface level, situation specific models are initiated by tools provided in the RuleBuilder and MediaMaster. The particular training scenario and script is maintained by the TaskMaster, while the resources of the entire system are maintained and distributed by TrainTrack, the session manager. Finally, the trainee's actual progress through the lesson is captured and measured by the student ProfileBuilder.

6.7.1 Interactive Process Models

A process is a structured object which evolves over time and which can be reasoned about. Reasoning about processes enables parallel execution, suspension and resumption of computational tasks, and dynamic ordering.

Different processes are represented differently. For example, modeling a student's errors, generating strategic hints, detecting that a solution is incomplete, and generating hypotheses about missing domain facts are all processes that are conceptually independent.

The rule base in an instructional program contains models of three different kinds of processes:

- the domain model (the subject material being taught),
- the inference procedure (how a model is used to solve problems), and
- the interaction model (how to communicate and interact with a student).

The objects, rules, and relationships of the subject matter form the *domain model*. The *inference procedure* is the set of algebraic and logical rules which control changes and interaction with the task model. The *interaction model* defines the rules of the interface and of the pedagogical model used to guide teaching. Inference and communication are models since they model how people solve problems and teach. The *task model* is the domain model plus the inference procedure, the facts and the way we use them. When the instructional courseware runs, these models are effective procedures; they enable EduSpace to solve problems and to communicate. Models are implemented and maintained as entities within the EduSpace system.

Domain, inference, planning, and communication form successively more general contexts for the processing and display of the situation specific model which defines the student's interaction with the lesson information.

What makes software reusable by different interpreting inference procedures is the ability to classify process constructs in different ways. Different interpretation procedures require different views of the model/process being reasoned about. These views take the form of new relations by which elements of the model/process are classified. In essence, *relations act as filters* by which different elements of the process model are preferentially collected and sorted for incorporation in a situation specific model.

Although we can model human reasoning using expert systems, there is always an aspect of creativity and improvisation to human performance that a body of facts and laws does not capture. Human experience is integrated with computation in EduSpace by making the participant an entity (the virtual body) and then maintaining a model of the participant within this entity.

6.7.2 Layers of Representation

Models of processes are represented as sets of rules and algebraic equations within the disposition of entities. Processes are modeled by rules which enforce a sequence of evaluation activities.

There are three levels of organization for qualitative modeling within the EduSpace software architecture. These models are organized in layers of increasing implementation detail. The *environment* forms the layer that the instructional designer works with. It is the most easy to understand, the most intuitive, and the most like natural behavior. The *calculus* is the necessary mathematical layer which mediates

between computation and cognition. The *implementation* layer is the specialized machine language required by the computer.

Environment, Task assumptions and constraints.

This level describes how the *task model* will be used, and the amount of detail required before action is taken. Tasks include the domain model and the inference procedures which define valid changes in the domain model. The environment includes database assumptions about symbolic information in the modeled world, and about particular cases, or examples, which define appropriate behavior. The environment also contains interactional data and communication constraints which impact the inference process.

Qualitative Calculus, Representations for the digital world.

This level contains the *conceptual primitives* which define the manipulable types of entities and their attributes. The calculus includes *relational networks* which structure the hierarchies and transitions between primitive entities. Finally, the calculus contains *process models* which provide views of causal, temporal, spatial, and subtype descriptions of states, substances, structures, and processes in the domain model. EduSpace SpecBoxes guide the instructional designer during the specification of tasks. The formal languages of a qualitative calculus are invisible to the designer at the interface level.

Implementation, Programming language.

The implementation includes the abstract tools provided by a computational language for describing the environment and the calculi which apply to entities in the environment. These tools include the specific techniques for describing and specifying entities and processes, such as frames, rules, objects, data sources, agenda, and metarules.

6.7.3 Inference

An inference procedure is not a list of facts or rules, but a program. It indicates how to do something, such as how to do diagnosis or how to teach. An inference procedure can also be called a *strategy*. It controls how knowledge is used. Inference procedures apply domain models to particular problems. There are as many kinds as inference procedures as reasons for wanting to use a model: design, diagnosis, repair, auditing, prediction, and control are the most typical reasons.

An *inference procedure*, then, is a program for gathering information about system behavior and the environment (data) in order to make assertions about the system producing this behavior (diagnosis), the system that could produce this behavior (design and control), or how a system will behave (prediction). Each inference procedure constructs a situation-specific model of some system.

There are four general inferential operators which generate transformations on processes:

- *diagnosis* is an analytic technique (describing processes in an existing system),
- *design* is a synthetic technique (designing a new system).
- *control* is a situation specific technique (describing how the current system should be configured, and possibly what the input should be, to produce desired behavior).
- *assembly* is a constructive technique (describing a manufacturing process within the model that will produce the desired new system).

Within EduSpace, inference procedures apply to the subject material (objects and events in the visible world), the instructional process (topics, initiatives, and outcomes), and the student's reasoning process (the student's actions and strategic explanations for them). In the EduSpace modeling tools, each inferential strategy is implemented as a partition of the dynamic database. That partitioned database represents a particular environment which follows a particular set of inferential rules. Behavior and computation in the database are achieved by match-and-substitute algorithms running on the EduSpace Accelerator Board.

6.7.4 Student Modeling

A *student model* embodies the student's goals, how the student has interacted with the instructional environment, the student's preferences, and what the program believes the student knows at any point in time.

In order to create an on-going student model, the EduSpace systems uses a transparent model called an *overlay model*. This model contains all the skills and knowledge the expert believes are important. Using powerful inference procedures, the system is then able to identify how much the student has learned while using the courseware, compared to the complete domain of what can be known about the topic.

Assisting this model is a *misconception model* which represents typical facts and procedures incorrectly believed or followed by students. The ability to identify faulty knowledge in the learner replicates how the expert teacher evaluates progress in individual learners.

Together, the overlay and misconception models are kept up-to-date as the student interacts with the program. The internal domain model is applied to determine what an expert would do in each situation. The student model is applied to interpret what the student is doing. How these models are used is briefly described below.

- For a given situation, the system determines what an expert would do next. This is used to provide hints to help the student, or suggest questions to probe his understanding.
- Given the student's behavior, the system updates the student model so it gives a consistent explanation of the student's capability (e.g., mark certain skill elements as known if they are manifest in the behavior, mark other skill elements as unknown if they are appropriate, but not manifested).
- Given a student's explicit statement of his beliefs (e.g., claims about appropriate goals or hypotheses about what is happening in the simulated world), the system updates the student model to be consistent with these beliefs.

Figure 6-4 illustrates the interconnectivity between the particular qualitative and process models which monitor student performance and potentially intervene with error correction and strategic advice.

EduSpace's optional guidance styles and dynamic, interactive branching allow a student to approach and explore new information in the way that is most natural for the person. This provides students with the opportunity to construct their own understanding of the task at hand.

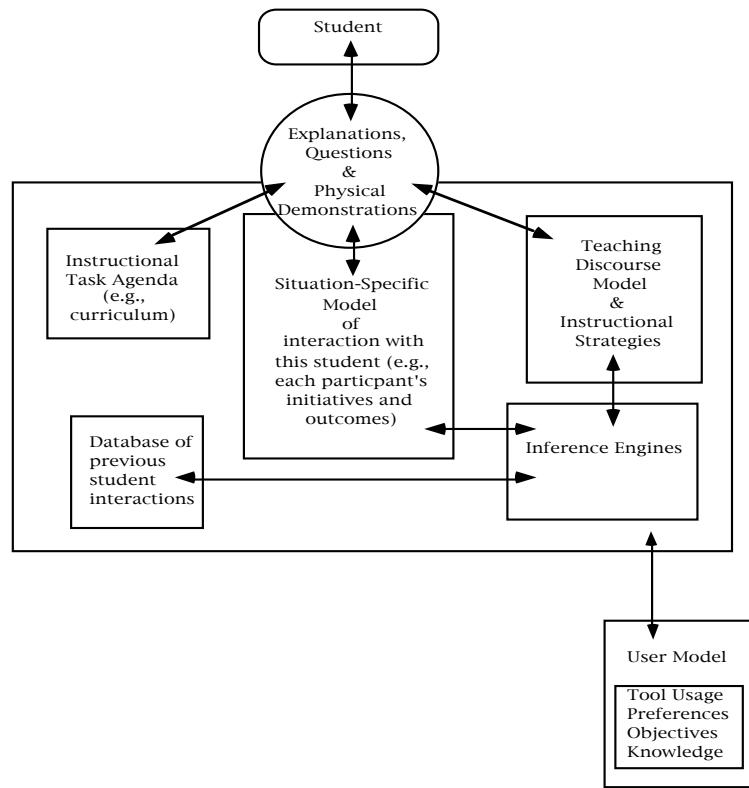


Figure 6-4: Architecture of the Monitor-Coach-Tutor

7. GLOSSARY

algorithm: Any specific method of solving a well-formulated problem, presented in a language that can be implemented on a computer.

artificial intelligence: A discipline concerned with the building of computer programs that perform tasks requiring intelligence when done by humans. However, intelligent tasks for which a decision procedure is known (e.g. inverting matrices) are generally excluded, whereas perceptual tasks that might seem not to involve intelligence (e.g. seeing) are generally included. For this reason, AI is better defined by indicating its range. Examples of tasks tackled within AI are: game playing, inference, learning, natural-language understanding, plan formation, speech understanding, theorem proving, and visual perception.

artificial reality: Coined by arts scholar Myron Krueger in the mid-1970s to describe his "computer-controlled responsive environments," which took an aesthetic approach to the human/computer interface. "An artificial reality perceives a participant's action in terms of the body's relationship to a graphic world and generates responses that maintain the illusion that his or her actions are taking place within that world." --M. Krueger, Artificial Reality II

ASIC: application specific integrated circuit

bandwidth: A range of frequencies in a spectrum that indicates a system's transmission capacity. The wider the bandwidth, the greater the amount of information that can be transmitted in real time.

beta testing: Software testing at selected customer sight prior to production release.

Boundary Mathematics: A formalism which uses spatial configurations to express the formal structure of mathematical systems such as logic, integers, and algebra. The central innovations in Boundary Mathematics are the use of non-representation, the void, to convey meaning, and the use of boundaries to represent structure. Traditional mathematical notations never use the void, and use textual strings to represent objects.

Boundary Logic: The application of boundary mathematics techniques to the representation of logic and to the computation of inference.

CAD: Computer-aided design. A CAD software program is a precision drawing tool that expedites the design process by providing versatility and automation during initial product or part design and revision processes. CAD programs also offer editing, printing, plotting, and data import and export functions.

CAM: Content addressable memory.

CBT: Computer-based training.

cyberspace: Coined by science fiction author William Gibson to describe a shared virtual environment whose inhabitants, objects, and spaces are comprised of data that is visualized, heard, and touched. It is "a consensual hallucination experienced daily by billions of legitimate operators in every nation, by children taught mathematical concepts...a graphic representation of data abstracted from the banks of every computer in the human system." --W. Gibson, Neuromancer

Virtual Reality occurs in cyberspace, but cyberspace does not necessarily involve virtual reality. It's been said that "cyberspace" is where you are when you're having a phone conversation. It's where your ATM money lives. It's the space through which electronic mail travels.

DataGlove™, VR glove: Gesture recognition device that permits the use of hand gestures to control activity and interaction in a virtual environment, including navigation by pointing and manipulation of objects. A lightweight glove lined with sensors that detect motion and send signals to the computer, the VPL DataGlove has fiber-optic cables along fingers and wrist and an orientation sensor for navigation and virtual object manipulation.

EduSpace™: The Oz instructional development and training environment, which includes interactive multimedia, inclusive video and audio display, rulebased software entities, and courseware development tools.

entity: An entity is a collection of resources that can accomplish a specific task and that provide intuitive behavioral interaction. EduSpace entities provide management of data, processes, and communications; they act as independent operating systems.

environment: In the context of VR, the environment is an inclusive model which a participant can experience from within.

ergocentric interface: Computer interface systems that record, respond to and amplify human behavior and intelligence. Traditional human-computer interfaces require people to adapt to the functionality of the computer. The mechanical metaphor is supported by calling people *users*. People using an ergocentric interface are called *participants*.

FDDI: Fiber distributed data interface.

fiber optics: Glass or plastic fibers over which modulated light pulses from laser or light-emitting diodes can transmit data. Fiber-optics offers much higher bandwidth than ordinary metal cables, and thus can transmit more information at a faster rate.

gesture: A physical movement by a person that conveys symbolic information. Usually refers to hand movements, but may not always require hands; for example, severely handicapped individuals may use foot or eye gestures to convey symbolic information.

GUI: Graphical user interface. A GUI includes windows, menus, icons, cursors, dialog boxes, control panels, virtual objects.

HMD, head-mounted display, inclusive display: A tightly coupled input and display suite for presenting 3D inclusive experiences. The unit is usually housed in a helmet with eyepieces that contain tiny video monitors. An optical system presents the video images at a comfortable viewing distance, usually in stereo. A tracking system records the orientation of the head, and controls the display of imagery so that the image appears as an environment. HMDs often include 3-D sound. The HMD is the most popular device for placing people in virtual worlds.

hyper: (-space, -text, -media) Nonsequential; includes links between two pieces of information. Hypermedia is electronic media (which could consist of text, graphic, sound, and video) containing links between various pieces of information within the work, allowing instant cross-referencing and browsing.

hyper-computer: Network-based parallel computer that brings multiple processors to bear on one or more tasks. The hyper-computer derives its power by accessing idle computers on the network, growing and shrinking depending on the network's unused capacity.

immersive: refer to inclusive.

input devices: Tools that transfer data into a form recognizable to the computer; common input devices include keyboards, position-trackers, speech recognition systems, joysticks, and VR gloves.

inclusive: A type of virtual reality in which the participant interacts within a virtual environment. Inclusion is often accomplished by stereoscopic head-mounted displays. In noninclusive VR such as a 3D monitor, the participant is "outside, looking in."

inclusive display: refer to head-mounted display

interface: A method of communication between two objects; an interface may be hardware or software. The keyboard is a textual interface between a human and a computer, for example.

Linda: A programming language that coordinates communication between computational processes, independently from the computations themselves, creating a generic parallel programming environment. Linda's power and expressivity result from its use of tuple space for interprocess communications and synchronization. Tuple space is a global, associative, object memory that holds tuples, or sequences of typed fields, where each field is an expression that has a value or a potential value.

modeling, surface and solid: Used in computer-aided design and the creation of virtual worlds, surface modeling helps conceptualize three-dimensional objects with complex irregular surfaces, such as a cuckoo clock, while solid modeling helps create the object and show its inner workings (a cuckoo clock cut in half to reveal its gears).

multimedia: The simultaneous presentation of a series of effects in more than two media, as by combining video images, flashing colored lights, sound recordings, etc.

navigation: In VR terms, navigation refers to moving the virtual body or perspective around inside the computer-generated environment.

network: (or "net") A system of software and hardware connected in a manner to support data transmission. A local area network might connect several dozen computers in one company; the Internet is a global network encompassing thousands of smaller computer networks.

node: Any device connected to a network; also, the point where such devices are connected.

OASIIS™: Optical And Speech Interactive Interface System. An input device that enables navigation and interaction within a virtual environment. OASIIS combines optical and speech pattern recognition tools to sense user movements and voice commands.

operating system: A master-control computer program that allows other programs, including application programs such as graphics software, to run.

output devices: Tools that transfer data and information from internal storage of a computer to external storage and displays.

parallel processing: Processing in which more than one particular process is active at any given instant, yet the result of several concurrent processes is coordinated.

pixel: A contraction of "picture element," or one point in a graphics image on computer display.

polygons: The building block of virtual environments. Virtual worlds are constructed of hundreds or thousands of colored polygons displayed on monitors or stereoscopic displays. The more polygons that a computer can generate and display per second, the more realistic the virtual world appears. Currently, standard workstation-powered virtual worlds are based on 1,500 to over 15,000 shaded polygons. "Lower-cost PC-based VR systems limit the graphic image of the virtual world to about 500 flat-shaded polygons to achieve a rendering rate of about 7 frames per second. (Alvy Ray Smith of Pixar has estimated that we perceive the equivalent of 80 million polygons at 30+ frames per second when we look at a view of the real world.)" --M. Bricken, *Cyberspace: First Steps*

reality: (1) The quality or fact of being real; (2) a person or thing that is real; fact; (3) the quality of being true to life; fidelity to nature; (4) Philosophy: that which is real.

real-time: (1) Time in which the occurrence of an event and the reporting or recording of it are almost simultaneous; (2) the actual time used by a computer in solving a problem, the answer to which is immediately available to control a process occurring at the same time.

situated learning: The theory that people learn in a context rather than in isolation. Situated information includes the activities of the physical body while learning, the events happening in the environment while learning takes place, the alternative agendas and expectations brought to the learning context by both student and teacher, and the personal, cultural, and historical events which have shaped the learner up until the time of learning.

space: In terms of virtual reality, "space" generally refers to dimensionality. Not to be confused with cyberspace.

3-D: Three dimensional; refers to an object's spatial position (x, y, and z axes). Visually, refers to a computer display that exhibits width, height and depth.

3-D sound: Reproduced sound that seems to emanate from points in front of, above, below, and to either side of the listener. 3-D sound systems can accurately recreate the original recording environment, providing an "acoustic photograph" of sound in an environment. 3-D sound systems trick your brain into perceiving the sound as coming from some place other than the speakers or headphones. This is possible due to recent advances in computer technology, digital signal processing, and psycho acoustic research.

UNIX™: Bell Laboratories Trademark. An operating system originally introduced by Bell Laboratories in 1971 for the DEC PDP 11 minicomputers. It was intended to provide a simple uniform environment in which relatively small numbers of users, with considerable shared interests over and above the fact that they were all using the same computer system, could collaborate on a shared project. UNIX has become very popular and has been implemented on a very wide range of systems, from stand-alone personal computer systems, through workstations and minicomputers, to mainframes and supercomputers. There has been some divergence's between a number of competing versions, and there is currently movement toward an internationally agreed definition of a UNIX standard.

VEOS: The Virtual Environment Operating System developed at the Human Interface Technology Laboratory at University of Washington. VEOS provides management of processes, data, communications, and entities in a virtual world.

virtual: (-reality, -community, -space, -world, - environment) According to Webster's, "being such practically or in effect, although not in actual fact or name." IBM started using the word in the late 1960s to refer to any nonphysical link between processes or machines, such as "virtual memory," random-access memory being simulated using disk drives. Currently used to describe computational environments at behave *as-if-real*.

voxel: A three-dimensional cell analogous to two dimensional pixels. Contraction of volume pixel.

VR, virtual reality: According to VR theorists William Bricken and Brenda Laurel, VR is, respectively, "an electronically mediated experience" and "a multisensory representation." As generally agreed, virtual reality refers to the ability to interact with data in a surrounding environment. Participants can enter and navigate through a computer-generated, 3-D world, changing viewpoint and interacting with objects within that environment. Virtual reality is characterized by *inclusion* (it feels as if you are "inside" it). Participants enjoy autonomy, or freedom to move around and manipulate virtual objects at will.

Today, in virtual reality, you can see and hear, point and move, pick up things and throw them, and sometimes touch and feel. The popular concept of VR involves interaction with a computer-generated world via head-mounted goggles, body suits, VR gloves and joysticks.

VRISC: Very reduced instruction set computation. Central processing units which rely on large numbers of instructions are easier to program but require more effort in silicon fabrication. Reduced instruction set computing (RISC) trades ease of microcoding for symmetric silicon layout and faster bulk computation. Very reduced instruction sets use only one computational activity implemented with massive chip symmetry.

8. REFERENCES

- G. S. Belkin and J. L. Gray, *Educational Psychology*, Wm. Brown: 1977
- M. Bricken, Virtual Worlds: No Interface to Design, in M. Benedikt (Ed), *Cyberspace: First Steps*, MIT Press: 1991
- M. Bricken, Summer Students in Virtual Reality: A Pilot Study on Educational Applications of Virtual Reality Technology, , in A. Wexelblat (Ed), *Softwhere*, Academic Press: 1992. Also HITL Technical Report
- W. Bricken, *Analyzing Errors in Elementary Mathematics*, Doctoral Dissertation, Stanford University: 1987a
- W. Bricken, *The Efficiency of Boundary Mathematics for Deduction*, Advanced Decision Systems Technical Report ADS-6824-1, 1987b
- W. Bricken, *VEOS: Preliminary Functional Architecture*, HITL Technical Report M-90-2, Univ of Washington: 1990
- W.J. Clancey, *Knowledge-Based Tutoring: The GUIDON Program*, MIT Press: 1987
- W.J. Clancey, Model Construction Operators, *Artificial Intelligence*, 53, Elsevier: 1992
- Edelman 1987
- J.D. Fletcher, *Multimedia Review*, Vol 2-1, Spring 1991
- D. Gelernter, *Mirror Worlds*, Oxford University Press: 1991
- M. Genesereth and N. Nilsson, *Logical Foundations of Artificial Intelligence*, Addison-Wesley: 1988
- G. Gery, *Making CBT Happen*, Weingarten Press: 1987
- W. Gibson, *Neuromancer*, Ace: 1984
- D.W. Johnson and R.T. Johnson, Interdependence and Interpersonal Attraction Among Heterogeneous Individuals: A Theoretical Formulation and a Meta-Analysis of the Research, *Review of Educational Research*: 1983
- M. Kreuger, *Artificial Reality II*, Addison-Wesley: 1991
- Lave and Wenger 1991
- B. Laurel, *Computers as Theater*, Addison-Wesley: 1991

S. Levy, , in P.S. Adler and J. Mincer, *Technology and the Future of Work*, Oxford Univ. Press: 1988

J. Mattoon, *Building a Framework for Future VR Systems*, Presentation at the 1991 DPMAEF Conference on Virtual Reality

J. Mincer, forthcoming

Patton 1991

Smoliar 1989

Success Magazine, September 1992 p. 27

Training Magazine, Industrail Training Report, October 1991 and November 1991

D. A. Turner, Miranda: A Non-Strict Functional Language with Polymorphic Types, in J. P. Jouannaud (Ed), *Functional Programming Languages and Computer Architecture*, Lecture Notes in Computer Science 201, Springer-Verlag: 1983

L. Vygotsky, *Mind in Society: The Development of Higher Psychological Processes*, Harvard Univ. Press: 1978

S. Wolfram, *Mathematica*, Addison-Wesley: 1988

Year 2001: High-Tech Service Report

1990 Military Manpower Training Report, cited in J.D. Fletcher, *Multimedia Review*, Vol 2-1, Spring 1991

9. APPENDIX: Courseware Development Examples

10. EXTENDED TABLE OF CONTENTS

| | |
|--|----------|
| CONFIDENTIALITY AGREEMENT | <i>i</i> |
| TABLE OF CONTENTS | 1 |
| LIST OF FIGURES | 2 |
| 1. INTRODUCTION | 3 |
| 2. EXECUTIVE SUMMARY..... | 5 |
| Figure 2-1: The EduSpace Advantage..... | 5 |
| 2.1 Oz PRODUCTS..... | 6 |
| OASIIS (Optical and Speech Interactive Input System) | 6 |
| The IDW-1 (Instructional Development Workstation)..... | 6 |
| Figure 2-2: EduSpace IDW-1 Developer Station..... | 7 |
| Figure 2-3: EduSpace PS-1 Presenter Station | 7 |
| The PS-1 (Presenter Station) | 8 |
| 2.2 PRODUCT FEATURES | 8 |
| The EduSpace Interface Chair..... | 8 |
| EduSpace Accelerator Board | 8 |
| EduSpace Development Tools | 8 |
| EduSpace Courseware..... | 9 |
| Proprietary Technology..... | 10 |
| 2.3 TECHNICAL SUMMARY | 10 |
| 2.4 MANUFACTURING..... | 11 |
| 2.5 MARKETS | 12 |
| 2.6 THE COMPANY | 12 |
| Company Strategy..... | 13 |
| Figure 2-4: Proforma Financial Highlights..... | 13 |
| 3. THE EduSpace PRODUCT | 15 |
| 3.1 COMPUTER-BASED TRAINING..... | 15 |
| 3.1.1 Current Training Courseware Development..... | 15 |
| 3.1.2 Current Training Delivery | 16 |
| 3.2 THE EduSpace APPROACH TO COMPUTER-BASED TRAINING. | 17 |
| 3.2.1 EduSpace Courseware Development and Delivery..... | 17 |
| 3.2.2 Educational Characteristics of EduSpace | 18 |
| Situated Learning | 18 |
| Social Learning | 19 |
| Individualized Instruction..... | 19 |
| 3.3 THE EduSpace TRAINING ENVIRONMENT | 20 |
| 3.3.1 The Instructional Designer's Interface | 20 |
| Figure 3-1: The Designer's Environment | 21 |
| 3.3.2 The Training Instructor's Interface | 22 |
| 3.3.3 The Student's Interface | 22 |
| Figure 3-2: The Student's Environment | 23 |
| 3.4 EduSpace COURSEWARE DEVELOPMENT TOOLS | 23 |

| | |
|--|-----------|
| Figure 3-3: EduSpace Instruction Components | 25 |
| Figure 3-4: EduSpace Development Tools..... | 25 |
| Figure 3-5: Elements of Eduspace Instruction Components..... | 26 |
| 3.4.1 The Five EduSpace Software Tools | 27 |
| TaskMaster..... | 27 |
| MediaMaster | 27 |
| RuleBuilder | 27 |
| ProfileBuilder..... | 27 |
| TrainTrack | 28 |
| 3.4.2 The EduSpace Graphical User Interface | 28 |
| Figure 3-6: EduSpace Lesson Organization Display..... | 30 |
| 3.5 THE EduSpace EXPERIENCE..... | 31 |
| 3.5.1 The Instructional Designer's Experience | 31 |
| Knowledge Acquisition..... | 31 |
| Starting a New Lesson..... | 31 |
| Defining Lesson Objectives..... | 32 |
| Specifying Training Tasks..... | 32 |
| Configuring Particular Activities..... | 33 |
| Figure 3-7: Lesson Display During Development.... | 34 |
| Organizing Media Elements..... | 34 |
| Specifying Teaching Methods | 34 |
| Assigning Interactivity Options..... | 35 |
| Building Interaction Rules | 35 |
| Building Behaviors with Graphs..... | 36 |
| Figure 3-8: Lesson Display of SpecGraphs | 37 |
| Figure 3-9: SpecGraphs for a Carburetor..... | 37 |
| Figure 3-10: SpecGraphs for Fuel Flow | 38 |
| Finishing A Lesson..... | 38 |
| 3.5.2 The Trainer's Experience..... | 39 |
| Setting Up the Session | 39 |
| Monitoring the Session..... | 40 |
| Evaluating the Session..... | 40 |
| 3.5.3 The Participant's Experience | 40 |
| 4. MARKETING and SALES..... | 43 |
| 4.1 THE MARKET | 43 |
| Figure 4-1: 1991 U.S. Training Market..... | 43 |
| 4.2 THE COMPETITION..... | 45 |
| 4.2.1 Computer-based Training Vendors..... | 45 |
| 4.2.2 Virtual Reality Vendors..... | 47 |
| 4.3 Oz COMPETITIVE ADVANTAGES | 48 |
| Figure 4-2: Competitive Product Features..... | 50 |
| 4.4 MARKETING STRATEGY..... | 51 |
| Large Domestic Corporations | 51 |
| Courseware Development Companies | 51 |
| Department of Defense..... | 52 |
| 4.5 ECONOMIC ADVANTAGES | 52 |
| 4.5.1 Salary Savings..... | 53 |
| Figure 4-3: Cost Comparison..... | 56 |

| | |
|--|-----|
| 4.5.2 Setup and Reconfiguration Costs | 56 |
| 4.5.3 Time to Floor..... | 57 |
| 4.5.4 Distribution..... | 57 |
| 4.5.5 Product Support..... | 58 |
| 4.6 COMMUNICATION PLAN..... | 58 |
| 4.7 CUSTOMER SERVICE | 59 |
| Figure 4-4: Oz Unit Sales Forecast..... | 59 |
| 5. THE COMPANY | .61 |
| 5.1 Oz CORPORATE DIRECTION..... | .61 |
| Goals..... | .61 |
| Strategy | .61 |
| 5.2 PRODUCT DEVELOPMENT | .62 |
| 5.2.1 Project Milestones | .63 |
| Figure 5-1: Product Development Phases..... | .63 |
| 5.2.2 Management Team | .64 |
| 5.2.3 Corporate Affiliates and Strategic Relationships | .66 |
| 5.2.4 Technology Agreements | .68 |
| 5.3 FINANCIAL SUMMARY | .68 |
| 5.3.1 The Corporate Structure..... | .68 |
| 5.3.2 Financial Projections | .68 |
| Figure 5-2: Oz Product Cost of Goods | .69 |
| Figure 5-3: Proforma Balance Sheet..... | .70 |
| Figure 5-4: Proforma Statements of Income..... | .71 |
| Figure 5-5: Proforma Cash Flow Statement..... | .72 |
| 6. EduSpace TECHNICAL DESCRIPTION..... | .73 |
| 6.1 TECHNICAL OVERVIEW..... | .73 |
| 6.2 INTERFACE HARDWARE | .75 |
| 6.2.1 The Interface Chair..... | .75 |
| 6.2.2 OASIIS..... | .76 |
| 6.2.3 Hardware Architecture | .78 |
| 6.3 COMPUTATIONAL HARDWARE..... | .78 |
| Figure 6-1: EduSpace Hardware System Architecture | .79 |
| 6.3.1 EduSpace Accelerator Board | .80 |
| Figure 6-2: EduSpace Accelerator Board | .81 |
| 6.3.2 VE-Chip and MS-Chip Computational Engines..... | .82 |
| 6.4 MATHEMATICAL APPROACH | .83 |
| 6.4.1 Boundary Logic | .83 |
| 6.4.2 Match-and-Substitute | .85 |
| 6.4.3 Dynamic Parallel Database..... | .87 |
| 6.5 SYSTEMS SOFTWARE..... | .88 |
| 6.5.1 Programmability | .88 |
| Figure 6-3: EduSpace Software System Architecture..... | .89 |
| 6.5.2 Data, Communications and Process Management | .90 |
| 6.5.3 EduSpace Operating Shell | .90 |
| 6.6 ENTITY SOFTWARE..... | .91 |
| 6.6.1 Entities | .91 |
| 6.6.2 Internal Processes of Entities..... | .93 |

| | |
|---|-----|
| 6.6.3 Software Tools | 93 |
| 6.6.4 Space Management..... | 94 |
| 6.6.5 The Virtual Body | 94 |
| 6.6.6 Multiple Participants | 96 |
| 6.6.7 Tools for Creating Entities | 97 |
| Changing the Static Form of an Entity..... | 98 |
| Changing the Dynamics of an Entity | 99 |
| Changing Spatial Entities..... | 101 |
| Editing Images..... | 101 |
| 6.7 DOMAIN AND INSTRUCTIONAL MODELS..... | 101 |
| 6.7.1 Interactive Process Models | 103 |
| 6.7.2 Layers of Representation..... | 104 |
| Environment | 105 |
| Qualitative Calculus | 105 |
| Implementation | 105 |
| 6.7.3 Inference | 105 |
| 6.7.4 Student Modeling..... | 106 |
| Figure 6-4: Architecture of the Monitor-Coach-Tutor | 108 |
| 7. GLOSSARY | 109 |
| 8. REFERENCES | 115 |
| 9. APPENDIX: Courseware Development Examples | 117 |
| 10. EXTENDED TABLE OF CONTENTS | 123 |