Software Specification of
A Mining Truck Simulator and Trainer

Frederick C. Harris, Jr., Yan W. Ha, Dianne M. Yumul, Joshua S. Estes, Christopher E. Miles

Department of Computer Science
University of Nevada, Reno, NV 89557
Email: fredh@cs.unr.edu

Abstract

In the surface mining industry, the cost of workplace-related accidents is high in terms of downtime and equipment repair. Training operators how to safely inspect and operate their vehicles is the major part of any training program. Traditional training methods are often expensive, ineffective, or even life-threatening. As a result, the mining industry is constantly searching for better training methods for their employees. This paper outlines the motivation for, and the development of, a virtual reality (VR) based Mine Truck Simulator and Trainer, which will cut costs and provide a risk-free interactive learning environment. It further discusses the specific requirements, functional and non-functional, with the aid of the Unified Modeling Language (UML). Directions of further research and development are also described.

Keywords: Software specifications, UML, VR, Mining, Simulator.

1 Introduction

Mining trucks are the leading source of serious injuries in the Mining Industry. Since 1994 mining vehicle accidents have been the primary cause of fatalities in the industry and in 2001 mining vehicle accidents accounted for more than fifty percent of the fatalities. The Mining Safety Health Administration has voiced their concern, stressing the need for and also requiring extensive training of mine vehicle operators to reduce workplace accidents. Many mining companies have responded to these new ideas by introducing new management philosophies [1]. But the instruction available, in the form of videotapes, manuals, and seminars, are not sufficient as they lack the trainee-to-vehicle interaction that is vital in developing the skills required to safely operate the equipment. The actual driving of the mining truck by the trainee, which is another option available, is dangerous if done in a production setting, or it is expensive if the vehicle must be taken out of production.

Our software, the Mining Truck Simulator and Trainer (MTST), will be an efficient and cost-effective alternative to train drivers to safely operate a mining truck. MTST is Virtual Reality (VR) based to provide a realistic and interactive environment. VR training tools have recently become popular because of their effectiveness and flexibility [2]. Studies have also shown that VR trained personnel perform required tasks to the standards sooner and maintain that ability [3].

Two previous VR based projects developed at the University of Nevada, Reno, are the Mine Vehicle Inspection Simulator (MVIS) and the Mine Vehicle Driving Simulator (MVDS) [4, 5]. MVIS presented a variety of 3D mining vehicles, and gave the user the ability to interact with the vehicles in order to inspect them for possible flaws. The MVDS provided a virtual world to train drivers. This project's achievements are the basic physics model of the mining truck, the modeling of the open-pit surface mine, and the dynamic hazard placement. Our project is closely related to MVDS, but we intend to develop more flexible software by implementing a physics model for the truck, a realistic representation of an open-pit surface mine, a networking interface to allow multiple users, and an intuitive graphical user interface.

This paper presents our project, the Mining Truck Simulator and Trainer (MTST) and describes, with the use of the Unified Modeling Language (UML) [6], its requirements and use-case specification. UML is used for specifying, visualizing and documenting artifacts of a software system. It allows the software system to be modeled as a set of interacting objects, making it applicable to object oriented systems and business processes.

The rest of this paper is organized as follows: in Section 2 we provide a general description of the software. In Section 3 we list the functional and non-functional requirements of MTST. We provide the Use Case model in Section 4, and the class diagram in Section 5. Finally, we cover the conclusions and future extensions to MTST in Section 6.

2. General Description

The goal of MTST is to provide a virtual world where trainees can perform typical driving tasks involved in open-pit surface mining. A number of scenarios will be available for the simulation to replicate different conditions and settings found in various open pit mines. We will integrate an intuitive graphical user interface to make the editing of a scenario as well as start-up and termination of the training session easy for the trainer and trainees. The training session will be conducted under the supervision of a trainer who can monitor, evaluate, and communicate with the trainees during the session. A networking interface will allow multiple trainees to perform various tasks in coordination with each other in the same open pit mine.

Before the training session, a scenario can be chosen for modification by the Scenario Manager. The Scenario Manager can place various dynamic and static hazards into the mine by simply selecting a hazard from a list and identifying the desired location. Other options available are setting the start and finish points, the load and unload locations, as well as setting the simulation weather.

When the Main Menu appears, it will list the options available to the trainer and trainee. The trainer will be able to initialize the session, select a scenario (which includes a mine and other hazards), and start the session. The option to initialize the session establishes the network connection, which enables multiple trainees to join the session as well as enabling trainer—trainee communication. The option to start the session closes the Main Menu and starts the Simulator. The Simulator enables the trainer to monitor and evaluate the performance of the trainees, and communicate with them through a chat box.

The trainees must join the session in the Main Menu, which transfers them to the Simulator. The Simulator will then present a virtual world and enable interaction in real-time. The virtual world will be the open-pit surface mine selected by the trainer. It will include a variety of weather effects, such as rain and fog, and hazards present in that environment. Inside the virtual world, the trainees must accomplish typical tasks, which consist of driving the truck, loading and unloading rock and soil on the specified locations, maneuvering difficult ramps and encountering obstacles. In order to make the driving as simple and realistic as possible, a steering wheel and pedals are utilized as the trainees input devices. However, the keyboard and mouse (or a joystick) can also be used. The accurate physics of the truck will be modeled for the simulation so as to imitate its actual acceleration, deceleration, and turning radius.

At the end of the simulation, the trainer will have the option to browse through the performance reports of the trainees, print the selected reports, and terminate the training session. Upon termination or shutdown, the network connection will be closed and communication will be disabled. The Main Menu will then appear for the next session.

All the above functionality will be implemented with the aid of the Torque game engine [7]. Torque is a professional development platform which features TCP/UDP networking capabilities, a 3D visual API, physics and collision models, and a C++ like scripting language for the development of a virtual world.

3. Requirements Specification

The Mining Truck Simulator and Trainer will offer many capabilities and will have an intuitive interface to make it easier for the end users of the product. The *Functional Requirements* lists the behavior of the system, while the *Non-functional Requirements* lists a number of specific properties the software should have [8].

3.1 Functional Requirements

MTST's most important functional requirements are listed below. Nevertheless, we believe they could provide a fairly complete picture of the tool's capabilities in its current, initial version. For traceability during the software development process each functional requirement has a number and is denoted using the format <R#>.

R1 The system shall provide an interface for the scenario manager to create a scenario and save the scenario.

R2 The system shall provide an interface for the scenario manager to edit a scenario and save the edited scenario.

R3 The system shall provide an interface for the trainer to select a scenario for a training session.

R4 The system shall provide an interface for a trainee to join the training session.

R5 The system shall provide a multiple-user networking environment for group training sessions.

R6 The system shall provide a monitoring system to alert the trainer if triggered by pre-defined events, such as hitting a hazard.

R7 The system shall provide a monitoring system to monitor the session, generate a report, and send a report to the trainer.

R8 The system shall provide a text-based communication interface for the trainer to communicate with the trainees, and for the trainees to communicate with each other.

R9 The system shall provide weather effects, such as fog, and rain.

R10 The system shall provide a first person view from inside the truck.

R11 The system shall provide a responsive instrument panel.

R12 The system shall provide a means to drive the truck.

R13 The system shall provide objects to run into.

R14 The system shall provide a means to load the truck.

R15 The system shall provide an interface for the trainer to initialize the session.

R16 The system shall provide an interface for the trainer to start the session

R17 The system shall provide a means for the trainer to enable a chat during the session.

R18 The system shall provide an interface for the trainer to end the session and shutdown the program.

3.2 Non-functional Requirements

Non-functional requirements for MTST include primarily constraints on the environment's implementation. Following is the list of these requirements, denoted using the format <T#>:

T1 The system shall use the Torque Engine.

T2 The system shall run in real-time.

T3 The system shall be realistic in every feasible aspect.

T4 The system shall run on Mac OS X, Linux, and Windows.

4. UML Specification

Based on the above lists of requirements, the UML-model of MTST has been built. In the following we present excerpts from both the behavioral part (use cases and scenarios) and the structural part of the MTST (part of the system's larger class diagram).

4.1 Use Cases and Scenarios

The Use Case diagram is another way of presenting and specifying the requirements of a system or software. It shows two types of objects, the actors and the use cases. The actors are the characters played by the people or other systems that use the software, in our case we have the Scenario Manager, the Trainer, and the Trainee. The use cases are the various functions and tasks the system performs. Because UML is flexible, it allows for more than one use case diagram to expand a particular use case.

Using the UML-notation, the UP-based approach and the guidelines presented in [8] we have drawn the use case diagram of MTST. Figure 1 shows the main use case diagram. Each use case has sub use cases, for example the sub use case for the Simulator is illustrated in Figure 2. We have also detailed all use cases (examples are given in Fig. 3 and Fig. 4), and specified a comprehensive set of scenarios (examples are given in Fig. 5 and Fig. 6). Based on these use cases and scenarios, other UML diagrams that describe system behavior and structure have been drawn, including the class diagram from which an excerpt is presented in Fig. 7.

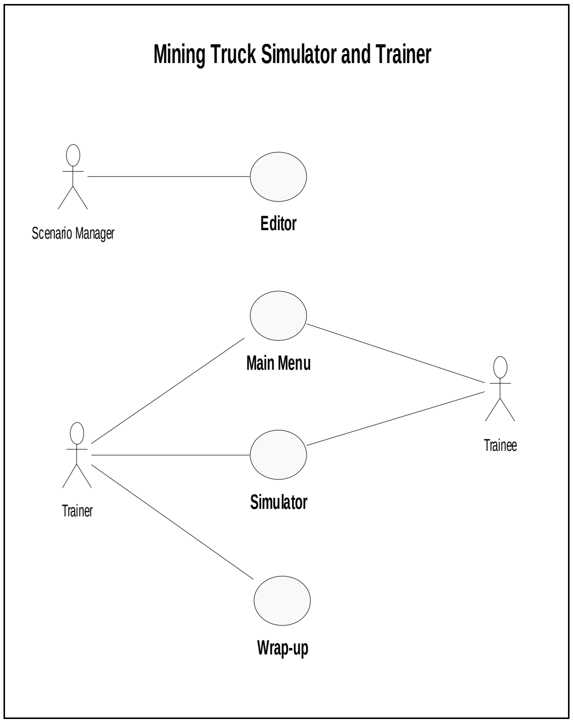

Figure 1: The MTST: Use Case Diagram

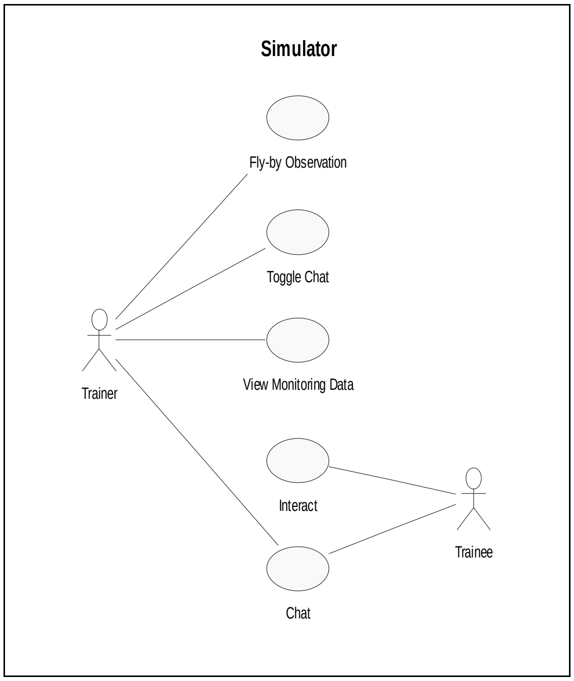

Figure 2: The Simulator: Use Case Diagram.

Use case: Main Menu—Start Session
ID: UC6
Actors: Trainer, Trainee
Preconditions: 1. A session is open via UC4. 2. An active scenario has been chosen via UC5. 3. All trainees have joined the session via UC6.
Flow of Events: 1. The trainer selects start. 2. The trainer moves from the main menu into the simulation 2.1. The trainer is presented with the simulation environment. 3. The trainees move from the main menu into the simulation. 3.1. The trainees enter the virtual environment 3.2. The trainees are given control of trucks placed at the appropriate starting locations. 3.3. Monitoring begins.
Postconditions: 1. The system transitions to simulation mode for all actors. 2. The simulation is active. 3. Monitoring is active.

Figure 3: Main Menu—Start Session Use Case

Use case: Simulator—Interact

ID: UC11

Actors: Trainee (driver)

Flow of Events:

1. The driver receives instructions from trainer.
2. The driver drives the virtual truck in order to complete the instructions.
    2.1. The driver interacts with the virtual world, navigating ramps paths and a variety of obstacles.
    2.2. The driver may approach the loader and receive cargo.
    2.3. The driver may go to the dumping location and dump the cargo.
    2.4. The monitoring system records statistics about the driver's actions.
3. The simulation ends.

Postconditions:

1. The trainer receives the overall summary of the monitoring system.

Alternative flow:

1. At any point the driver may quit the session.

Postconditions:

1. The virtual world has changed appropriately.

Figure 4: The Simulator—Interact Use Case

Scenario for Use case: Simulator—Interact

ID: UC11/S01

Actors: Trainee (driver)

Primary Scenario:

1. The driver receives instructions from trainer.
2. The driver drives the virtual truck in order to complete the instructions.
    2.1. The driver interacts with the virtual world, navigating ramps paths and a variety of obstacles.
    2.2. The driver approaches the loader and receives cargo.
    2.3. The driver goes to the dumping location and dumps the cargo.
    2.4. The monitoring system records statistics about the driver's actions.
3. The simulation ends.

Secondary Scenarios:
IncorrectResponse & DriverDisconnect

Postconditions:

1. Trainer receives monitoring system summary.
2. The virtual world has changed appropriately.

Figure 5: Primary and Secondary Scenarios for the Simulator—Interact Case

Scenario for Use case: Simulator--Interact

Secondary Scenario: IncorrectResponse

ID: UC11/S02

Actors: Trainee

Preconditions:

1. The simulation is active.

Secondary Scenario:

1. The driver receives instructions from the trainer.
2. The driver drives the virtual truck in order to complete the instructions.
    2.1. The driver interacts with the virtual world, navigating ramps paths and a variety of obstacles.
    2.2. The driver collides with a hazard or obstacle.
    2.3. The driver does not receive cargo.
    2.4. The driver goes to the wrong dumping location.
    2.5. The monitoring system records statistics about the driver's actions.
3. The simulation ends.

Postconditions:

1. Trainer receives the overall summary of the monitoring system.
2. The virtual world has changed appropriately in response to the driver’s commands.

Figure 6: Secondary Scenario for the Simulator—Interact Case

4.2 Class Diagram

Due to its dimensions, we present in Fig. 7 only a part of the Mining Truck Simulator and Trainer’s class diagram (analysis level). Also due to its dimensions we have placed it later in the paper. Complete details on attributes and, especially, on methods are fleshed out in a larger class diagram (design level), whose dimensions preclude its inclusion in this paper. Nevertheless, we believe the excerpt from the class diagram presented in Fig. 7 provides a good indication of the complexity of MTST.

5 Future Extensions

There are numerous possible extensions to MTST to enhance its usability and capabilities. Specifically we consider extending MTST’s capabilities with the following:

- Providing a variety of mining vehicles for trainees to operate.
- Better physics models with parameters for all of these new vehicles.
- A better monitoring system such that the trainer can tabulate what the trainee has done as well as interact with the trainees.
- More obstacle hazard models that the scenario manager can use.
- Sound for better immersion in the VR environment and force-feedback steering wheels.
- Better instrument panel design and operability that provides more feedback to the trainee.
- Mechanisms to generate summary reports on the trainees as well as a database to collect these.
- More mine models as well as tools to convert them into an MTST format.

6 Conclusions

We have presented in this paper a Mining Truck Simulator and Trainer, an environment intended to provide a comprehensive and effective training tool for drivers of large mining vehicles. This VR environment will provide better experience with the most common issues that trainees face, such as communicating with the coordinators and the peers, maneuvering through the mines, navigating through traffic, dealing with weather, and overcoming the difficulties springing from the size of these vehicles. In order to ensure a sound, rigorous development of the MTST software a systematic construction process has been followed, leading to the creation of a software model whose main components at the specification level have been presented in this paper.

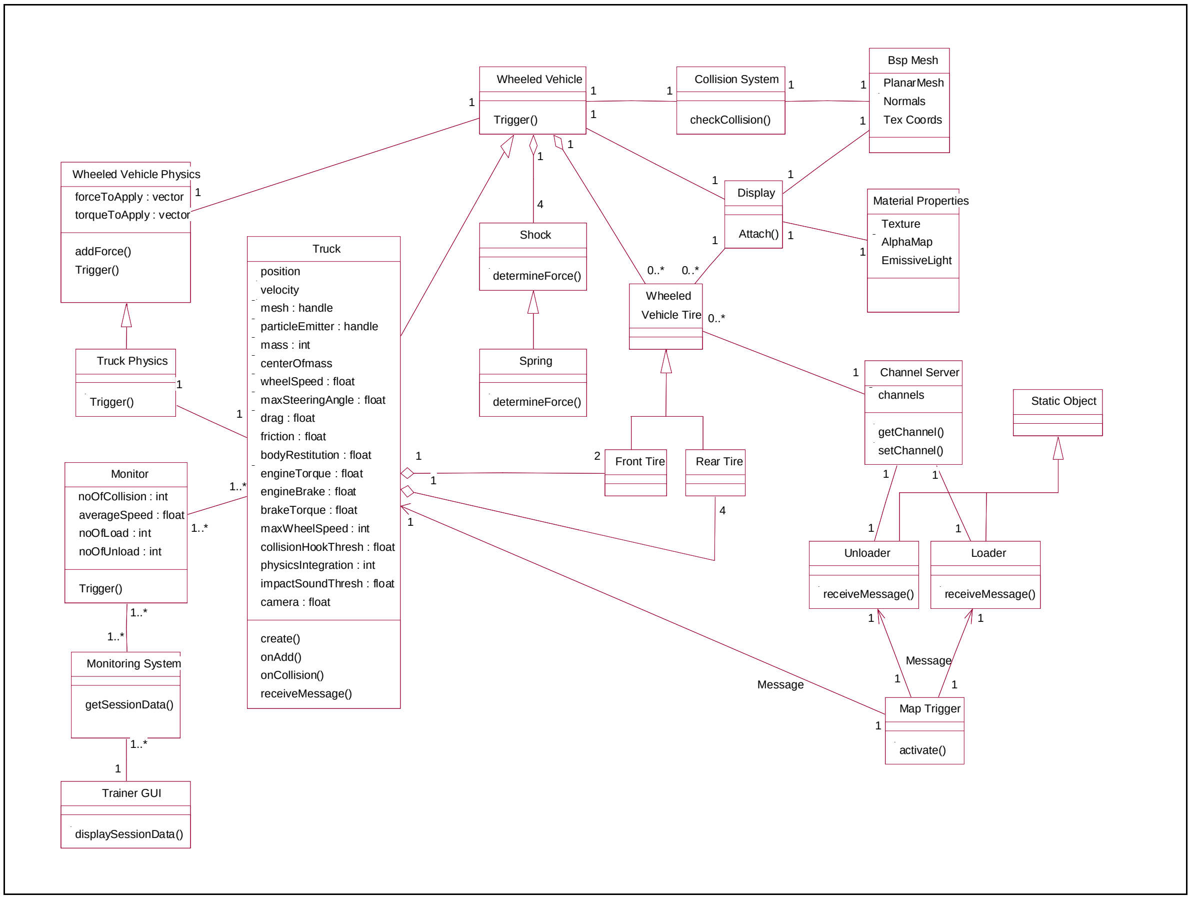

Figure 7: The MTST Class Diagram (partial)

A number of possible extensions have also been described in the paper in support of our belief that the Mining Truck Simulator and Trainer will prove to be a very practical tool for training new and experienced mining vehicle operators.

At this point in time, the Mining Truck Simulator and Trainer's UML analysis model has been completed and the tool's implementation is undergoing. The first working version of this environment is expected to be available in a few months. On a longer time frame, advanced functionality along the lines presented in Section 5 will also be incorporated in the simulator.

References

[1] B. G. Staley, "Culture Shock-Changing Attitudes to Safety in Mines." In *Proc. Of Safety, Hygiene and Health in Mining*, pp. 263-273, Harrogate, England, 1992, Institution of Mining Engineers.

[2] James D. Foley, Andries van Dam, Steven K. Feiner, John F. Hughes, and Richard L. Phillips. *Introduction to Computer Graphics*. Addison Wesley, Reading, MA, 1997.

[3] CAE available at http://www.cae.com, accessed April 24, 2003

[4] Damien Ennis, Benjamin Lucchesi, Nerissa Oberlander, Keith Wesolowski, Frederick C. Harris, Jr., and Pierre Mousset-Jones, "Surface Mine Truck Safety Training: A VR Approach to Pre-Operational Vehicle Inspection" in Kadri Dagelen, editor, *Proceedings of APCOM'99 Computer Applications in the Mineral Industries 28th International Symposium*, pp. 811--818, October 20-22, 1999, Colorado School of Mines, Golden, Colorado.

[5] Benjamin Lucchesi, Nerissa Oberlander, Frederick C. Harris, Jr., and Pierre Mousset-Jones, "Surface Mine Truck Safety Training: Scenario Setup for a VR Driving Simulator," in Q. Yang, editor, *Proceedings of the 12th International Conference on Computer Applications in Industry and Engineering (CAINE '99)*, pp. 62-65, Ramada Inn Downtown, Atlanta, GA, November 4-6, 1999.

[6] OMG's UML Resource Page, available at: http://www.omg.org/uml/, accessed March 14, 2003.

[7] Garage Games, Developers of the Torque Game Engine, available at http://www.garagegames.com/, accessed April 24, 2003.

[8] J. Arlow and I. Neustadt, *UML and the Unified Process: Practical Object-Oriented Analysis & Design*, Addison-Wesley, 2002.