BENCHMARKING USER PERFORMANCE BY USING VIRTUAL REALITY FOR TASK-BASED TRAINING

Jason Lucas, Graduate Research Assistant,
Virginia Tech;
jlucas06@vt.edu

Walid Thabet, Associate Professor,
Virginia Tech;
thabet@vt.edu

ABSTRACT: Conveyor belts have a high accident and fatality rate associated with them because of the dangerous environment their constantly moving parts create. Because of this high fatality rate, different methods are being considered to improve current safety training methods. By looking at the principles of cognitive learning and what makes a computer-based training program successful, a safety training program using virtual reality (VR) is being proposed. The training program structure includes four steps in creating a comprehensive two phase training program that will train personnel on the required operational processes in an instructional-based phase and then test their abilities through an interactive task-based session that tracks the user's progress and choices, tallies points based on corrective actions taken, and gives immediate feedback and consequences to the user's actions. This paper focus on the task-based phase of the prototype development and what steps are taken in creating an individual exercise. One example exercise is described in detail from choosing the material that is tested, implementing the animations and coding using Right Hemisphere's Deep Creator and LISP coding, implementing the tracking methods, and the output file that has been designed to keep track of the user's performance.

KEYWORDS: virtual reality, safety, training, conveyor belt,

1. INTRODUCTION

Conveyor belts are an important form of bulk material transportation within the mining industry and offer a method for constantly transporting large amount of materials. With this large capacity and constant movement comes a risk of injury and fatality. Goldbeck (2003) reported that between 1996-2000 that there were 459 injury cases (fatal/non-fatal) due to surface belt conveyors; 13 cases (2.8%) were fatalities and 22 cases (4.8%) were permanent disabilities, 42% of the cases reported (including 10 of the 13 fatalities) occurred while performing belt maintenance and inspection. Goldbeck (2003) also reported that the cost of a fatality in a mine was estimated to be $1.02 million in 1986; which is equivalent to $1.9 million in 2007 according to the US Bureau of Labor and Statistics (2007) inflation value. This value was calculated by considering medical expenses, worker's compensation, accident investigation, loss of family income, and lost production value. Permanent disability accidents of less severity were estimated at $237,000 and each lost-time accident was also estimated at an average of $5,000 (Goldbeck, 2003) or a 2007 value of $440,820 and $9,3000 respectively (www.bls.gov). These statistics are the basis of research leading to the development of a new safety training program for belt conveyors.

The majority of the fatalities occur due to failure to follow safety procedures and guidelines specified during training. MSHA's Code of Federal Regulations (CFR) part 46 and 48 requires only 24 hours of training for new hires. This training is typically given as 4 hours of training before the employee starts with the other 20 hours within the first 60 days of work (Goldbeck, 2003). Training procedures and guidelines are not adequately addressed by the major professional organizations such as ANSI conveyor safety standards, OSHA regulations, and the safety engineering professional associations and are left to the owner/operator to fulfil MSHA's requirements (Shultz, 2002, 2003). Inadequate training by owners of both operation and maintenance personnel is one of the many factors that cause conveyor accidents. Current training programs are lecture-based and are merely dependent on slides and video presentations to convey safety guidelines and procedures. Without actually reflecting upon and experiencing the information presented, it may be quickly forgotten and its learning potential lost (Gibbs, 1988). To enhance the learning it must be tested out in new situations. Another drawback to current safety training methods is that consequences are inadequately experienced or not experienced at all. This is because the real environment would not allow the trainee to perform what-if scenarios on safety training.

Virtual reality (VR) offers a potential for a safety training program to reduce the number of injuries and fatalities through the representation of real-life events and environments digitally. VR offers a cognitive learning method for trainees to think through actions needed to complete tasks, act them out in a virtual environment, and come in contact with direct consequences while receiving immediate feedback. With the user having control over the environment and playing out tasks and activities, it is more likely they will be able to perform those tasks safely in real life situations. Testing methods can also track user performance and determine a user's skill level. It is this cognitive learning experience that places interest of the use of VR above other current methods of safety training within the mining industry for conveyor belts.

A VR-based safety training program for conveyor belts is currently being developed as a four step process (Fig. 1). Through data collection of site visits and literature review, specific training areas were defined. The training is then implemented in a program prototype that consists of two modules, an instructional-based module (Lucas, et. al, 2007a) and a task-based module. The instructional-based module introduces the data that has been collected and the training areas that have been defined to the user where as the task-based module then tests the knowledge of the user to help determine the user's skill level. Upon completion these modules will be evaluated by novice users consisting of faculty and students for ease of use and then by industrial professionals for feedback on accuracy and completeness. Based on comments and suggestions the system will be reviewed and revised.

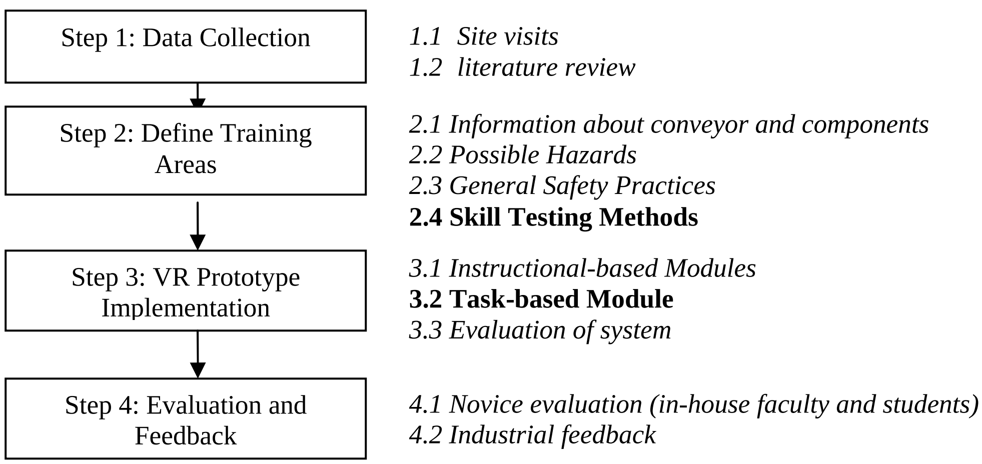

Fig. 1: Four steps to safety training program structure

The focus of this paper is on the second phase of the research. Within this phase a prototype implementation for the task-based module (step 3.2) is being developed. The paper also discusses the skill testing methods (step 2.4) that were used in development. Current training methods and why VR and cognitive learning can be beneficial to the industry as a safety training program. Current task-based training research being conducted with the use of VR is presented. It discusses in some detail the skill testing, how it is implemented in the task-based session, and how the user's actions are tallied and feedback is given to the user.

2. COGNITIVE LEARNING AND VIRTUAL REALITY IN TRAINING

Training has evolved from traditional classroom training to computer-based interactive training in many fields. Classroom training is still utilized and found desirable because of the knowledge that a quality training professional can instil in trainees during classroom sessions, however computer-based training is becoming more popular in educational, commercial and industrial sectors as it is believed to enhance the cognitive learning of the user. Cognitive learning is a process of acquiring knowledge and understanding through thought, experience, and the senses. The information that is acquired thorough cognitive learning undergoes a process known as cognitive information processing (CIP). CIP is based on the thought process behind the behaviour. The change in the behaviours is observed, but only as an indicator to what is going on in the trainee's mind. CIP is used when the trainee plays an active role in seeking ways to understand and process the information that he or she receives and relate it to what is already known and stored within memory. CIP seeks to explain how learning occurs in a "multistage" theory of memory. In this model of learning, the human mind collects information and moves it around to be associated with other existing related pieces of information and stored in larger files (Driscoll, 1994).

A similar model of cognitive learning is the integration of working memory and long-term memory. Working memory is the "center of cognition and scaffolds all the active thinking activities that occur" (Clark & Mayer, 2003). Working memory (Baddeley & Hitch, 1974) deals with the consciousness of our everyday lives. Where as long-term memory is the final stop in the cognitive information-processing path where the information is permanently stored (Shiffrin & Atkinson, 1969). Integration of new knowledge from working memory into long-term memory is called encoding. Later this knowledge from long term memory is pulled back into working memory to apply it within a real-life context. This is termed retrieval (Clark & Mayer, 2003). Thus the learning progress reaches completion when acquired knowledge is successfully retrieved and applied in multiple contexts. Meaningful learning ensures that well learned content held in long-term memory is easily brought into working memory. Retrieving and applying the data stored in long-term memory is crucial to solving high level problems involving complex interactivities (Ericsson & Kintsch, 1995).

These theories of cognitive learning can be applied to virtual reality. Within a VR training environment, the user goes through a series of training (instructional-based session) and rehearsals (task-based training). Unlike with traditional slide show methods and videos, it is an active learning experience where the user is in control. The situations that the user is put into virtual require them to think out proper actions. If improper actions are taken the consequences are immediately shown. These actions and reactions as the user goes through the system are placed in working memory until they are encoded into a larger file of similar information in the long-term memory. When the user comes across a real-life situation the appropriate knowledge is retrieved from long-term memory and applied to the situation (fig. 2) where the user would already know of the consequences of mishandling a situation.

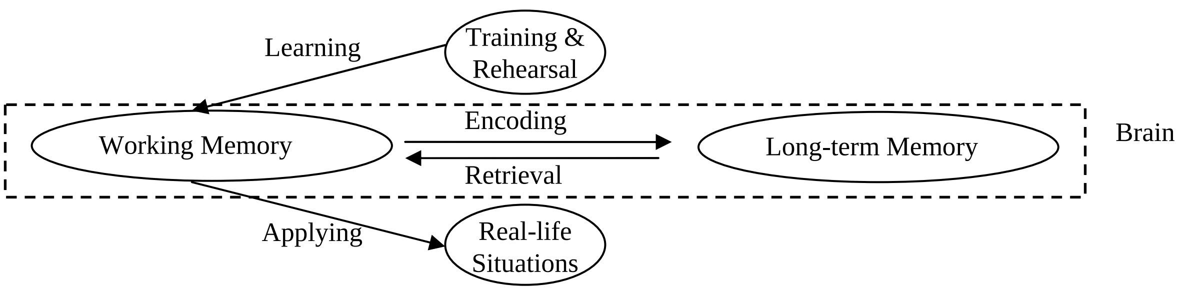

Fig. 2: Overall Steps for Second Phase Prototype Implementation

With the use of cognitive learning theory research has shown that computer-based training applications can enhance the training process and give the trainees a better understanding of the material when they are interactive and active in nature. Some work that has been done by using VR and computer-based representations for crisis simulations are: for fire-fighters (Querrec, et. al., 2001), mine safety training (Filigenzi et.al., 2000), security in a refinery (Haller, et. al., 1999), buildings' fire detection systems (Bukowsky, et. al., 1997) and shipboard fire fighting training (Tate, et. al., 1997). Some other examples of VR simulation that replicate an action/reaction scenario and task-based training methods within the mining industry have been conducted by the School of Mining Engineering at the University of New South Wales, Australia with simulated situations of safety in underground mining dependant upon an individuals actions (Stothard, et. al., 2004). Similarly, work at the Spokane Research Center (NIOSH) used VR to test the knowledge gained through training by giving trainees tasks within a VR environment to complete safely (Orr, et. al. In Press).

A VR simulation should replicate the work environment into which a trainee will be placed and then test the trainee with a problem based learning exercise (McAlpine & Stothard 2003). A simulation should also include a method of testing trainee's ability to identify and remedy risk with a method to quantify. In order to enhance the cognitive learning process, the simulation should also demonstrate to the trainee the consequences of poor decision-making and risk taking behaviour (Stothard, et. al., 2004).

With the models of cognitive learning coupled with the ability of VR to simulate a realistic working environment, a task-based session is being developed for safety training with the use of conveyor belts in mining. This prototype replicates a working environment, trains the user of the information that they will be responsible for, and lastly tests the knowledge by assigning a task to be completed, this process is described later in the paper.

3. TASK-BASED PROTOTYPE DEVELOPMENT

Based on the theories of cognitive and active learning, coupled with the features that have been observed as making a VR program successful, the task-based phase of the prototype development is being developed as part of four step structure for creating a complete safety training program. This task-based phase consists of multiple exercises that test the user's knowledge of the information that is distributed to them in the first, instructional-based, phase. Each exercise is designed by using a five part framework that evolves around instructional design theory and the ADDIE (analyse, design, develop, implement, and evaluate) model. A more advanced design model as described by Crawford (2004) is the Eternal, Synergistic Design and Development Model that revolves around the user's needs and their critique and feedback. Similarly, this research project is focused around the training needs of industry professionals and then is revised according to feedback and comments received from the industry, or the project's comments.

The five part framework for each exercise is as follows (fig. 3): determine skills to test, define training scenario, determine consequences and reward, animation and programming, and then implementation and testing for quality assurance. In each exercise it is essential to determine what the user will be tested on by forming a reasonable selection of activities to place within one exercise for the user to complete while in the virtual environment. The feedback given and questions asked to the user have to be defined with a tally system to keep track of the actions. Once the actions and reactions are defined, they have to be animated and programmed within the system, and lastly the system is put into implementation for testing and quality control.

analyze	**1: Determine skills to test** *(examples in italics)* • Safety procedures o *De-energize and lock-out & tag-out processes for belt. Replacing guarding after maintenance performance.* • Hazard Recognition o *Recognizing guarding requirements and cleaning excessive material.*	Information from step 1 is related to the information that is distributed to the user in Instructional-based Module. The information is received through site visits and literature review.
design	**2: Define Training Scenario** • Specify task that needs to be completed • Environmental Conditions • Define step by step activities o *Inspect belt, energize belt, check for proper performance*	Scenarios are defined through examination of accident statistics and what went wrong to cause the accident.
develop	**3: Determine Consequences and Reward** • Determine reactions to actions • Determine tally point system • Determine system questions and answers	Reactions (injury/ fatality) are judged on what is likely to happen in real life. Points are given on relative seriousness of action.
implement	**4: Animation and programming** • Animate what is needed to complete assigned task o *Animations and programming are required for energizing the belt, replacing components, removing hazards, etc.* • Program tallies • Design system output o *.txt file output file stating steps of process taken and points accumulated for recognizing hazards and correcting situations.*	Animations are completed using Right Hemisphere's Deep Creator. LISP coding is done within the program through the built-in LISP editor to aid in animations and tracking. The output is created with the user's name as a .txt file.
evaluate	**5: Testing and evaluation**	Implement, test, and revise. Industry professional review for quality assurance.

Fig. 3: Task-based Prototype Implementation Framework

When the session is created it is important that the session's objectives are clearly defined for the user. The main scenario needs to be defined as well as the environment that the user is working in. The objective will set out the tasks the user will have to complete, each task is then broken into sub-tasks. This will help in programming the needed animations and consequences accurately as well as help the programmer determine what questions will be asked about completing the sub-tasks. Once the tasks and sub-tasks are defined, the consequences and tally system can be determined. The tasks that the user will be required to complete fall under two general categories, those of "Safety Procedures, and those of "Hazard Recognition," table 1 lists some examples of these.

Table 1: Exercise 1 Organization.

Objective: Start the shift by inspecting the belt to ensure all safe guards are in place and all hazards removed. After energizing the belt, make sure it is running properly before going on break.	
Environment Conditions: Start of shift, the belt is de-energized, and there are no other employees in the area.	
Safety Procedures	**Hazards Recognition**
o Lock-out & Tag-out before performing maintenance. o Sound alarm when starting conveyor belt	o Ensure proper guarding o Ensure proper railings o Remove built-up material

Once the objective, environment conditions, and tasks are defined, a flowchart (fig. 4) is created to visually describe the processes that the user will go through. Included within the flowchart is the maximum number of points available for each action, the questions that are asked with each task with the possible answers, consequences for missed actions or wrong answers, and what animations need to be programmed within the system.

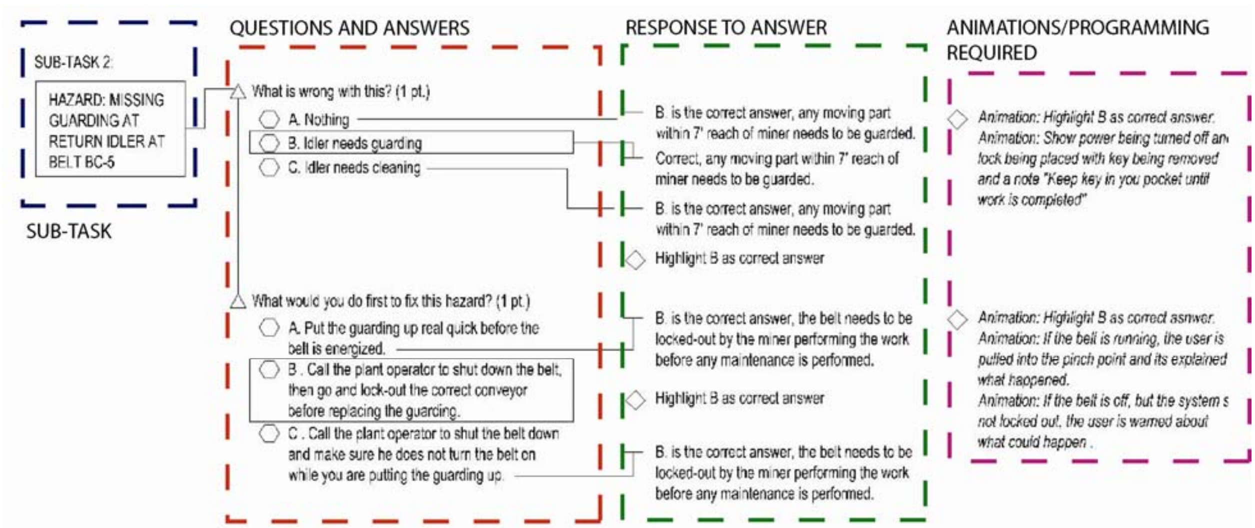

Fig. 4: Partial scenario flowchart

The example shown in Figure 4 shows an example of a sub-task within a developed session where the hazard of a missing idler guard exists. Each sub-task is listed in line on the left hand side of the flowchart. When that item is selected by the user, a series of questions are asked. The questions with the possible answers are listed to the right of the sub-task. Each possible answer is connected to a response to the answer including corrective actions and consequences for wrong answers. Lastly, what is being programmed and animated for each consequence and action is listed to the right of the response. The amount of points that the user receives for finishing certain tasks and answering certain questions varies based on the severity of the consequences if the action was not completed accurately. Analysis of accidents and case studies were used in helping to determine the severity in placing point values on actions. The analysis revealed that 56% of all conveyor related fatalities that occurred between 1995 and 2006 were due to entanglement (Lucas, et. al., 2007b In Press). With this in mind, the actions requiring the replacement of guarding are worth more than the amount of points available on removing built up material which could result in costly damage to the belt if the material interfered with the rotation of parts but not as likely to cause fatalities and serious injuries. The overall flow chart includes all of the tasks, sub-tasks, questions, answers, and possible consequences with the adjoining animations for each action. Once the flowchart is developed and the prototype can be implemented by using the flowchart as a storyboard and checklist for the needed animations.

4. SCENARIO IMPLEMENTATION

After the flowchart is completed the 3D environment is programmed and animated for each session. This step takes a completed conveyor belt model that has been developed in Autodesk's 3DStudioMax, exported as a .3ds file and imported into Right Hemisphere's Deep Creator. In Deep Creator basic animations, such as rotation of idlers and moving the mapped .jpg across the belt surface, are applied. These animations are then connected to a trigger animation object that turns the conveyor belt system on within the virtual environment. The trigger object is attached to the control room switch (Fig. 5). When the user clicks on the button to turn on the belt, the animations start, and the belt appears to be moving across the rotating idlers. Sounds that are applied to the environment are also turned on to give the effect of working within the actual environment. LISP coding is applied to the trigger animation object so that it is easier to follow what animations are connected to the trigger. LISP coding also allows for the check of variables that exist within the environment. One such variable is that if the power breaker is locked, meaning maintenance is being performed, it cannot be turned on until the lock is removed. This is achieved by the use of system variables and the use of "if" statements.

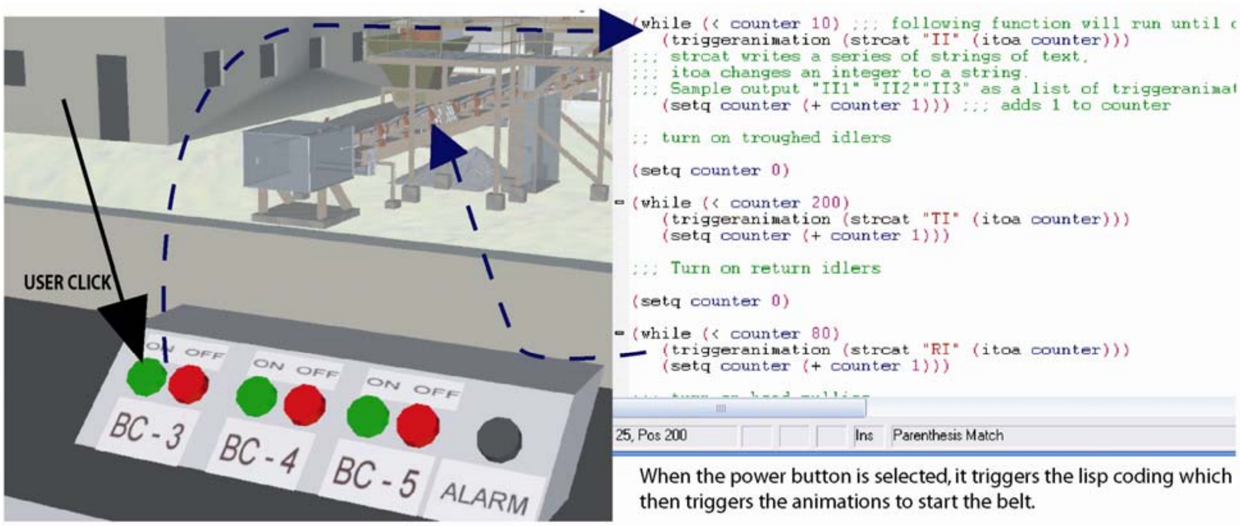

Fig. 5: Power Switch and Example of LISP Coding Used to Make Belt Turn On.

After the model is completed with animations to make it appear that it is operational, the consequences and rewards from the user's actions and questions that are answered have to be programmed into the system. LISP coding is again utilized to activate the proper animations and display messages to the user about their actions and the environmental conditions. One situation that relies on the system variables programmed with the LISP coding deals with an action of replacing missing guarding of the return idler (fig. 6). Before the conveyor is turned on the power lever is in the down position and the variable "lever" equals "0", and when the system is turned on the power lever is in the up position and the variable "lever" equals "1". When the variable is "0", points are rewarded and the "Guarding successfully replaced, +3pt." message is displayed. If the variable is "1" and the user clicks in the vicinity of the missing guard in order to replace it the user is digitally entangled between the idler and belt. This is done with a camera animation that takes the user into the idler and then drops them to the ground with an "Entanglement due to unguarded idler…" message appearing. After the animation the session starts over, keeping track of the termination that is then noted on the designed print-out.

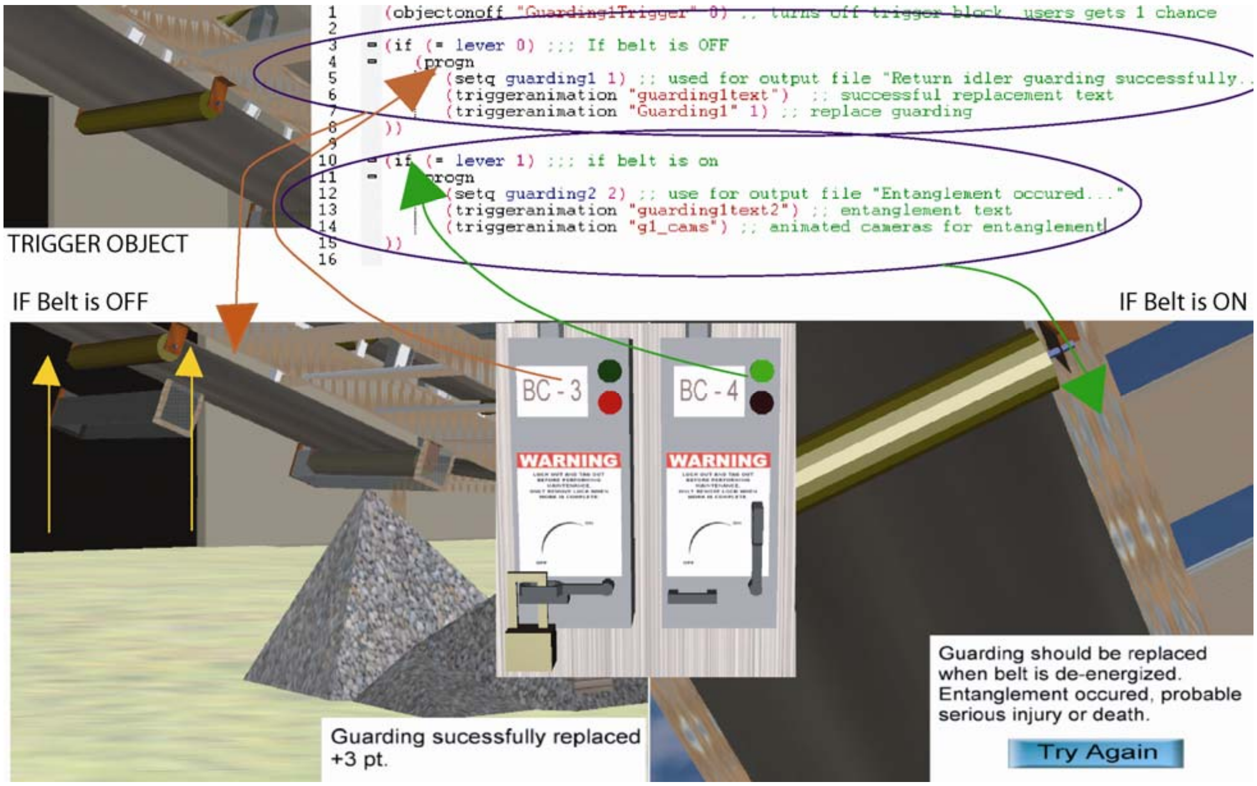

Fig. 6: Using system variables allows for different results when the same object is clicked by the user.

Other consequences are shown through the use of proximity type sensors. In order to accomplish this, two objects are needed, a plane, that is not rendered, to determine a collision point to trigger an animation, and a mass that represents the user within the environment. The mass is a cube, that does not render, that is placed at the location of the camera that moves with the viewport. When the mass and collision plane come in contact with each other, it triggers an animation. This situation exists when the user gets to close to a broken railing without first clicking on it to fix the hazard.

When all the proper system variables are put into place, the progress of the user can then be tracked through the LISP coding. A system variable is set up as a counter that tallies points after each successful step is taken and questions are answered properly. At the beginning of the session this counter is set at “0” and points are added to the sum upon completion of a sub-task, the new sum is then stored as the system variable. The total sum at the end of the session is then used to determine a percentage score for the user.

The final animations and programming required is to set up an output file (Fig. 7) that can then be saved and printed so an employer can keep track of their employees training and performance during training. In designing the output file, it was chosen that the name of the file will correspond with the name of the user. At the beginning of the session the user inputs their name into a text edit field, this is then stored as a variable which is used in creating a .txt file. The information that is included within the .txt file is the user’s name, the exercise title, task, environment conditions, questions and given answers, and the steps that the user took to complete the session. In order to determine what the user did and did not complete, the system variables are taken into a series of “if, then” statements that then write a string of text to the output file (highlighted in figure 7). At the end, a percentage based on the accumulated tally and the maximum available points are given and written as another string of text. This output file allows an administrator to view what the user has done without needing to watch or record each step as they are doing it.

```
Task Based Session 1: Morning Start-up

Task: Start the shift by inspecting the belt to ensure all safe guards are in
place and all hazards removed.  After energizing the belt, make sure it is
running properly before going on break.

Environment Conditions: Start of shift, the belt is de-energized, and there are
no other employees in the work area.

User: Jason Lucas

Hazards found:
Missing guarding at RETURN IDLER replaced +3 pts
Missing guarding at MOTOR not found, 0 pts (possible 3)
MATERIAL ACCUMULATION removed, +1 pt
BROKEN RAILING fixed, +2 pts

Fixing Broken Idler:
System locked out and tagged out before working, +3 pts
Guarding removed, +1 pt
Idler Replaced, +1 pt
Guarding replaced, +3 pts
System unlocked and powered on, +1 pt

Total 15 out of 18 points

Possible Injuries or Fatalities:
Improper guarding at motor can cause serious injury.  Any moving parts within 7'
reach of miner needs to be properly guarded.
```

Fig. 7: Sample from output file created at end of session.

The last step of creating an exercise is to have a test group go through it to make sure that all possible solutions were accounted for within the planning session, and to make sure all programming/computer bugs are eliminated before it is put into final implementation. The potential problem is not planning for, or leaving an outlet for, a common condition that can arise. This is where the test group of a variety of people with different backgrounds and experiences can be of tremendous benefit to the developers. Other problems can arise if too much information is being programmed into one exercise, mistakes can be made with trying to track the processes, and the user has a higher chance of becoming confused with what is being asked. This is why a series of exercises with various levels of difficulty covering the range of knowledge needed for proper and safe operation was chosen instead of a single, inclusive training session.

5. EVALUATION METHOD AND DISCUSSION

After its completion, the entire prototype will go through unstructured evaluations for accuracy and completeness by industry professionals. As the system is reviewed by the professions, comments and thoughts are recorded and suggestions are taken. As part of the professional feedback, the desk-top system will be examined at MSHA's TRAM Conference in October of 2007 in Beckley, West Virginia. The comments are then reviewed and the system is revised to make sure all valid comments are included to improve the knowledgebase of the system. The system is also going through structured evaluations with users of various skill-sets to determine the usability of the system and the effectiveness of the user interface. The comments and observations from these two study groups will be used to revise the system. The structured evaluation will examine the usability and benefit of a desktop interface and compare it to the CAVE (Cave Automatic Virtual Environment) at Virginia Tech, and a PowerPoint based presentation. Within this evaluation, each participant will go through one version of training and be given the same general knowledge test at the end of the session. Influences from skill level will be used to adjust scores through the use of a comparative test at the beginning of the session to judge the skill level. Finally, all of the comments and evaluation information will be used to revise the system (Fig. 8).

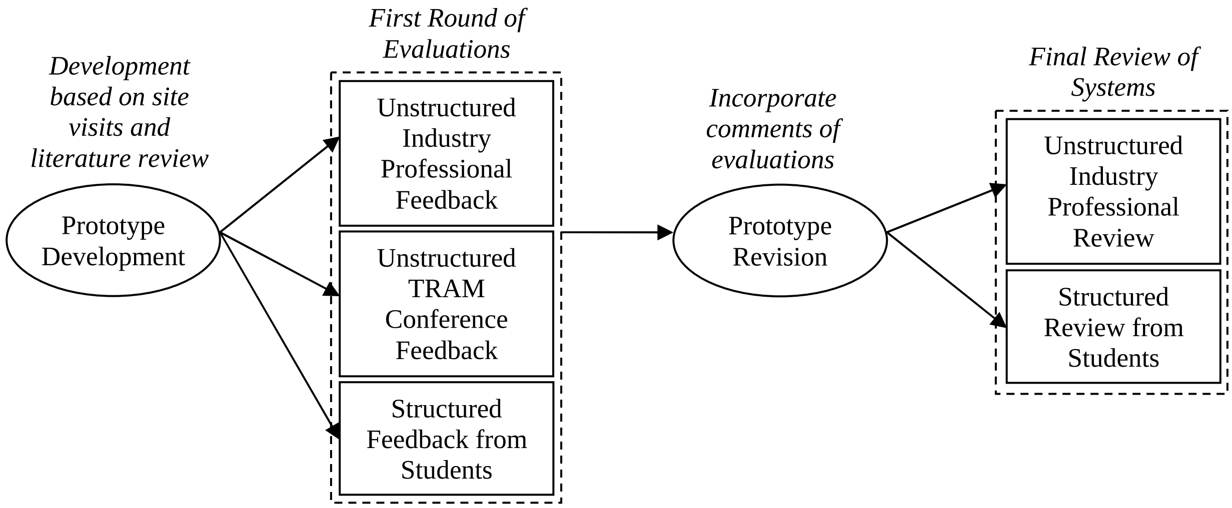

Fig. 8: Evaluation structure.

Once the system is revised, it will go through another round of structured evaluations. This will help determine the ultimate goal of the research, which interface is more beneficial to develop for a future training program. Based on past comparisons and initial discussions both the CAVE and desktop application are expected to be more beneficial then the typical PowerPoint training methods but a better placement of preference will be available after the first round of evaluations are completed.

It is hoped that this research can be a valuable alternative to classic training methods and validate the goals of the research to provide an effective method of training that can ultimately reduce the number of accidents and fatalities that occur due to the lack of training. The knowledge and face to face interaction that occur with a skilled training professional cannot be replaced, however it is believed that programs being developed in this research and future research can offer an invaluable tool to those training professionals to test the knowledge of users and provide a safe training environment to replicate real-life events.

This type of training exercise can cover a wide variety of activities that are required in working in a mining environment with conveyor belt systems. The flexibility that programs such as Deep Creator, coupled with basic programming in common languages such as LISP, allows for almost endless possibilities in training. When models of cognitive learning are employed in the development process of these exercises, allowing the user to make their own decisions in the learning process, the user can face the consequences and actions that would occur in real life if those actions have been taken. In conclusion, a VR based training that can replace on-the-job training with a safe and realistic environment that would otherwise be expensive to replicate, can be an extremely valuable asset in improving safety training for conveyors throughout the mining industry.

6. ACKNOWLEDGEMENTS

This research is being developed with support of a NIOSH Grant # 1 R01 OH008716-01 *Virtual Environment (VE) Applications to Improve Mining Health and Safety Training.* The views expressed in this paper are those of the authors and do not necessarily reflect those of NIOSH.

7. REFERENCES

Baddeley, A., & Hitch, G. (Eds.). (1974). “Working memory” (Edition 8). New York: Academic Press.

Bukowsky R., Séquin C. (1997) "Interactive Simulation of Fire in Virtual Building Environments", *Proceedings of the 24th annual conference on Computer graphics and Interactive techniques*, Los Angeles, California, 35-44.

Clark, R. C., & Mayer, R. E. (2003). *e-Learning and the Science of Instruction: Proven Guidelines for Consumer and Designers of Multimedia Learning*. Pfeiffer, San Francisco.

Crawford, C. (2004) “Non-linear instructional design model: eternal, synergistic design and development”, British Journal of Educational Technology, 35 (4): 413-420.

Driscoll, M. P. (1994). *Psychology of Learning for Instruction.* Massachusetts: Allyn and Bacon.

Ericsson, K. A., & Kintsch, W. (1995). “Long-term working memory”. *Psychological Review*, (102), 211-245.

Filigenzi M.T., Orr T.J., and Ruff T.M., (2000) "Virtual reality for mine safety training", Applied Occupational Health & Environment, 15(6) 465-469

Gibbs, G. (1988). *Learning by Doing: A Guide to Teaching and Learning Methods*. (London, Further Education Unit) The Geographical Discipline Network. Retrieved Sept. 20, 2007. http://www2.glos.ac.uk/gdn/publ.htm

Goldbeck, L. (2003) “Conveyor Safety and Education”. *Aggregates Manager*.

Haller M., Kurka G., Volkert, J., and Wagner, R. (1999) "omVR - A Safety Training System for a Virtual Refinery", Proceedings of ISMCR'99, Workshop on Virtual Reality and Advanced Human-Robot Systems, Tokyo, Japan, June

McAlpine, I. & Stothard, P.M. 2003, "Using multimedia technologies to support PBL for a course in 3D modelling for mining engineers", Proc. ED-MEDIA 2003 World Conference on Educational Multimedia, Hypermedia and Telecommunications, Honolulu, Hawaii, 24-28 June 2003, p2449-2455, publ. Norfolk, VA: Association for the Advancement of Computers in Education.

Lucas, Jason, Thabet W., and Worlikar, P. (2007a) "Using Virtual Reality (VR) to Improve Conveyor Belt Safety in Surface Mining", part of conference proceedings, 24th W78 Conference 2007 & 5th ITCEDU Workshop & 14th EG-ICE Workshop, "Bringing ITC knowledge to work" June 26-29, 2007, Maribor, Slovenia.

Lucas, Jason, Thabet, W., and Worlikar, P. (2007b) “A VR Based Training Program for Conveyor Belt Safety,” In Press, *ITCon.*

Orr, T.J., M.T. Fligenzi, and T.M. Ruff, “Desktop Virtual Reality Miner Training Simulator”, *International Journal of Surface Mining, Reclamation and Environment.* In Press.

Schultz, George A. (2003). “Training for Conveyor Safety”, *Materials Handling Management,* 58(11), 28-29.

Schultz, George A. (2002). “Conveyor Safety and Regulartions”. *Materials Handling Management,* 57(10), 18-20.

Shiffrin, R. M., & Atkinson, R. C. (1969). “Storage and retrieval processes in long-term memory”. Psychological *Review*, (76): 179-193.

Stothard, P.M., Galbin, J.M., and Fowler, J.C.W. (2004) “Development, Demonstration, and Implementation of a Virtual Reality Simulation Capability for Coal Mining Operations.” *Proceedings ICCR Conference,* Beijing, China.

Tate D.L., and Sibert, L. (1997) "Virtual Environments for Shipboard Firefighting Training", Proceedings of *1997 Virtual Reality Annual International Symposium* (VRAIS '97). 61.

U.S. Bureau of Labor and Statistics (2007) "CPI Inflation Calculator", http://data.bls.gov, retrieved on May 15, 2007. U.S. Department of Labor.

Querrec, R. and Chevaillier, P. (2001) “Virtual Storytelling for Training: An Application to Fire Fighting in Industrial Environment.” In *Proceedings of the international Conference on Virtual Storytelling: Using Virtual Reality Technologies For Storytelling* (September 27-28, 2001). O. Balet, G. Subsol, and P. Torguet, Ed. Lecture Notes In Computer Science, vol. Springer-Verlag, London, 201-204.