DEVELOPMENT OF VIRTUAL WORLDS IN AGRICULTURE AND NATURAL RESOURCES FOR SIMULATION EXPERIMENTS AND ELEARNING USING ONTOLOGY-BASED SIMULATION

By

YUNCHUL JUNG

A DISSERTATION PRESENTED TO THE GRADUATE SCHOOL OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

UNIVERSITY OF FLORIDA

2013
To my advisor, committee members, my parents and my fiancée
ACKNOWLEDGMENTS

It has been a long journey since beginning my studies at the graduate school level. I always liked to pursue new topics and to learn new technologies for the future, and I have finally reached the end of my journey.

I would like to sincerely thank my advisor, Dr. Howard W. Beck. It was lucky for me to meet and collaborate with him. His ideas and advice have stimulated me to achieve my research objectives. His immense passion for research and intellectual insights were great motivating factors in my work.

Secondly, I would also like to extend my appreciation to committee members for their encouragement and guidance to complete this work. Dr. Michael E. Bannister introduced the concept of the learning forest system, and I am very thankful to him for directing the Virtual Learning Forest project sponsored by National Institute of Food and Agriculture (NIFA). Dr. Greg Kiker has guided me to understand systems based on natural resources and systematic approaches. Dr. Benjamin C. Lok has given valuable advice by seeing the usability of my work with his enormous experiences in virtual worlds. Dr. Kelly T. Morgan has been a great help for guiding me in developing various practical simulations with valuable field experimental experience.

I would like to thank my friends, Chang-Hwan, Chang-Burm and Steve Shin at UF. Their family also gave me great pleasure of life in Gainesville and help me to feel the warmth of family.

Many thanks go to my parents, my brother and his family. Finally, I take this opportunity to thank my fiancé, Kyungmi for her long-lasting love, supports and sacrifices.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>ACKNOWLEDGMENTS</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>LIST OF TABLES</td>
<td>8</td>
</tr>
<tr>
<td>LIST OF FIGURES</td>
<td>9</td>
</tr>
<tr>
<td>LIST OF ABBREVIATIONS</td>
<td>12</td>
</tr>
<tr>
<td>ABSTRACT</td>
<td>13</td>
</tr>
<tr>
<td>CHAPTER</td>
<td></td>
</tr>
<tr>
<td>1 INTRODUCTION</td>
<td>15</td>
</tr>
<tr>
<td>Statement of Problem</td>
<td>15</td>
</tr>
<tr>
<td>Limitations of VWE as an Educational Tool</td>
<td>16</td>
</tr>
<tr>
<td>Lack of Methods of VWE for Agriculture and Natural Resources</td>
<td>19</td>
</tr>
<tr>
<td>Appropriate Approach for Building Virtual Objects for VWEs</td>
<td>21</td>
</tr>
<tr>
<td>Specific Objectives</td>
<td>25</td>
</tr>
<tr>
<td>Approach</td>
<td>25</td>
</tr>
<tr>
<td>Dissertation Layout</td>
<td>28</td>
</tr>
<tr>
<td>2 VIRTUAL LEARNING FOREST: AN APPROACH FOR DEVELOPING EDUCATIONAL EXPERIMENTS AND SIMULATIONS WITHIN A VIRTUAL WORLD ENVIRONMENT</td>
<td>30</td>
</tr>
<tr>
<td>Background</td>
<td>30</td>
</tr>
<tr>
<td>Virtual World Environment (VWE)</td>
<td>31</td>
</tr>
<tr>
<td>Virtual Learning Forest (VLF) World</td>
<td>32</td>
</tr>
<tr>
<td>Main Features of VLF</td>
<td>35</td>
</tr>
<tr>
<td>Scene Components for Virtual World</td>
<td>35</td>
</tr>
<tr>
<td>Tree object</td>
<td>35</td>
</tr>
<tr>
<td>Terrain object</td>
<td>37</td>
</tr>
<tr>
<td>Forest object</td>
<td>38</td>
</tr>
<tr>
<td>Instruments for Mensuration</td>
<td>39</td>
</tr>
<tr>
<td>Sampling area tool</td>
<td>39</td>
</tr>
<tr>
<td>Logger tape</td>
<td>40</td>
</tr>
<tr>
<td>Clinometer</td>
<td>41</td>
</tr>
<tr>
<td>Diameter tape</td>
<td>41</td>
</tr>
<tr>
<td>User Interface Objects</td>
<td>42</td>
</tr>
<tr>
<td>Welcome board</td>
<td>42</td>
</tr>
<tr>
<td>Tool bag</td>
<td>43</td>
</tr>
<tr>
<td>Instruments guide</td>
<td>43</td>
</tr>
<tr>
<td>Logger note</td>
<td>43</td>
</tr>
</tbody>
</table>
### 3 VIRTUAL WORLD DESIGN APPROACH FOR ABSTRACTING AND CHARACTERIZING BASIC VIRTUAL OBJECTS

**Background**

- Complexities of Proprietary Model for VWE
  - Model Structure
  - Data Flow
  - Graphic Model Format

**Conceptualization of the VLF Educational Objects**

- Domain Interests
- Conceptualization
- Model Representation

**Abstraction of Objects into Form of Virtual Objects**

- Tree Virtual Object
- Clinometer Virtual Object
- Logger Tape Virtual Object
- Diameter Tape Virtual Object
- Sampling Area Virtual Objects: Center pole, Boundary tape, and Tree tag

**Summary**

---

### 4 AN ONTOLOGY-BASED APPROACH TO IMPROVING MODEL INTEROPERABILITY

**Background**

**Model Interoperability**

- Ontology-based Simulation Model Approach
- Exchangeable Digital Content File Format Approach

**Ontology-based Model and System**

- Ontology Design
  - Tree ontology
  - Sampling area tool
  - Diameter tape
  - Logger tape
  - Clinometer

**Ontology Implementation**

- Authoring tools
- Building an ontology

**Digital Content Generator**

- Graphic model geometry generator
- Graphic model module generator
# LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2-1</td>
<td>List of average launching speeds of VLF and computer specifications (tested at July ~ Aug, 2011)</td>
<td>55</td>
</tr>
<tr>
<td>2-2</td>
<td>List of questions and scores of evaluation form: Question Category A</td>
<td>56</td>
</tr>
<tr>
<td>2-3</td>
<td>List of questions and scores of evaluation form: Question Category B</td>
<td>57</td>
</tr>
<tr>
<td>2-4</td>
<td>List of questions and scores of evaluation form: Question Category C</td>
<td>58</td>
</tr>
<tr>
<td>2-5</td>
<td>List of actual and measured tree height and DBH at experiment sector A</td>
<td>58</td>
</tr>
<tr>
<td>2-6</td>
<td>List of actual and measured tree height and DBH at experiment sector B</td>
<td>59</td>
</tr>
<tr>
<td>2-7</td>
<td>Statistical values representing errors and measurement agreements</td>
<td>59</td>
</tr>
<tr>
<td>3-1</td>
<td>Abstraction of Tree virtual object</td>
<td>91</td>
</tr>
<tr>
<td>3-2</td>
<td>Abstraction of Clinometer virtual object</td>
<td>91</td>
</tr>
<tr>
<td>3-3</td>
<td>Abstraction of Logger Tape virtual object</td>
<td>91</td>
</tr>
<tr>
<td>3-4</td>
<td>Abstraction of Diameter Tape virtual object</td>
<td>92</td>
</tr>
<tr>
<td>3-5</td>
<td>Abstraction of Sampling Area Tool virtual object</td>
<td>92</td>
</tr>
</tbody>
</table>
## LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2-1</td>
<td>An example of conceptualization of forest learning system</td>
<td>60</td>
</tr>
<tr>
<td>2-2</td>
<td>An example of conceptualization of a forest learning system</td>
<td>61</td>
</tr>
<tr>
<td>2-3</td>
<td>Virtual Learning Forest (VLF) world Client/Server architecture showing</td>
<td>62</td>
</tr>
<tr>
<td></td>
<td>independent and extendable relationships between VWE and developed</td>
<td></td>
</tr>
<tr>
<td></td>
<td>virtual world elements</td>
<td></td>
</tr>
<tr>
<td>2-4</td>
<td>A virtual object of the sampling area tool</td>
<td>63</td>
</tr>
<tr>
<td>2-5</td>
<td>A virtual object of the logger tape</td>
<td>64</td>
</tr>
<tr>
<td>2-6</td>
<td>A virtual object of the clinometer which is used to measure the height of</td>
<td>65</td>
</tr>
<tr>
<td></td>
<td>tree</td>
<td></td>
</tr>
<tr>
<td>2-7</td>
<td>A virtual object of the diameter tape</td>
<td>66</td>
</tr>
<tr>
<td>2-8</td>
<td>The Welcome Board that is displayed at the start of each session</td>
<td>67</td>
</tr>
<tr>
<td>2-9</td>
<td>The Tool Bag of the instruments that are available for use in the VLF.</td>
<td>68</td>
</tr>
<tr>
<td>2-10</td>
<td>An user interface object of User Guide</td>
<td>69</td>
</tr>
<tr>
<td>2-11</td>
<td>The Logger Note used by a student to record tree measurements made with</td>
<td>70</td>
</tr>
<tr>
<td></td>
<td>the instruments</td>
<td></td>
</tr>
<tr>
<td>2-12</td>
<td>An example of enhancement of user interface design and interaction:</td>
<td>71</td>
</tr>
<tr>
<td></td>
<td>Diameter tool</td>
<td></td>
</tr>
<tr>
<td>2-13</td>
<td>Locations of two experiment sectors in the virtual forest</td>
<td>72</td>
</tr>
<tr>
<td>3-1</td>
<td>Structure of Classes which represent a forest virtual object following</td>
<td>93</td>
</tr>
<tr>
<td></td>
<td>a proprietary method used for OpenWonderland format</td>
<td></td>
</tr>
<tr>
<td>3-2</td>
<td>Structure of Classes which represent a HUD object for instrument (diameter)</td>
<td>93</td>
</tr>
<tr>
<td></td>
<td>following the required proprietary method used for OpenWonderland format.</td>
<td></td>
</tr>
<tr>
<td>3-3</td>
<td>A client architecture design for data-centered representation of instrument</td>
<td>94</td>
</tr>
<tr>
<td></td>
<td>objects</td>
<td></td>
</tr>
<tr>
<td>3-4</td>
<td>A client architecture design for model-centered representation of instrument</td>
<td>94</td>
</tr>
<tr>
<td></td>
<td>objects</td>
<td></td>
</tr>
<tr>
<td>3-5</td>
<td>A use case diagram of the Virtual Learning Forest</td>
<td>95</td>
</tr>
</tbody>
</table>
A sequence diagram of the virtual learning forest system, which focuses on the interaction and behaviors between actors and virtual objects. ................. 96

A collaboration diagram of the virtual learning forest system. ....................... 97

Longleaf pine trees ................................................................................. 98

A longleaf pine tree geometry model. .................................................. 99

Level of detail of the longleaf pine consisting of 4 tree digital resolutions....... 100

Major parts of the clinometer (Suunto PM-5 model). .............................. 100

Clinometer (Suunto PM5 model) digital content ....................................... 101

Geometries of 3D clinometer digital content ........................................... 102

Major parts of the logger tape and detail of the ruler scale ....................... 102

Logger tape (Stanley Powerlock) geometry model ..................................... 103

Major parts of diameter tape (JIM-GEM Pocket Diameter Tape) and drawing showing how to read a measurement ........................................ 104

Diameter tape digital content................................................................. 104

Materials and tools used for assigning a sampling area ......................... 105

Sampling area tools (center pole, boundary tape, tree identification tag) .... 106

A diagram of the longleaf pine model geometry ontology ...................... 136

A diagram of the longleaf pine model behavior ontology ....................... 137

A diagram of sampling area tool model geometry ontology ................... 138

A diagram of sampling area tool model behavior ontology .................... 139

A diagram of diameter tape model geometry ontology .......................... 140

A diagram of diameter tape model behavior ontology ........................... 141

A diagram of logger tape model geometry ontology ............................... 142

A diagram of logger tape model behavior ontology ............................... 143

Taxonomy of the virtual object (clinometer) domain ontology ................ 144

A diagram of clinometer model geometry ontology ................................ 145
A diagram of clinometer model behavior ontology .......................................................... 146

An implementation example of an instrument (clinometer) ontology within LyraBrowser: ............................................................................................................. 147

VLF Digital Content class and its subclasses ................................................................... 148

Taxonomical hierarchy of VLF DC Clinometer 3D Model class ................................. 148

Subclasses and associations of VLF Clinometer Shape class ................................. 149

Example of building a group of model behaviors for VLF instruments with the SimulationEditor, an ontology-based simulation authoring tool ............................ 150

Representation of Equation element class and their subclasses in the LyraBrowser: ............................................................................................................ 150

An equation object for representing the tree height calculation with view angle in percentage and distance, in equation tab of EquationEditor .............. 151

An example of defining a symbol (SYMBOL Tree Height) within Symbols tab in the EquationEditor: ......................................................................................... 151

An example of geometry instances in ontology representing a scale wheel part of an instrument (clinometer) within LyraBrowser .............................................. 152

An example of generated COLLADA code for a scale wheel part of clinometer model: .......................................................................................................... 153

An example diagram of Clinometer model module generated from the ontologies ....... 154

A simple example of 3D model interoperability: Longleaf pine forest model ......... 155

A simple example of 3D model interoperability: Clinometer model ....................... 156
DEVELOPMENT OF VIRTUAL WORLDS IN AGRICULTURE AND NATURAL RESOURCES FOR SIMULATION EXPERIMENTS AND ELEARNING USING ONTOLOGY-BASED SIMULATION

By

Yunchul Jung

December 2013

Chair: Howard W. Beck
Major: Agricultural and Biological Engineering

The virtual world environment (VWE) is an emerging technology that offers advantages for experiencing various problems and exploring issues associated with diverse domains in a virtual world without disadvantages of the real world environments such as lack of access, expense, and in many cases potential hazards. Although VWEs offer capabilities and potentials for building an agriculture virtual world, there are many problems related to creating agriculture digital content and sharing it in the domain. This research on development of virtual worlds in the agriculture and natural resource domain facilitates development of simulation experiments and educational tools by applying ontology-based simulation methodology, and resolves some emerging problems on development of virtual world environments.

To achieve this goal, a case study was developed in which basic virtual objects in the agriculture and natural resource domain are characterized, and development methods are established for implementing digital content. A digital content ontology is constructed for storing the elements of digital content irrespective of graphic model formats. And, an ontology-based simulation methodology is developed for the VWE to
realize behaviors of objects in real world. As a experimental example, a virtual world for e-learning in forestry called the Virtual Learning Forest (VLF) is created, and various instruments used for field laboratory-based learning activities are developed. Generation engines for graphic model and digital content are constructed and utilized to create digital products automatically by cooperating with a model geometry ontology and a model behavior ontology.

An evaluation of system usability was conducted for 30 students in the VLF, and students gave slightly higher score than middle score (4 points in 7 scales) for the design of instruments. For the learning efficiency, test scores shows that students developed their skill by increasing accuracy of measurement. The result of the evaluation showed that the VLF system can enhance student experience of system usability, accessibility to digital content and expandability of virtual worlds in the domain. A model interoperability test was conducted by porting the content of the VLF into two different platforms: OpenSimulator (http://opensimulator.org) and OpenWonderland (http://openwonderland.org). It is verified that the applied ontology-based VWE methodology improves capability of sharing graphic models and object behavior and shows the possibility of building enhanced and expanded digital content. It will serve to facilitate sharing agriculture digital assets by improving model portability and interoperability between VWEs, and contribute to decreased cost and complexity of building virtual worlds for simulation experiments and education.
CHAPTER 1
INTRODUCTION

The virtual world environment (VWE) is an emerging technology that offers advantages for experiencing various problems and exploring issues associated with diverse domains in a virtual world without disadvantages of the real world environments such as lack of access, expense, and potential hazards. Although VWEs offer capabilities and potentials for building an agriculture virtual world, there are many problems related to creating agriculture digital content and sharing it in the domain. This research on development of virtual worlds in the agriculture and natural resource domain facilitates development of simulation experiments and educational tools by applying ontology-based simulation methodology, and resolves some emerging problems on development of virtual world environments. The research involves four main tasks:

- Characterizing basic virtual objects and development method
- Developing ontology-based simulation methodology for VWE
- Implementing virtual worlds by case studies
- Evaluating ontology-based simulation methodology for VWE

Statement of Problem

The virtual world environment (VWE) is an emerging technology that provides Internet-based social networking services to diverse virtual communities within realistic 3D virtual worlds of buildings, facilities and environmental areas such as forests (Kelton, 2008; Bishop, 2009; Beck, 2009). It has attracted much attention because of its capability of reducing restrictions of space and time and the possibility for building unlimited virtual world with virtual objects. There are many successful applications of virtual world technology in the areas of medicine (Ann Myers Medical Center, 2010;
Rogers, 2009; Virtual Neurological Education Centre, 2007), business (Honey et al., 2009), and education (Campbell, 2009; Oloruntegbe and Alam, 2010). But, it is just beginning to gain interest in the agriculture and natural resource domain (Prada et al., 2010; NOAA, 2007; Schott, 2009; Beck, 2009; Jung et al., 2010). Various challenges exist for development of virtual education and simulation environment, and these include:

- Limitations of VWE as an educational tool
- Lack of methods of VWE for agriculture and natural resources
- Need for a more efficient and cost effective approach for building virtual objects for VWEs

**Limitations of VWE as an Educational Tool**

Virtual world is an emerging technology used by an increasing number of educational institutions around the world and is considered a technology, environment, medium, learning and teaching tool. The technology has been reported by over 100 institutes for education constructed and used a virtual campus in Second Life (http://www.secondlife.com) (Hudson, 2007). For educational purpose, virtual world extends the classroom into a virtual space to enhance student’s experience with augmented materials and to accelerate the learning processes through direct emersion within a problem space. This contrasts with classic textbooks or multimedia web-based e-learning programs that teach about the subject but do not provide direct experience within the domain of study. Educational applications in a virtual world enable animation, simulations and role playing, distance learning, collaborative experiences, and experiential projects. The virtual world concepts are used in various domains, including architecture and building, art and design, business, cultural studies, health, tourism,
social science, emergency services, and scientific simulations (Swanson and Kane, 2007).

There are research studies reporting the experience of student learning and teaching by using virtual worlds. Hudson (2007) taught a journalism class for a group of 20 students in a virtual world. Various discussions which were conducted in the virtual worlds resulted in enhancing student’s participation in class and encouraging more discussion. It was determined that the virtual world engages the students in learning from the start and encourages collaboration and cooperative communication between students. Similarly, Lemmon and Kelly (2009) utilized the virtual world for training students in interviewing skills, and it was suggested that interview skills can be enhanced through virtual training in role playing scenarios. In those experiments, although a virtual class room in a virtual world contributed to learning, communication and collaboration among students, simple mimicking the teaching procedures in the virtual world was insufficient for student to utilize rich virtual objects in their learning.

Furthermore, Senges and Alier (2009) determined that virtual worlds supported opportunities for education purposes, but it would be a challenge building the informal and holistic learning scenario for most education institutions. Gregory and Tynan (2009) provided survey results of 11 students on learning experiences in the virtual world, and indicated that students felt that their learning in the virtual world was informative, engaging and educative when the virtual world provided concrete learning scenarios.

Simultaneously, in most cases of virtual experiments, various virtual objects are required and implemented for educational purposes. Honey et al. (2009) studied learning experiences in simulating patient care scenarios, and it was reported that a
virtual world helped students learn skills to care for patients. But, it required intensive work to building the virtual clinic and props. Similar results are reported by various researchers in education including a constructivist learning environment (Gül et al., 2008), a case study of database administration laboratory work (Cranitch and Rees, 2009), the evaluation of an emergency medical treatment computer game to support triage training for decision making that prioritizes mass casualties in terms of treatment (Jarvis and Freitas, 2009), science teaching and learning (Oloruntegbe and Alam, 2010), and the experiential learning study (Wood, 2009). Virtual objects for these experiments are critical resources of a virtual world that consume expensive work. There is a problem with sharing the virtual objects in other experiments.

There is research reporting a bright but cautious future for learning in the virtual world. A survey result from educators in Second Life (Swanson and Kane, 2007) indicates that the virtual world offers great potential for the future Web, but it will take more time to become a mainstream of learning because users feel uneasy about virtual objects. Wood (2009) suggested that a method of learning by virtual world would need a more cautious approach, because he claimed that the technical problems of the virtual world contributed to student dissatisfaction with the learning experience. Campbell (2009) utilized a virtual world for learning simulations by role playing to improve professional decision making among pre-graduate teachers taking a unit in professional ethics. It is reported that the virtual world enhanced interest and experience of students in learning. But, he claimed that the effectiveness of the virtual world is limited by lack of reality of virtual objects, efficiency of collaboration tools, and information for usage of the complex system.
Freitas et al. (2009) proposed an evaluation methodology for supporting evaluation of specified learning activities in a virtual world instead of using a general survey questionnaire. By applying it, he concluded that students were positive about using the tools for supporting collaboration based on shared interest. And, it is claimed that appropriate technical support and well-structured sessions in virtual world are essential for providing enriched experiences from the virtual world.

**Lack of Methods of VWE for Agriculture and Natural Resources**

While virtual world technology has been used for learning and teaching in various domains, in the domains of agriculture and natural resources it just beginning to gain interest. In the early applications of virtual reality technique which focused on 3D simulated virtual environment rather than social networking of virtual world, Orland et al. (2001) proposed communication capability for landscape planning decision support systems. Geographic information system (GIS) tools are utilized to represent geographic data such as soil profiles in 3D, and the system facilitated hypermedia and Internet-based communication. However, it is a closed system in which only prebuilt virtual objects are utilized, and provides a limited access to the system for highly trained users, and collaboration and communication between users in the virtual environment are restricted.

With recent advances of Web technology like the new paradigm of Web 2.0 utilizing dynamic web contents such as blog, wikis, video sharing to interact and collaborate with users (Anderson, 2007; Boulos and Wheeler, 2007), in the agriculture and natural resource domain web applications supporting virtual communities are increasingly used for educational purposes. An agriculture extension program at Ohio State University utilizes a blog site for gathering agriculture news with a real-time
repository of information by two way communication (Kleinschmidt, 2009). The FAO (Food and Agriculture Organization) site services various audios for international and regional communities through national broadcasters. And, Bemis-Schurtz and Bussmann (2009) proposed an e-learning site using Web2.0 applications for learning about ecosystems. The ecosystem is a real-time conversation-based system, and provides desktop widgets and web widgets to find an interesting community and to suggest an idea. And, Twitter and mobile communication is utilized to share the ideas and improve communications.

Virtual world technology facilitates a Web 2.0 service hosting other applications and services by using visualization. In the 3D web, it provides more intuitive ways to navigate multimedia collections, and to browse information spaces and document collections in virtual libraries. The National Oceanic and Atmospheric Administration (NOAA) launched a virtual world that facilitated education about the Earth’s environment in many ways (NOAA, 2007). In the virtual world users can stand on a plane while going through a hurricane or experience the ice caps melting by standing on them. Virtual islands are utilized to educate people about the Earth's oceanic and atmospheric features.

In agriculture and natural resources education, several virtual worlds are developed as games (Farmville, 2009; SimFarm, 1993) within which students follow rules and steps that are predefined to achieve educational knowledge. Prada et al. (2010) proposed a game implemented in the on-line virtual world platform, the OpenSimulator, for learning the potential effects of agriculture in the environment such as the impact of fertilization in water quality and production of food. They assessed the quality of the
game and user’s experience by using a questionnaire on challenge, difficulty, control, cleanness, improvement knowledge, and awareness of the impact of agriculture in the environment. It showed good acceptance by the users and positive result on enhancement of awareness of impact in agriculture. Ecosystems and forests in virtual worlds are usually represented as a landscape. In virtual world environment like Second Life there are various examples (http://secondlife.com/destinations/nature), which provide diverse forms of nature and environments. They are useful for experiencing virtual nature and enjoying artificial scenes, but the reality of features is limited by graphic resources and capability of the VWE. Lack of educational tools and entities restrict use of them in teaching and learning.

Beck (2009) suggested prototypes of virtual laboratories and entities by using the Lyra VWE for agriculture education where students can interact with 3D, simulation-driven learning experiences modeled after real laboratories and entities. Examples are provided of a virtual greenhouse, soil profile and trees proposing how to design and use them in the domain of agriculture and natural resources.

**Appropriate Approach for Building Virtual Objects for VWEs**

Within the traditional approaches of developing virtual worlds by manual programming and hand constructing art works, constructing virtual worlds requires much effort to create structured data of graphic models and implement complicated behaviors, relationships and processes among objects.

VWEs are developed with different programming languages and different graphic engines, and supporting graphic model formats are various. For example, OpenWondeland (2010, www.openwonderland.org) VWE developed in Java supports model modules complying with jMonkey (http://jmonkeyengine.org), a graphic engine,
API, and graphic model formats in X3D (http://web3d.org/x3d) and COLLADA (http://collada.org) are allowed in the VWE. Meanwhile, OpenSimulator VWE is developed with C++, and supports models created with graphic primitives oriented from Second Life VWE server.

A model module is a program set indicating the associated graphic model object and manipulation processes for the model. Although it is a useful and powerful structure to integrate graphic models and behavior, it is limited in sharing with other VWEs adopting different system formats.

The complicated information in a module targeted for proprietary VWEs can be constructed from data structures containing abstracted information by using an ontology. Ontology is a collection of concepts and relationships among these concepts in a specific domain (Noy et al., 2000), and a promising technology for knowledge reuse and knowledge sharing (Beck, 2003). There are some efforts to apply the ontology-based approach to the VWE. Anthopoulos et al. (2010) proposed a virtual learning environment for training in profiling, negotiation and crisis management. The environment utilized a rich knowledge base of professional experiences on negotiation methods, crisis management, decision making and legal affairs. An ontology was adapted to express terminology and instance information related with problem solving procedures and collaboration. Meanwhile, Ahmed and Gracanin (2010) addressed the interaction interoperability issues of 2D input devices for virtual world, and proposed a framework adopting ontologies of input devices and tasks for abstracting the input and output of various devices and interaction techniques to various tasks.
Ontologies are utilized for enhancing interoperability of data and model. Suman et al. (2010) designed and developed a virtual dental school by using open source virtual world technology (OpenWonderland). It is implemented in the distributed client architecture for maintaining applications of the virtual world flexibly. And, the application architecture was improved by representing configurations of virtual objects in XML explicitly that includes the information of model behavior, storage location, orientation and scale. Jeschke et al. (2009) proposed a software infrastructure by utilizing the OpenWonderland (2010) VWE for enabling the interlinking and integration of experimental superstructures and simulations of nanotechnology in natural science and engineering. A subject ontology is developed to distinguish experiments from simulations by describing the input and output data of a simulation and evaluation algorithms explicitly. Sanchez-Ruiz et al. (2008) assessed the data interoperability problem in the virtual worlds for high resolution weather data from various agencies, and suggested to use ontologies to manipulate the semantics of visual representations and operations associated with the data. It facilitated recording data operation by users through their avatars in the virtual world, and supported transforming them into various resources such as multi-language source code, visualizations, and web documents by external plug-ins. However, these efforts contributed to utilize the ontology for representing internal system knowledge, not to expand the approach to interoperate the model within other VWEs.

For generating and manipulating graphic objects in virtual world automatically, Beck (2009) introduced an approach of ontology-based data management and provided prototypes of greenhouse, soil profile and trees. Applying an ontology-driven approach
to build a virtual world can provide ways to describe the properties and structures of an object explicitly with reduced efforts, and it can enable to create them automatically. For example, once the prebuilt virtual objects such as soil and plant are stored into the ontology, they can be generated automatically by using the virtual object ontology instead of creating another virtual world and objects manually.

For simulated models in virtual worlds, there is a need for supporting communication between virtual objects by using embedded models or processes during simulation or interaction in the VWE. Ontology-based approaches to developing dynamic models, which explicitly represent knowledge of concepts in system and relations between them, have shown successful results in building simulation systems generated automatically from a structured model ontology, and authoring tools are used to build and maintain dynamic models without programming (Beck et al, 2008; Kwon et al, 2010a).

The ontology-based simulation approach consists of a set of procedures: 1) creating an object relationship diagram, 2) defining concept with symbol, definition, and related information, 3) developing a model or relationship between concepts in the form of mathematical representation, and 4) automatically generating code for running the simulation. In a similar way, it is possible to develop an ontology-based simulation for a virtual object in a virtual world.

For eLearning systems such as a forest learning system, interactions between instruments and nature-driven objects can be described. For example, a logger tape object used to measure distance from instrument to target object can have a process providing a distance value by interacting with object. The process can be represented
as a mathematical expression of vectors of the object's positions. In addition, object
behavioral processes such as showing changed tape line when the instrument is moved
will be useful for describing interactions within and between objects. It is possible by
abstracting virtual objects according to the role of model, view and control.

Specific Objectives

The objectives of the virtual world simulations developed in this dissertation was
constructed to address the following objectives.

- Implementing virtual worlds by case studies on creating virtual forest world with
  the generated models and the simulated forest population.
- Evaluating the VWE by conducting a survey on the Virtual Learning Forest and
  measuring educational achievement
- Characterizing basic virtual objects which belong to the agriculture and natural
  resource domain for educational experiences and simulation, and developing
  procedures from designing to implementing objects.
- Developing an ontology-based simulation methodology for VWE by implementing
  information structure of simulation and experiments with an ontology and using
  them to automatically generate digital models and build virtual objects across
  different VWE platforms.

Approach

A Virtual world environment (VWE) is a genre of online community taking the
form of a computer-based environment where users can interact with others to use and
create virtual 3D objects (Bishop, 2009). Use of the interactive 3D virtual environment
and avatar-based social networking has increased the number of people using virtual
worlds for diverse purposes including education (Bray and Konsynski, 2007). In
particular, it enables creation of a virtual world which is able to deliver educational
experiences around the Internet to different users from different countries to interact,
provide, gather and evaluate information.
The VLF for the forestry laboratory exercise in the virtual world were initially built based on the OpenWonderland platform. The VLF is a composite module system composed of the longleaf pine module and terrain model. The level of detail of the tree module is varied by switching different detailed models according to the distance from the avatar, and resulted in significant increasing the performance of the virtual world system. In the near distance a fully developed 3D model is used to enhance the reality of the world. For the mensuration exercise, three field instruments, logger tape, diameter tape and clinometer, are developed as HUD components which help students to feel like they are using real instrument in the virtual world.

A prototype VLF is developed in the traditional way of implementation by manual coding following the proprietary model protocol of the VWE, which gives a solid, concrete virtual world. Three types of virtual objects are developed, which are 1) terrain object, 2) forest object and 3) instrument object. A virtual object of terrain is an object focusing on a graphic model and its geological information, while a forest object is a combined model of graphic model of a tree and dynamic model simulating population of trees by generating geological and physical information for each tree based on a single prototype tree model. Virtual objects of instruments are objects which stress their behaviors including interactions with the controller (avatar) and communications with target objects as well as graphic model and interface for representing data.

With the traditional methods, virtual objects are expressed in a proprietary format oriented in the VWE, including information about relationships between graphic models, behaviors and processes. But, this results in limited accessibility to inflexible structures and information when it is required to create a new virtual world and to modify
processes in a virtual object. Behaviors of virtual objects are implemented by using proprietary languages required by the specific VWE, so it is difficult to understand and share them efficiently. Processes (logic) of calculations within virtual objects such as instruments can be represented in an independent form as mathematical expressions.

To solve some of the problems that come from the traditional development methods, the approach presented here improves the flexibility of the model building process by introducing ontology-based data management technique. Model interoperability is enhanced in two ways; 1) ontology-based simulation model development, and 2) ontology-based exchangeable digital content. The ontology-based data management system is developed to organize information on domain-specific educational systems such as forest field size, tree distribution and population information. It cooperates with a simulation model generation engine to build virtual worlds. The ontology-based exchangeable digital content system is developed to organize information related with digital content such as shape, material and position by excluding proprietary software and data formats. The digital content ontology is created to define digital elements. It contains concepts for extended model feature such as model behavior and related processes that describes model interaction with the external environment.

The VLF system has been evaluated by different user group through testing different aspects of the system during all system development stages. Evaluation tests consist of analytical evaluation by an expert group and usability evaluation by a student group. At an early stage of development, a prototype model was evaluated by experts familiar with forestry field experiments and teaching. This evaluation was used to
resolve questions about the basic approach and to improve 3D object design and interaction design. During the analytical evaluation, virtual object format and design were discussed and selected from diverse possible graphic formats. Designs of interaction and user interface were issues discussed over the entire development stage, and they have been improved by modifying systems based on results of discussions. With the VFL system, a field test for achieving qualitative evaluation of the system performance has been conducted with 20 computers having the same system specification and running under same network environment. The test focused on estimating system performance and system speed and providing system specification guideline. Usability testing for system evaluation and evaluation of lessons learned are conducted with students taking a related class. The questions of system speed, level of intuitiveness and usefulness are included on the evaluation items, and it helps to get quantitative scores evaluating the value of the system for students.

**Dissertation Layout**

Chapter 2 describes the development of the Virtual Learning Forest (Objective 1) and illustrates virtual objects developed for experiments and main features of the system. Evaluation studies (Objective 2) are presented to explain the usability of the VLF system developed by the proposed method.

Chapter 3 describes approaches for abstracting and characterizing virtual objects to identify the role of model elements (Objective 3). Chapter 3 also explores several problems with traditional development methods for virtual objects.

Chapter 4 describes an ontology-based approach to improve model portability and interoperability (Objective 4), including examples of defining features and function,
creating digital content, developing a model ontology, and generating a virtual object automatically from the content in the ontology.

Chapter 5 summarizes conclusions and contributions, and identifies future directions.
CHAPTER 2
VIRTUAL LEARNING FOREST: AN APPROACH FOR DEVELOPING EDUCATIONAL EXPERIMENTS AND SIMULATIONS WITHIN A VIRTUAL WORLD ENVIRONMENT

Background

Virtual World Environment (VWE) as an eLearning tool attracts much interest with its capability for online-based communication and collaboration and possibility for mimicking a real world with virtual objects (Callaghan et al., 2008; Hodge and Collins, 2010). However, the high-level technical knowledge required and limited resources available for constructing virtual objects becomes a challenge for utilizing the technology widely.

The Virtual Learning Forest (VLF) project is an effort to adopt the VWE for forestry education in order to improve traditional computer-based training. It was designed to provide various virtual laboratory experiences to student without limitations of space and accessibility, and provide opportunities for improving student field skills.

In the VLF, students create an avatar that is a virtual person object representing themselves in the virtual world. Then, students take measurements in order to estimate the tree height and tree diameter of the forest. Students begin by identifying a sample plot, a small region within which all the trees are measured. Next students utilize virtual tools to measure tree diameter. To measure the tree height with virtual clinometer, students measure a distance from the tree at first, and then use a clinometer. The measurements are recorded in the logger note.

The VLF for the forestry laboratory exercise was developed within the OpenWonderland platform. It is a composite module system consisting of environment modules and field instrument based on graphic models and Java program code. Forest and terrain modules are two environment modules, and a forest of 400 longleaf pines is
created that occupies a 2 acre terrain. Field equipment is used to measure tree height and diameter, and includes a logger tape, diameter tape, and a clinometer that mimic shapes and behavior of real instruments. A sampling area tool is used to identify a region of the forest in which all the trees are measured and includes a center pole, boundary tape and tree tags. User interface objects including a logger note and tool bag facilitates recording data from measurements and provides help to students for guiding their activities.

**Virtual World Environment (VWE)**

VWE is a computer-based environment where users can interact with others, and use virtual 3D objects (Bishop, 2009). Use of the interactive 3D virtual environment and avatar-based social networking has increased the number of people using virtual worlds for diverse purposes including education (Bray and Konsynski, 2007). In particular, it enables creation of a virtual world which is able to deliver educational experiences around the Internet to different users from different countries to interact, provide, gather and evaluate information (Ramasundaram et al. 2005).

There are various VWEs including commercial packages and open source environments. Second Life (Linden Lab, 2003) is a popular and mature environment which has been utilized for virtual campuses of many universities including the University of Florida, Harvard University, and Stanford University. However, it requires student to subscribe to use the service, and provides only an open network based on virtual “islands” (regions within the entire OpenWonderland landscape). Like other commercial packages such as OLIVE (2010), the limitations of cost and accessibility for building virtual worlds for experimental and educational purposes encourage using open source technologies instead such as OpenSimulator (OpenSimulator, 2007), Croquet
(Croquet, 2007) and OpenWonderland. OpenSimulator is a platform (Microsoft OS) dependent system, so compatibility with other operating systems can be restricted. Croquet is an open source project based the programming language Smalltalk (Smalltalk, 1972).

OpenWonderland is an open source tool and environment written in Java for building and running virtual worlds. Thus, it has better cross hardware platform independency compared with OpenSimulator, and offers better opportunities to utilize the wider range of resources available in Java. Asset management architecture in OpenWonderland supports extendibility. Graphic models can be developed with common external graphic tools like Blender (Blender, 1995) and imported into OpenWonderland utilizing formats such as X3D (2004) and COLLADA (2004) which are standard formats for sharing graphic models through the World Wide Web (WWW). Behaviors of graphic models also can be implemented by using Java Script internally.

**Virtual Learning Forest (VLF) World**

For forestry education, the VWE shows much potential for offering immersive learning experiences that improve on traditional computer-based training. It can offer laboratory experiences that complement real laboratories which may be expensive, dangerous, or inaccessible to students. The Virtual Learning Forest (VLF) project (Bannister, 2009) attempted to create a new forestry educational delivery paradigm by adopting the VWE technology. It was hoped to overcome limitations of cost, time, distance and space. The project aimed to achieve the following objectives (Bannister, 2009): (1) "strengthen and expand the skills students learn in their existing field exercises", (2) "offering a variety of virtual laboratory experiences to a large number of students across the US and internationally so that they can build experience in forest
types they would not have the opportunity to physically visit", (3) "providing students opportunities to interact with their peers at other institutions via collaborative exercises in each other’s corresponding virtual forests and ecosystems, thereby instilling in them, early in their training, the importance and advantages of developing relationships with distant colleagues in natural resource management", and (4) "offering these opportunities on line, available to students “24/7”, without the necessity for travel, thereby expanding learning opportunities for a large number of students at little cost".

During the beginning stage of system development, there were discussions about the effects of educational environment changes on the new paradigm as well as technical issues. And, they contributed to expand considerations for roles of the system in the educational environment and for the required features of system.

An example describing a conception of roles of the VLF and its possible relationships with educational environment is given at Fig. 2-1. Changes in educational policy or situation stimulate the development of the Virtual Learning Forest system in the school domain. The system consists of a server, virtual forest libraries, model ontologies related with instrument, tree growth or forest generation, and GIS data for terrain and forest. The school domain includes teacher, student, and system manager and researcher, and distance education students have close relationship with the school. Other schools and international institutes utilize the VLF for education, and cooperate to enhance the learning program and to improve assets of the virtual environment.

The school or system manager responds to the requirements of education methodologies according to the changes of educational situation, policy, and technical improvement. And, requirements from community affect to decide the direction and
content of education in school. Teacher provides educational material to the learning system and set plans for field experiments. Students use the learning system for learning and developing field skill through the virtual environment. With responses from student and feedback from other users, researchers evaluate the virtual learning system and learning contents. Results of the evaluation are used for improving learning system and enhancing the learning environment. As future contributions of the VLF, students graduated from school seek chances to use knowledge and field skills at work places by being employed. Distance students who work at the job and learning field skills through virtual learning system contribute to communities by using learned knowledge.

An example of a conceptual model diagram is shown in Fig. 2-2 concentrating on the functionalities of the VFL. A big circle represents a system boundary of the VLF, while small circles in the big circle indicate functions of the VLF. There are resources of information and data at the left side of system boundary. The forest generation process consists of terrain creation and tree creation. These processes utilize topological and environment data from the GIS. Data created for the forest and obtained from the instruments are associated with data storage. A prototype virtual world consists of a forest ecosystem and virtual educational instruments for measuring and recording information about trees in the forest to support real field experiment activity in classes. Objects in the domain are categorized into natural resources (tree and terrain) and data gathering instruments (clinometer, logger tape). A tree is an essential virtual object for representing a rich ideal forest, and a forest can be enhanced by combining appropriate generation models. A terrain model can be enhanced by implementing and adopting a generation interface using a land elevation model based on GIS data.
Figure 2-3 illustrates the prototype VLF client/server architecture. Server side components of OpenWonderland consist of a game server (DarkStar) on the web server container and a voice communication server (jVoice Bridge). Client components consist of system layers (game server protocol, jMonkey - 3D graphic engine, and VWE client) and virtual world layers. System layers enable communication with server components, to render virtual objects, and to maintain system assets in the local (student) computer. Virtual world layers implement the VLF world object and related applications which are designed to be independent to the world itself. Application data are provided through server/client communication from the server.

**Main Features of VLF**

The VLF is a composite module system including scene components, instrument components, and user interface components. Scene components are used to build the virtual forest, which includes the longleaf pine tree module, the terrain model and the forest module by combining the two components in the scene. Instrument components including a logger tape, a clinometers and a diameter tape, are developed as a head-up display (HUD) component. User interface components are 2D virtual objects formed in HUD object without associating with a 3D graphic model, and include the Welcome Board, Tool Bag, User Guide, and Logger Note object.

**Scene Components for Virtual World**

The forest world consists of three scene components that are structured objects in the scene; 1) tree object, 2) terrain object and 3) forest object.

**Tree object**

Detail level and style of tree model affect the performance of virtual world. Highly detailed (high resolution) tree models improve the reality of scene in the virtual world
(Palubicki et al., 2009), but require considerably complicated features and delicate texture resources which can overburden graphics hardware resource used for rendering the required detail. Thus, it is critical to choose a balance between reality and performance for the target virtual world.

A longleaf pine tree model from the SpeedTree® (SpeedTree, 2002), a fully developed commercial 3D tree library, was modified and adapted by reallocating shapes and adjusting resolution of textures to adjust the quality level of the model. As a leaf of the longleaf pine tree is a needle shape, it is usually featured as a group of leaves rather than each individual leaf shape to enhance the graphic performance but keep a high level of visual quality.

For better system performance, it is helpful to use various shaped and styles of a 3D model, which can range in level of detail from a low resolution billboard 2D plane to a fully developed high resolution 3D model in which every branch and leaf is represented by a vector object. The level of detail (LOD) technique, which dynamically manages the appearance of detailed features, is applied to the tree model. One form of LOD is the continuous LOD (Lindstrom et al., 1996) which controls the number of vertices used to describe shapes according to the distance from the avatar’s viewpoint. When it is possible to use the technique, it is created automatically in the gaming engine used to render the VWE.

The discrete LOD technique switches different models at various levels of detail for the same object and can significantly increase graphic performance, even more than the continuous LOD. It is applied to the longleaf pine tree module with four different style models, which include three 2D plane tree models and one partial 3D vector
model. The 2D models use an image of a longleaf pine as a texture and are created by intersecting image planes having different angles of orientation. They are set to appear at the range of 0m, 30m, 60m and 90m. In range above 90m a one plane bitmap tree model is used. In the near range within 0-30m from the avatar position, the highest resolution 3D model (617 edges, 83 faces, 15 materials) appears with a cone-shaped trunk textured with fully detailed bark.

**Terrain object**

A terrain model is built in a plane object representing a 2 acre area. The model represents only the surface of terrain, and is mapped with a simple grass texture. The 2 acre area of the terrain is chosen as a small but realistic forest size, while considering the system performance of the virtual world containing 400 trees mapped onto the terrain.

To improve the user experience, the topographical characteristics of longleaf pine habitats are considered for build the terrain model, and the topology involves area, elevation and steepness. This information was collected based on the typical longleaf pine habitat. Most longleaf pine habitats occur around the southeastern United States, and much of the state of Florida state consists of longleaf pine habitat (Peet and Allard, 1993).

The elevation data set for Florida state is created by using the digital elevation model (DEM) of Florida that is published with 30m resolution by the U.S. Geological Survey (USGS DEM, 1999). Using a geographical information system (GIS), the DEM dataset is overlapped with a Florida county boundary dataset (USCB, 1990) and a Florida forest inventory dataset created by the USDA Forest Service (USDA, 2009), which contains more than 20 data elements including tree sampling results such as
height and volume of a selected tree. By processing raster analysis using GIS, mean and variation of elevation in the longleaf pine habitats are obtained.

Based on those values, some referencing vertices are generated by using a normal distribution, and some manual changes are added to fix odd elevations. They are added to the elevation data set of the terrain plane model, and it is refined into a terrain model using groups of triangulated planes.

**Forest object**

A forest module constructs the virtual forest world by compositing the longleaf pine modules and the terrain model in the scene of the virtual world. It generates and loads 400 tree models and a 2 acre terrain model, and applies scale factor and transformation factors to 400 trees. Tree distribution and scale information are created and utilized within the module. All 400 trees are created from the same original prototype tree model, but each tree is scaled to a particular height and diameter.

A tree distribution is created with random numbers. Two random numbers sets for 400 trees are generated by the Java random number generator ([http://docs.oracle.com/javase/7/docs/api/java/util/Random.html](http://docs.oracle.com/javase/7/docs/api/java/util/Random.html)), and the real numbers between 0 and 1 represent coordinates of trees in the 1M x 1M scale terrain. Actual coordinate values are defined by multiplying actual scale of target terrain (90M x 90M).

Average tree age in an area can be obtained by GIS data (Florida forest inventory data from USDA Forest Service) analysis with longleaf pine habitats data. Sizes by tree age and scale variation of tree are generated by applying a normal distribution to the standard age-height relation (Jack et al., 2005) and the height-diameter relation (Platt et al., 1988).
Instruments for Mensuration

Mensuration involves measuring physical parameters (e.g. height, diameter) of trees in a forest. During the field experiment for tree measuring, mensuration is conducted with various instruments in accordance with the measuring procedures. The logger tape, clinometer and diameter tape are common instruments for mensuration which are utilized for determining the quality and dimensions of trees. The sampling area tool is a combined instrument consisting of sampling boundary tape, center pole, and tree tag.

Most instrument objects consist of a graphic model of the instrument and student interface HUD object displaying enlarged measurement scale to make it possible for students to read the instrument scales. The graphic model of an instrument is presented in the virtual world, so other students can see the appearance of model. Each student owns a set of virtual instruments. HUD objects are only shown at a student’s individual screen (the individual student’s HUD is not visible to the other students).

A Head-up display technique is utilized in the VWE to develop the instruments, and shows a transparent display that presents data without requiring the student to look away from the normal viewpoint. Students can view information by looking head-up and forward instead of having to angle down to look at lower instruments because information is displayed in the direction in which the avatar is looking.

Sampling area tool

A sampling area tool (Fig. 2-4) is utilized to represent a current experiment area where students undertake measurement. All trees within the sampling are to be measured by the student. Measurement activities with the instrument are conducted within a small circular plot area having around 40 ft radius over the whole experiment
area. The radius can be changed by students if necessary. The sample area tool consists of a center pole, tree tags, and area boundary tape.

A center pole is a graphic model representing a center of the sampling area. The location of the center pole is set by the student by clicking a certain ground point. The location of this point can be picked arbitrarily by the student. The sampling area boundary is marked by a boundary tape that appears on the ground at 40 ft away from the center pole. The tape is composed of line segments that form into a circle. A tree tag is a graphic object containing a unique identification number for each tree, and is placed besides each tree within the sampling area. Students use the Tree tag ID number to record measurement data for each tree.

Logger tape

In the real world, a logger tape is usually a 100 foot long metallic tape that is spring-loaded for retraction into a metal box (much like a home measuring tape), which has distances marked in feet. The end of the tape is often fitted with a nail, so that it may be fastened into the bark of a tree, enabling a single person to measure the distance from a tree to a point on the ground.

The logger tape is implemented in the VWE with a tape graphic model and a HUD object (Fig. 2-5). The tape graphic model appears at the scene when a student activates the tool. Tape line from tape body to a target object is represented with a temporary line object, and is appears as a straight line (sagging effects were not incorporated). An user interface for reading a measurement from the tape is a HUD object that shows the enlarged view of the instrument including a part of tape body and tape line indicating a measured value. By using an image of the instrument, it helps students feel like they are using a real instrument in the virtual world. It shows the
distance measurement in feet and meter scales on left and right side at the head of the tape as the tape is moved forward and backward from the target tree.

**Clinometer**

In forestry, a clinometer is used as an angle measuring device used to determine the height of a tree. It uses simple trigonometric relationships to determine tree height given the distance the student is standing from the base of the tree (this is determined by the logger tape) and the angle from the ground to the top of the tree. This angle is determined by viewing the top of the tree through a viewfinder available in the clinometer. The clinometer typically has two scales: a percent scale for the tangent value of the angle (on the left) and a 1:66 scale (on the right). The 1:66 scales gives the direct height reading at location 66 feet away from the tree.

The clinometer is represented in the virtual world with a clinometer graphic model and a user interface for reading measurements (Fig. 2-6). The graphic model appears in front of the avatar when the instrument is activated. The user interface is formed as a HUD object displaying a dynamic measurement of both scales. Scale rulers in the object are represented with line graphic elements, and changes of view angle are reflected instantly by updating location of lines and referenced values of the ruler.

**Diameter tape**

A diameter tape is a tape measure used to determine the circumference of the trunk of a tree. A diameter tape has a smaller and more compact shape compared to the logger tape, and gives measurements in inches and centimeters when wrapped around the tree. By conventional standard, measurement of the tree diameter occurs at “breast height”, and it is specifically at 4.5 ft above the ground.
In the virtual world, the diameter tape (Fig. 2-7) is created with a tape graphic model and a user interface HUD object. The tape model appears in front of a tree where student clicks on the tree at the point where the measurement is to be taken. When clicking onto the tree, student can refer the height of the point represented at a tape ruler in the HUD object.

The HUD object contains two rulers for diameter measurement and height measurement. The height measurement is represented in a vertical dynamic ruler. The diameter ruler is mimicked by displaying a real image representing a tape wrapped around a tree, and the tape ruler and numbers are updated as the measurement changes. The reading value is calculated internally by accumulating distances between neighboring vertices on an imaginary mesh plane which intersects at the trunk of tree.

**User Interface Objects**

Several user interface objects were developed to provide support to students while conducting experiments. These include the Welcome Board showing objectives for learning, Instruments Guide to explain the tool usage, a Tool bag for selecting instruments, and a Logger Note tool to record measurement data. They are implemented in HUD objects, and no 3D graphic model is associated with them.

**Welcome board**

A welcome board object (Fig. 2-8) is a user interface showing information for users when they first access the VLF. It is implemented as a HUD object which contains the owner's symbol image, title of the virtual world (VLF), and version of the VLF, and objectives which the students are to undertake during an experiment. It appears as a dialog in the middle of the OpenWonderland application window at the begin of each session.
Tool bag

A tool bag object (Fig. 2-9) is a HUD object which appears at the lower-right corner of the application window. It is an image panel showing a bag of tools which are the instruments which can be utilized in the VLF. It contains the logger tape, diameter tape, clinometer, GPS (which matches with the locator object providing current location of the user in the virtual world), and logger note. An instrument is activated/deactivated by mouse click actions, and a snapshot image is shown when the mouse pointer is placed over an instrument image. The snapshot helps the user to remember how the instrument is utilized in the VLF.

Instruments guide

An instrument guide object (Fig. 2-10) is a sub-tool of the tool bag object, and it appears as an image panel describing step-by-step procedure for use of the selected instrument. It consists of snapshots of all phases of the tool use, and a student can learns details about controls for using the instrument.

Logger note

A logger note object (Fig. 2-11) is a virtual object which records student's data entry. It is developed as a HUD object containing data fields and data table. It is utilized for keeping the recorded measurements during the experiments. And, the data can be sent to the remote database or saved in a CSV (Comma Separated Values) file.

It is designed to let students record their tree measurements for all trees in a particular sampling area. To track the sampling area location and trees in the boundary, plot number and plot center coordinate in the virtual world can be recorded in the logger note). A data table contains columns for tree measurements recorded by tree ID.
number, DBH (inches), distance (feet), two upside/downside clinometer readings for tree height measurements.

**Evaluation of VLF System**

The developed VLF system has been evaluated by different user groups by testing different aspects of the system during several development stages. Evaluation tests consist of qualitative evaluation by an expert group and usability evaluation by a student group. The evaluations were done on a fully implemented version of VLF running under OpenWonderland.

**Evaluation of System during Development**

At an early stage of development, a prototype model was evaluated by experts that are familiar with forestry field experiments and teaching. This evaluation was used to resolve issues on the basic approach and to improve 3D object and interaction design. During the qualitative evaluation, virtual object format and design were discussed and selected from diverse possible graphic formats. Designs of interaction and user interface were issues discussed over the entire development stage, and they have been improved by modifying the system based on results of discussions.

Figure 2-12 shows an example of the modified diameter tape tool. The tool was designed with a grid plane and a HUD interface to display the position of the measured point on the trunk (Fig. 2-12-a). With discussion results on the user interface design, the grid plane was replaced with tape around the tree trunk. And, a new tape for showing the height of measuring position was added to the HUD interface to improve interaction efficiency for users (Fig.2-12-b).
Evaluation of System Performance

With a prototype of the VFL system, the system performance was continuously tested with diverse computer systems, because the VLF took much computer resources to initialize the virtual world with 400 detailed trees requiring heavy, instant memory usage of 370 MB (32 bit system) ~ 560 MB (64 bit system). The tests focused on estimating system performance and system speed, and providing system specification guideline. Table 2-1 shows the diverse computer specifications and the system loading speed results tested during Jul ~ Aug, 2011.

In the table, 8 resource specifications of computer system are presented with information on machine name, operating system, CPU, RAM size, and graphic card. Two types of personal computers, desktops and laptops, are utilized for testing, which are connected on wired or wireless Internet connections. The G3D rating explaining the 3D graphic ability of a graphic card is referred from a video card benchmark site (http://videocardbenchmark.net) as of Jun, 2013. The VLF launching speed is the measured time to completely to display every virtual object in the world. The launching times (except case 4) represents an average time over 20 times of trials at each machine that is utilized at the development phase.

With the machines of case 4, a field test for achieving qualitative evaluation of the system performance was conducted with 20 computers having the same system specification and running under the same network environment at a computer laboratory located in the University of Florida Agriculture and Biological Engineering Department. The test resulted in 22 seconds of average system loading time. Virtual instruments operated the way they were intended, with satisfactory performance speed, and all users could shared and collaborate in the experiments from their individual workstations.
Through tests on diverse computers, the results showed that higher speed of CPU and graphic card contributed to improve the system loading speed as did bigger RAM size. Being a server-client system type system, it was much affected by Internet connection speed at the first loading of virtual components. High specification graphic card showed better system performance, but it also needed good support of fast CPUs and large RAM size.

**Evaluation of System Usability : System Design**

Usability testing and evaluation of lessons learned were conducted with 30 students studying forestry and agriculture at the University of Florida. A survey was used that includes questions about system speed, level of intuitiveness and usefulness, and it was intended to get quantitative scores measuring the value of the system. The evaluation form is presented at Appendix A.

The questionnaire has three categories of questions about; 1) user and computer system (Table 2-2), 2) user experience of virtual experiment (Table 2-3), and 3) user satisfaction for the system (Table 2-4). And, most questions (except about computer system type, level of experience, and simulation time) have a seven-point scale format where 1 means not good, 7 means very good, and 4 indicates middle of satisfaction value.

Category A (Table 2-2) consists of 8 questions about user background and computer system the student was using. Most users have modern computers system manufactured within about the past two to three years, and most have a good graphic supporting system (critical for good performance of the VWE graphics). 90% of students in the survey already had knowledge of field experiments, but they had little prior experience within 3D games (1.2/7.0). Although they have little direct experience
with virtual worlds (1.9/7.0), more students knew about virtual world systems such as Second Life (2.7/7.0). After taking instruction about the VLF system prior to use of the VLF, they showed high expectation for positive result after learning the system (5.3/7.0), and expected to see moderate reality (4.1/7.0) from the system.

In category B (Table 2-3), there are questions about each experiment step with a virtual instrument. Questions were designed to get answers about efficiency and usefulness of forest data gathering experiments in the virtual world. With respects to the user guide and virtual tool box, the tool box design was found to be highly intuitive, but caused problems because of its size and position in the screen. Users reported that the user guide gave helpful tips.

The measuring area tool received good scores on displaying area center pole, tree number tag and boundary lines, but scored lower for realism and intuitiveness because of the complexity of the tool that is composed of three different components. The diameter tape tool obtained good scores on its design and usefulness but also scored lower on intuitiveness. Responses for the logger tape tool showed high satisfaction for realism, intuitiveness and usefulness because of its simple design and ease of use. Results for the clinometer tool showed similar results to the logger tape. For reporting note tool, there are many replies that gave low score on its intuitiveness, but the score of usefulness was high.

Most users had difficulty using virtual instrument at first by following the user guide because they were not familiar with the tools. As they gained more experience, users felt that their proficiency for using the tool increased. This is to be expected, but improved tool design can increase the rate at which students learn how to use the tools.
These results also need to be compared with real world tools (for example, compared with how quickly do students learn to use a real clinometer).

Category C (Table 2-4) consists of 10 questions about overall experience of the virtual experiments. Users reported little difficulty on following instruction in the user guide (3.5/7.0), and items on ease of controlling virtual tools obtain low score (3.2/7.0). Realism of virtual world needs to be improved (3.7/7.0), and students requested that there would be more materials helping user to understand content in the virtual world (3.8/7.0). However, users experienced improvement on their proficiency on using the virtual system. The measurement completion time was reduced from 523 seconds (first trial) to 396 seconds (last trial). They obtained good impression on the implementation of the VLF (4.8/7.0) than their expectation, and were satisfied with completeness of the system (4.4/7.0) and results of the experiment (4.3/7.0).

**Evaluation of System Usability: Learning Efficiency**

Learning efficiency of VLF was tested to evaluate how well students learned from the system. The test measured how the VLF helped students to learn field skills, and the results came from two different tests at different experiment locations. Students conducted two experiments in order of sector A and B, and they had enough time between the experiments to gain experience with virtual tools and procedures. It is assumed that students improved their skills with iterative experiences. The hypothesis imposed in this evaluation design is that the experiment result of sector B will show better accuracy then the experiment result of sector A. Evaluation was conducted by comparison between measurements of tree height and diameter at breast height.

Two experiment locations (Fig. 2-13) are selected randomly in the map of the target forest generated in the virtual world. Sector A is a circular area with 40ft radius
which is located at middle of north-west area of the forest, and there are 19 trees whose heights vary in range from 15 m ~ 24 m and whose DBHs are in range of 14.5 cm - 35.0 cm. Similarly, sector B is located at middle of south-east area of the forest including 16 trees. Tree heights and DBHs are various in range of 21 m - 26 m and 31 cm - 34 cm.

Each student was expected to enter into the virtual forest with given coordinates of center position of each sector A and B. They were instructed to follow below procedures during the experiments.

1. Once logged into the forest, conduct experiments in sector A and sector B.
2. Find the location of center position of each sector with the Navigator tool of the VLF, and then create a sampling area with the Sampling Area tool.
3. Measure the DBH with the DBH tool by seeking the height on the tree from ground with the Height Pole tool, and record the value at the Logger Note tool.
4. By using the Logger Tape tool, find a good location , and measure the height of the tree with the Clinometer tool. And then, record the distance from the tree and height of the tree at the Logger Note tool.
5. Complete the measuring and recording for all tree in the sector A.
6. Record the measured data in the Logger Note tool, and present the data.
7. Compare the experience of using instruments with the user guide for instruments.
8. Repeat the step 2 - 6 for sector B.

There was no given time limits for students to complete the measurements. The test focused on evaluating how accurately users gather data on tree diameter and height using the VLF, and the result came from the comparison of means and standard deviations of known (by the system) tree height and DBH in the two sectors.
The test was conducted by 30 students individually, using the same generated forest (same trees and same tree sizes), and it was designed that experiment of sector B follows by experiment of sector A. Measuring results are summarized in mean and standard deviation of each tree in two sectors, and they are shown in Table 2-5 and 2-6, respectively.

Measurements at sector A are shown in Table 2-5. There are actual and measured value of 19 trees in the sector. Tree number is assigned during generation of a forest in order of tree size from small tree to large tree. It indicates that sector A consists of groups of small tree among 400 trees in the forest. Distribution of most tree heights ranges from 15 m to 20 m. Generally, tree trunk diameter at breast height is proportional to the size of the tree. Measured values are given in mean and standard deviation for tree height and DBH. Average difference of actual tree height and mean of measured values for trees in sector A is 0.486 m, and standard deviation is 2.033. For DBH, average difference of actual tree diameter and mean of measured values is 0.677 cm, and standard deviation is 1.95.

In Table 2-6, measurements at sector B are shown, and they came from measurement of 16 trees. The tree group consists of bigger trees than average size tree among 400 trees in the forest. Average difference of actual tree height and mean of measured values for trees in sector A is 0.410 m, and standard deviation is 1.740. For DBH, average difference of actual tree diameter and mean of measured values is 0.523 cm, and standard deviation is 1.49.

To test the agreement of measurement between actual values and measured values of tree height and DBH, some statistics values indicating the agreement are
calculated and compared. The mean squared error (MSE), the root mean squared error (RMSE), and the mean absolute error (MAE) represent the error between actual value and measured value. Nash-Sutcliffe (1970) model efficiency coefficient and Willmott (1981) agreement index are utilized to describing the accuracy of student measurement. The equations are given below.

\[
MSE = \frac{1}{N} \sum_{i=1}^{N} (Y_i - \hat{Y}_i)^2, \quad RMSE = \sqrt{MSE}, \quad MAE = \frac{1}{N} \sum_{i=1}^{N} |Y_i - \hat{Y}_i|
\]

\[
EF = 1 - \frac{\sum_{i=1}^{N} (Y_i - \hat{Y}_i)^2}{\sum_{i=1}^{N} (Y_i - \bar{Y})^2}, \quad INDEX = 1 - \frac{\sum_{i=1}^{N} (Y_i - \hat{Y}_i)^2}{\sum_{i=1}^{N} (|Y_i - \bar{Y}| + |\hat{Y}_i - \bar{Y}|)^2}
\]

where,

\(Y_i\) : the actual value

\(\hat{Y}_i\) : the measured value by student

\(\bar{Y}\) : the average of the \(Y_i\) values

The MSE and MAE are classical measures of agreement that eliminate the problem of compensation between under- and over- prediction. The MSE has an advantage to be decomposed into separate contributions of error, so it is useful for indentifying the sources of error. Usually, it is convenient to work with the RMSE. On the other hand, the MAE is a measure of agreement to avoid compensation between under- and over-prediction. It is useful to examine overall model error in simple cases.

The EF and the INDEX are utilized to assess the predictive power of student measurement. If the student measurement is perfect, then \(Y_i = \hat{Y}_i\) for each situation i
and EF = 1 and INDEX = 1. Thus, the EF or INDEX value that is close to 1 indicates that the student measurement is close to the actual value.

In Table 2-7, it is shown the comparison results of error and measurement agreement. Errors of student measurement for tree height are reduced from 0.263 (MSE), 0.513 (RMSE) and 0.486 (MAE) at sector A to 0.216 (MSE), 0.465 (RMSE) and 0.410 (MAE). It indicated that the errors of MSE, RMSE, and MAE are improved by 17%, 9% and 15%, respectively. The EF was increased from 0.905 to 0.931, and the INDEX was also improved from 0.975 to 0.983. Both values get closed to 1 that means the perfectness of the measurement. The results show that students are able to improve their measurement accuracy of tree height at sector B with previous experience of sector A.

Meanwhile, errors of DBH student measurement indicated much improvement of students' field skill. The errors of MSE (0.334), RMSE (0.465) and MAE (0.523) at sector B are less than the errors of MSE (0.0.556), RMSE (0.746) and MAE (0.677) at sector A. The improvements of errors, MSE, RMSE and MAE, are 39%, 37%, and 22%, respectively. The EF was increased from 0.977 to 0.982, and the INDEX was also improved from 0.994 to 0.996.

By comparisons of measurement errors and efficiency, the results showed that students improved their measurement for tree height and DBH with reducing errors. Through two experiments at two different sites, it seems that students obtained enough experience at first measurement trial in sector A, and then the proficiency on the virtual tools contributed to enhance student's performance at the later experiment in sector B.
Based on the result of error comparison, it is verified that students improved their virtual field skills with iterative experiences within the VLF.

In the future, further evaluation in which students' experiments at real world and virtual world can be compared may explain how the VLF can contribute to improve students' field skill in real world.

**Summary**

A virtual learning forest (VLF) was developed by using virtual world technology to enhance education experiences of forest data collection (tree size measurements) without restrictions on time, distance and cost. VLF consists of three scene components, tree, terrain, and forest, and virtual instruments for mensuration including a diameter tap, logger tape and clinometer.

Scene components are designed by focusing on realism of the natural environment. Data on the topology of the longleaf pine ecosystem was utilized to build the terrain and forest model. LOD technique is applied to the tree model for providing rich detail of tree in the virtual world from different distances. Virtual instruments mimic field equipment with tool graphic models and a HUD view for reading measurement.

Some evaluations were conducted during the development. The result of expert evaluation for the instrument design helped to enhance user acceptance. The system performance test provided various performance guides according to CPU, RAM and graphic card type. And, a performance test with 20 computers having same specifications showed that the VLF worked in a stable fashion and with good performance. The system usability evaluations for tool design and usage gave a result that the user interfaces of instrument represented reality of the tools well, but more support was needed to help students understand the tools. And, it showed good
acceptance and overall satisfaction for the VLF by students. The evaluation for learning efficiency was conducted at two random sites to compare the improvement of tool use proficiency. The results showed that students reduced the measurement errors at the next site with the experience of learning at the first site.
<table>
<thead>
<tr>
<th>No.</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Machine</td>
<td>Dell Precision 370</td>
<td>Dell Latitude D830</td>
<td>Dell Precision T3400</td>
<td>Dell Precision M6600</td>
<td>Sony Vaio VGN-TT16LN</td>
<td>Sony Vaio VPC</td>
<td>Sony Vaio VGN-Z520N</td>
<td>Apple Macbook</td>
</tr>
<tr>
<td>Machine Type</td>
<td>Desktop</td>
<td>Laptop</td>
<td>Laptop</td>
<td>Desktop</td>
<td>Laptop</td>
<td>Laptop</td>
<td>Laptop</td>
<td>Laptop</td>
</tr>
<tr>
<td>OS</td>
<td>Windows XP Professional (32 Bit)</td>
<td>Windows 7 Enterprise (64 Bit)</td>
<td>Windows 7 Enterprise (64 Bit)</td>
<td>Windows XP Professional (32 Bit)</td>
<td>Windows 7 Professional (32 Bit)</td>
<td>Windows 7 Professional (32 Bit)</td>
<td>Windows 7 Ultimate (64 Bit)</td>
<td>Mac OS X v10.5 Leopard</td>
</tr>
<tr>
<td>CPU (GHz)</td>
<td>2.8</td>
<td>2.2 (dual)</td>
<td>3.2 (dual)</td>
<td>2.5 (dual)</td>
<td>1.4 (dual)</td>
<td>1.8 (dual)</td>
<td>2.26 (dual)</td>
<td>2.4 (dual)</td>
</tr>
<tr>
<td>RAM (GB)</td>
<td>4</td>
<td>4</td>
<td>8</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>Graphic Card</td>
<td>Radeon X1650 (256MB)</td>
<td>NVIDIA Quadro (256MB)</td>
<td>NVIDIA Quadro (256MB)</td>
<td>Mobile Intel® QM67 Express</td>
<td>Mobile Intel® 4 Series (GS45)</td>
<td>Mobile Intel® 82845G</td>
<td>Mobile Intel® NVIDIA GeForce 9300M GS</td>
<td>Intel® GMA X3100</td>
</tr>
<tr>
<td>Internet Connection</td>
<td>Wired</td>
<td>Wired</td>
<td>Wired</td>
<td>Wired</td>
<td>Wireless</td>
<td>Wireless</td>
<td>Wireless</td>
<td>Wireless</td>
</tr>
<tr>
<td>G3D rating&lt;sup&gt;(1)&lt;/sup&gt;</td>
<td>112</td>
<td>258</td>
<td>258</td>
<td>65</td>
<td>40</td>
<td>12</td>
<td>85</td>
<td>122</td>
</tr>
<tr>
<td>(1-8300)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VLF Launching Time&lt;sup&gt;(2)&lt;/sup&gt; (Sec)</td>
<td>24</td>
<td>18</td>
<td>15</td>
<td>22&lt;sup&gt;(3)&lt;/sup&gt;</td>
<td>21</td>
<td>19</td>
<td>14</td>
<td>17</td>
</tr>
</tbody>
</table>

<sup>(1)</sup> Resource from [http://www.videocardbenchmark.net](http://www.videocardbenchmark.net) (as of 06/10/2013), Higher number means better quality.

<sup>(2)</sup> Average spending time to launch the VLF for 20 attempts.

<sup>(3)</sup> Average spending time to launch the VLF for 20 computers.
## Table 2-2. List of questions and scores of evaluation form: Question Category A

<table>
<thead>
<tr>
<th>Item</th>
<th>Questions</th>
<th>Mean</th>
<th>Std</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1</td>
<td>What is the operating system of your computer? (Windows, Mac, Linux)</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>A2</td>
<td>How old is your computer? (can assume CPU, RAM, Graphic card, year)</td>
<td>2.7</td>
<td>1.2</td>
</tr>
<tr>
<td>A3</td>
<td>How long have you played with 3D game? (year)</td>
<td>1.2</td>
<td>0.5</td>
</tr>
<tr>
<td>A4</td>
<td>How well do you know about Virtual World?</td>
<td>2.7</td>
<td>1.1</td>
</tr>
<tr>
<td>A5</td>
<td>How much have you experienced Virtual Learning Environment?</td>
<td>1.9</td>
<td>0.8</td>
</tr>
<tr>
<td>A6</td>
<td>Did you take a class course related with field experiment?</td>
<td>0.9</td>
<td>-</td>
</tr>
<tr>
<td>A7</td>
<td>How much do you expect to get positive results on learning after you are instructed about VLF?</td>
<td>5.3</td>
<td>0.8</td>
</tr>
<tr>
<td>A8</td>
<td>How much reality do you expect to experience before you start VLF?</td>
<td>4.1</td>
<td>1.2</td>
</tr>
<tr>
<td>Item</td>
<td>Question</td>
<td>Mean</td>
<td>Std</td>
</tr>
<tr>
<td>------</td>
<td>--------------------------------------------------------------------------</td>
<td>------</td>
<td>------</td>
</tr>
<tr>
<td>(1)</td>
<td>Step #1 : user guide &amp; VLF tool box</td>
<td></td>
<td></td>
</tr>
<tr>
<td>B1</td>
<td>How do you define the helpfulness of VLF user guide?</td>
<td>4.9</td>
<td>1.1</td>
</tr>
<tr>
<td>B2</td>
<td>How do you define the accessibility of VLF Tool Box?</td>
<td>3.8</td>
<td>1.0</td>
</tr>
<tr>
<td>B3</td>
<td>How do you define the intuitiveness of VLF Tool Box?</td>
<td>6.1</td>
<td>0.9</td>
</tr>
<tr>
<td>(2)</td>
<td>Step #2 : measuring area tool</td>
<td></td>
<td></td>
</tr>
<tr>
<td>B4</td>
<td>How do you define the usefulness of tool instruction at screen?</td>
<td>3.3</td>
<td>0.8</td>
</tr>
<tr>
<td>B5</td>
<td>How do you define the correctness of displaying area center and tree number tags?</td>
<td>5.5</td>
<td>0.7</td>
</tr>
<tr>
<td>B6</td>
<td>How do you define the clearness of area boundary?</td>
<td>6.1</td>
<td>0.5</td>
</tr>
<tr>
<td>B7</td>
<td>How do you define the intuitiveness of the tool usage?</td>
<td>2.8</td>
<td>0.5</td>
</tr>
<tr>
<td>B8</td>
<td>How do you define the realism of the tool?</td>
<td>2.7</td>
<td>0.6</td>
</tr>
<tr>
<td>(3)</td>
<td>Step #3 : DBH tool</td>
<td></td>
<td></td>
</tr>
<tr>
<td>B9</td>
<td>How do you define the intuitiveness of the tool usage?</td>
<td>3.3</td>
<td>0.8</td>
</tr>
<tr>
<td>B10</td>
<td>How do you define the realism of the tool design?</td>
<td>5.2</td>
<td>0.4</td>
</tr>
<tr>
<td>B11</td>
<td>How do you define the usefulness of interface design for reading measurement?</td>
<td>5.9</td>
<td>0.5</td>
</tr>
<tr>
<td>(4)</td>
<td>Step #4 : Distance measuring tape tool</td>
<td></td>
<td></td>
</tr>
<tr>
<td>B12</td>
<td>How do you define the usefulness of tool instruction at screen?</td>
<td>5.2</td>
<td>0.3</td>
</tr>
<tr>
<td>B13</td>
<td>How do you define the intuitiveness of the tool usage?</td>
<td>5.6</td>
<td>0.3</td>
</tr>
<tr>
<td>B14</td>
<td>How do you define the realism of the tool design?</td>
<td>4.5</td>
<td>0.8</td>
</tr>
<tr>
<td>B15</td>
<td>How do you define the usefulness of interface design for reading measurement?</td>
<td>5.8</td>
<td>0.4</td>
</tr>
<tr>
<td>(5)</td>
<td>Step #5 : clinometer tool</td>
<td></td>
<td></td>
</tr>
<tr>
<td>B16</td>
<td>How do you define the usefulness of tool instruction at screen?</td>
<td>5.6</td>
<td>0.8</td>
</tr>
<tr>
<td>B17</td>
<td>How do you define the intuitiveness of the tool usage?</td>
<td>6.2</td>
<td>0.5</td>
</tr>
<tr>
<td>B18</td>
<td>How do you define the realism of the tool design?</td>
<td>5.3</td>
<td>0.9</td>
</tr>
<tr>
<td>B19</td>
<td>How do you define the usefulness of interface design for reading measurement?</td>
<td>4.9</td>
<td>1.2</td>
</tr>
<tr>
<td>(6)</td>
<td>Step #6 : reporting note tool</td>
<td></td>
<td></td>
</tr>
<tr>
<td>B20</td>
<td>How do you define the usefulness of tool instruction at screen?</td>
<td>4.7</td>
<td>1.4</td>
</tr>
<tr>
<td>B21</td>
<td>How do you define the intuitiveness of the tool usage?</td>
<td>2.5</td>
<td>1.2</td>
</tr>
<tr>
<td>(7)</td>
<td>Step #7 : iteration of experiments over target area</td>
<td></td>
<td></td>
</tr>
<tr>
<td>B22</td>
<td>How do you define the easiness of finding area centers by location tool?</td>
<td>3.1</td>
<td>0.8</td>
</tr>
<tr>
<td>B23</td>
<td>How do you define your learning proficiency as iterating experiments through areas</td>
<td>4.2</td>
<td>0.9</td>
</tr>
</tbody>
</table>
Table 2-4. List of questions and scores of evaluation form: Question Category C

<table>
<thead>
<tr>
<th>Item</th>
<th>Questions</th>
<th>Mean</th>
<th>Std</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1</td>
<td>How do you evaluate the VLF instruction taken before conducting the virtual experiment?</td>
<td>3.5</td>
<td>0.9</td>
</tr>
<tr>
<td>C2</td>
<td>How long did you take to complete first measurements at first experiment area? (sec)</td>
<td>523.0</td>
<td>23.5</td>
</tr>
<tr>
<td>C3</td>
<td>How long did you take to complete last measurements at last experiment area? (sec)</td>
<td>396.0</td>
<td>19.8</td>
</tr>
<tr>
<td>C4</td>
<td>How much do you define the effect of computer performance?</td>
<td>4.5</td>
<td>0.8</td>
</tr>
<tr>
<td>C5</td>
<td>How much do you define the easiness of controlling virtual instruments?</td>
<td>3.2</td>
<td>0.9</td>
</tr>
<tr>
<td>C6</td>
<td>How much difference do you feel between your expectations for VLF and actual implementation?</td>
<td>4.8</td>
<td>0.7</td>
</tr>
<tr>
<td>C7</td>
<td>How much do you define the completeness of the VLF system?</td>
<td>4.4</td>
<td>1.1</td>
</tr>
<tr>
<td>C8</td>
<td>How much do you define the realism of the virtual contents?</td>
<td>3.7</td>
<td>1.2</td>
</tr>
<tr>
<td>C9</td>
<td>How much do you define the value of system supports for understanding the VLF?</td>
<td>3.8</td>
<td>0.9</td>
</tr>
<tr>
<td>C10</td>
<td>How much are you satisfied with the VLF?</td>
<td>4.3</td>
<td>1.5</td>
</tr>
</tbody>
</table>

Table 2-5. List of actual and measured tree height and DBH at experiment sector A.

<table>
<thead>
<tr>
<th>Tree Number</th>
<th>Actual Height (m)</th>
<th>Measured Value</th>
<th>DBH (cm)</th>
<th>Measured Value</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Actual Value</td>
<td>Mean</td>
<td>Stdev</td>
<td>Actual Value</td>
</tr>
<tr>
<td>12</td>
<td>15.24</td>
<td>14.58</td>
<td>2.216</td>
<td>14.48</td>
</tr>
<tr>
<td>35</td>
<td>16.46</td>
<td>16.01</td>
<td>1.950</td>
<td>22.61</td>
</tr>
<tr>
<td>36</td>
<td>16.76</td>
<td>17.31</td>
<td>2.488</td>
<td>11.94</td>
</tr>
<tr>
<td>49</td>
<td>17.07</td>
<td>17.24</td>
<td>2.123</td>
<td>16.51</td>
</tr>
<tr>
<td>52</td>
<td>17.09</td>
<td>16.47</td>
<td>2.274</td>
<td>19.81</td>
</tr>
<tr>
<td>53</td>
<td>17.37</td>
<td>16.79</td>
<td>2.433</td>
<td>13.46</td>
</tr>
<tr>
<td>71</td>
<td>17.68</td>
<td>18.41</td>
<td>2.265</td>
<td>19.30</td>
</tr>
<tr>
<td>86</td>
<td>18.29</td>
<td>18.89</td>
<td>2.868</td>
<td>23.62</td>
</tr>
<tr>
<td>94</td>
<td>18.59</td>
<td>18.10</td>
<td>2.045</td>
<td>23.37</td>
</tr>
<tr>
<td>96</td>
<td>18.90</td>
<td>19.55</td>
<td>2.368</td>
<td>16.51</td>
</tr>
<tr>
<td>108</td>
<td>19.24</td>
<td>18.74</td>
<td>2.637</td>
<td>17.27</td>
</tr>
<tr>
<td>113</td>
<td>19.20</td>
<td>19.72</td>
<td>2.062</td>
<td>22.86</td>
</tr>
<tr>
<td>118</td>
<td>19.51</td>
<td>19.27</td>
<td>2.271</td>
<td>20.83</td>
</tr>
<tr>
<td>138</td>
<td>19.81</td>
<td>20.19</td>
<td>2.336</td>
<td>35.05</td>
</tr>
<tr>
<td>143</td>
<td>20.12</td>
<td>19.64</td>
<td>2.477</td>
<td>29.05</td>
</tr>
<tr>
<td>147</td>
<td>20.22</td>
<td>20.92</td>
<td>2.023</td>
<td>24.13</td>
</tr>
<tr>
<td>173</td>
<td>20.63</td>
<td>20.27</td>
<td>2.145</td>
<td>23.37</td>
</tr>
<tr>
<td>174</td>
<td>20.73</td>
<td>20.90</td>
<td>2.333</td>
<td>24.38</td>
</tr>
<tr>
<td>355</td>
<td>24.08</td>
<td>23.68</td>
<td>2.033</td>
<td>30.99</td>
</tr>
</tbody>
</table>
### Table 2-6. List of actual and measured tree height and DBH at experiment sector B.

<table>
<thead>
<tr>
<th>Tree Number</th>
<th>Tree Height (m) Actual Value</th>
<th>Tree Height (m) Measured Value</th>
<th>DBH (cm) Actual Value</th>
<th>DBH (cm) Measured Value</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Actual Value</td>
<td>Mean</td>
<td>Stdev</td>
<td>Mean</td>
</tr>
<tr>
<td>209</td>
<td>21.34</td>
<td>21.11</td>
<td>1.850</td>
<td>30.99</td>
</tr>
<tr>
<td>211</td>
<td>21.64</td>
<td>21.08</td>
<td>1.705</td>
<td>21.34</td>
</tr>
<tr>
<td>218</td>
<td>21.44</td>
<td>21.88</td>
<td>1.847</td>
<td>27.69</td>
</tr>
<tr>
<td>259</td>
<td>22.25</td>
<td>22.16</td>
<td>1.732</td>
<td>32.26</td>
</tr>
<tr>
<td>277</td>
<td>22.56</td>
<td>23.12</td>
<td>1.894</td>
<td>28.70</td>
</tr>
<tr>
<td>279</td>
<td>22.56</td>
<td>21.89</td>
<td>1.790</td>
<td>29.72</td>
</tr>
<tr>
<td>282</td>
<td>22.56</td>
<td>23.52</td>
<td>1.968</td>
<td>30.48</td>
</tr>
<tr>
<td>298</td>
<td>23.16</td>
<td>22.96</td>
<td>1.809</td>
<td>26.42</td>
</tr>
<tr>
<td>315</td>
<td>23.47</td>
<td>23.78</td>
<td>1.496</td>
<td>21.59</td>
</tr>
<tr>
<td>317</td>
<td>23.47</td>
<td>23.13</td>
<td>1.821</td>
<td>24.64</td>
</tr>
<tr>
<td>347</td>
<td>24.08</td>
<td>23.78</td>
<td>1.736</td>
<td>24.89</td>
</tr>
<tr>
<td>383</td>
<td>25.30</td>
<td>24.62</td>
<td>1.639</td>
<td>33.02</td>
</tr>
<tr>
<td>385</td>
<td>25.60</td>
<td>25.18</td>
<td>1.543</td>
<td>27.18</td>
</tr>
<tr>
<td>388</td>
<td>25.60</td>
<td>25.84</td>
<td>1.847</td>
<td>33.02</td>
</tr>
<tr>
<td>393</td>
<td>25.91</td>
<td>25.56</td>
<td>1.413</td>
<td>33.22</td>
</tr>
</tbody>
</table>

### Table 2-7. Statistical values representing errors and measurement agreements.

<table>
<thead>
<tr>
<th>Agreement</th>
<th>Tree Height</th>
<th>Diameter (DBH)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Sector A</td>
<td>Sector B</td>
</tr>
<tr>
<td>MSE(1)</td>
<td>0.263</td>
<td>0.216</td>
</tr>
<tr>
<td>RMSE(2)</td>
<td>0.513</td>
<td>0.465</td>
</tr>
<tr>
<td>MAE(3)</td>
<td>0.486</td>
<td>0.410</td>
</tr>
<tr>
<td>EF(4)</td>
<td>0.905</td>
<td>0.931</td>
</tr>
<tr>
<td>INDEX(5)</td>
<td>0.975</td>
<td>0.983</td>
</tr>
</tbody>
</table>

(1) Mean Squared Error  
(2) Root Mean Squared Error  
(3) Mean Absolute Error  
(4) Nash-Sutcliffe model Efficiency  
(5) Willmott Agreement Index
Figure 2-1. An example of conceptualization of forest learning system. It represents a system concept diagram.
Figure 2-2. An example of conceptualization of a forest learning system. It represents system interactions and events.
Figure 2-3. Virtual Learning Forest (VLF) world Client/Server architecture showing independent and extendable relationships between VWE and developed virtual world elements.
Figure 2-4. A virtual object of the sampling area tool. It consists of a graphic model (center pole, tree name tag, area boundary) and a rendering model to handle boundary size and location.
Figure 2-5. A virtual object of the logger tape. It contains a graphic model for tape, a process to define distance from target, and a rendering model to handle changes of shape by moving position.
Figure 2-6. A virtual object of the clinometer which is used to measure the height of tree. It contains a process to define measurement based on angle of view.
Figure 2-7. A virtual object of the diameter tape. Diameter is calculated by intersecting a grid object to a tree object, and it results in showing diameter value in display and appearing tape line around tree.
Figure 2-8. The Welcome Board that is displayed at the start of each session.
Figure 2-9. The Tool Bag of the instruments that are available for use in the VLF.
Figure 2-10. An user interface object of User Guide.
Figure 2-11. The Logger Note used by a student to record tree measurements made with the instruments.
Figure 2-12. An example of enhancement of user interface design and interaction: Diameter tool - (a) before modification, and (b) after modification.
Figure 2-13. Locations of two experiment sectors in the virtual forest.
CHAPTER 3
VIRTUAL WORLD DESIGN APPROACH FOR ABSTRACTING AND CHARACTERIZING BASIC VIRTUAL OBJECTS

Background

In the previous chapter, a prototype VLF is developed in the traditional way of implementation by manual coding following the proprietary model protocol of the VWE. Three types of virtual objects are developed, which are 1) terrain object, 2) forest object and 3) instrument object. A virtual object of terrain is an object focusing on a graphic model and its geological information, while a forest object is a combined model of the graphic model of a tree and dynamic model simulating size and spatial distribution of a population of trees by generating geological and physical information for each tree. Virtual objects of instruments are objects which stress their behaviors including interactions with the controller (avatar) and communication with target objects as well as a graphic model of instrument shape and an interface for representing data.

With the traditional development methods, virtual objects are expressed in the form of structures oriented in the VWE, which wrap complicated information of relationships between graphic models, behaviors and processes into a proprietary format. But, it results in an inflexible structure and makes it difficult to create a new virtual world or modify processes in a virtual object. Behaviors of virtual objects are implemented by using programming languages and expressions required by the specific VWE, so it is difficult to understand and share them efficiently. Processes (logics) of calculation within virtual objects like instruments can be represented in an independent form like mathematical expressions to enhance extendibility of model by using tag-based expressions or web-based services.
In this chapter, some issues of proprietary model complexities will be described, including model structure, data flow and graphic format. And, it will be shown that flexibility of virtual objects can be enhanced by abstraction of virtual objects implemented, in particular the object’s graphic model, behavior and model processes can be represented in an abstract, platform independent manor.

Complexities of Proprietary Model for VWE

In the previous chapter, the prototype VLF was developed. The development of virtual objects follows the proprietary format of OpenWonderland. It causes some limitations of model structure, data flow and graphic model to reuse the objects in different virtual worlds and to share them with other VWEs.

Model Structure

OpenWonderland provides a proprietary way to develop virtual objects. To represent detail, object specific behavior or process it is usually implemented as a module object, a virtual object. It is efficient to develop a complex model by using methods supported in the specific VWE, but it causes difficulties for sharing information and representing behavior and process clearly, or moving the virtual objects to other VWE platforms. That usually requires that the virtual objects be rewritten entirely in the language required by the platform.

In the development of the prototype virtual world, environment objects (terrain and forest) are implemented as module objects, and instrument objects are designed to use additional HUD components in addition to a graphic model representing a device. They are represented in solid, concrete formats, in which behavior, process and graphic model are mixed complicatedly. It is difficult to distinguish them from fundamental structure explicitly.
For example, to show the pattern of plug-in structures, two class diagrams of virtual objects are shown in Fig. 3-1 and 3-2. Fig. 3-1 illustrates classes which represent a forest virtual object following the proprietary format of OpenWonderland. Colored classes are required classes for representing a forest module in the VWE. PineForestCellRenderer class includes a description of the tree graphic model resource, forest generation process and dynamic LOD tree model control, which are represented in external classes. Terrain object and graphic models representing instruments use similar format with the forest object.

In the OpenWonderland, a virtual object can have a window-based interface, a HUD component, which can be used for presenting data. Fig. 3-2 shows the class structure which represents a HUD object for an instrument (diameter tape) in a proprietary OpenWonderland format. A HUD object can include classes for view manager, window decoration, and manipulation of external virtual object. DiameterJPanel class describes window components of HUD object such as size, position, and contents of HUD window. DiameterTapeEntity class represents graphic entities that are associated with the HUD object. The graphic entities include a diameter geometry model and an instance, internal diameter tape around the tree trunk. It describes the model loading method, location and transformation, and dismissal. The diameter tape object is created with segments of line graphic component.

**Data Flow**

It is possible that there are different architectural designs for a virtual world depending on the purpose of the system. A client architecture design for data-centered representation of an instrument object is shown in Fig. 3-3. It is utilized for the development of the prototype VLF. Actions of user and objects are represented as solid
lines, and dash lines describe information paths. The 3D instrument object (for sharing appearance) and HUD window (for data representation) are separated for efficient manipulation of shape and data representation respectively. A HUD window is the main controller for a user when the instrument is activated, and a graphic model of object is instantiated and attached to an avatar.

The measuring process is invoked by user's action such as a picking location on the target object to be measured. Location data are sent to the HUD window in the client side, data is calculated, translated, and presented by following a pre-defined interface specified by a developer. The user can utilize a rich representation by using the HUD window for his/her own use, but the client side requires many resources for data representation.

An ontology is a way to map the abstracted data and relations into a data structure, and it captures data in complicated relationships. The ontology can be used to create data structures for a virtual object. A model integrates graphic model, data representation and logic in the instrument virtual object, and manipulates information through a context interface communicating with an ontology manager on the server side. Required resources in the client side can be reduced and an object can be formed by utilizing a highly abstract process. What is needed is work on reducing the burden on the server, enhancing data representation, and developing a generic, rich ontology for interoperability.

Fig. 3-4 shows a client architecture design for a model-centered representation of instrument objects. It is an alternative to the data-centered representation. It focuses on the role of the model rather than data flow. Although the data-centered representation
shows that each object such as a HUD component and virtual object model manages
data and relation inside of them, the structure of data control is complex and it is difficult
to build the data flow that is independent to the based VWE. Meanwhile, a model-
centered representation gathers related information in the model and simplifying data
flow.

**Graphic Model Format**

Many VWEs are restricted in support of multiple graphic formats, because they
are based on a specific graphic engine. OpenWonderland supports graphic formats of
X3D and COLLADA, while Croquet utilized a OGRE (Object-oriented Graphic
Rendering Engine, www.ogre3d.org) file format, and Second Life has its own graphic
elements which can be used internally.

A graphic format contains data for graphic element structure and graphic engine
specific attributes for controlling graphic elements and effects such as animation and
scripts. And, it is usually formed as a runtime object (a binary file), so it need a specific
program library to interpret its contents. It can be converted into other formats by some
Digital Contents Creation (DCC) software such as Blender, but there are limitations on
file format which can be supported.

The efforts to create a compatible graphic format focused on representing only
common graphic elements like geometry and material, and COLLADA is an example. It
excludes graphic engine oriented attributes from the contents, and It describes the
elements in a human readable form such as XML representation. Although its simplified
contents description provides limited supports to proprietary attributes, the approach
can be utilized to represent a graphic model in a diverse data description form such as
ontology which is independent of graphic format and graphic engine.
Conceptualization of the VLF Educational Objects

The prototype VLF is an e-Learning system that attempts to improve student field skill with an educational virtual world. The system includes many complicated, diverse concepts in a virtual world. For an example, a small area of a longleaf pine forest ecosystem was built with a terrain object and longleaf pine trees. The forest imposes information of population, distribution and tree size. Educational field instruments were created by mimicking the shape (geometry model) and use in the real world.

As an educational learning system, it populated with many actors such as student, teacher and system manager, and there were many complicated relationships between the actors and their activities against environmental objects (e.g. nature) and equipment objects (e.g. instrument). By expanding the scope of interests on the system to the educational environment, it is possible to define the role of objects in the system and to capture detailed features of them (Checkland, 1999).

As an example of the systematic approach, Soft System Methodology (SSM) (Checkland, 1999) provides a framework for understanding the complicated system situation. SSM is a methodology for analyzing complex problems, and is useful for developing system models such as the VLF. It uses iterative development cycles including system requirements to develop the design of the system, and the methodology makes it possible to implement a prototype in a short period. And, with a recursive iteration of analysis, design and implementation, system requirements are realized and implemented rapidly. Usually, SSM consists of several steps. The problem in the domain is described with a rich picture, and possible solutions are defined to improve the problem situation. A conceptual model is developed, which is based on the
activities of the actors in the domain. And, implementation is followed by recursive comparisons with reality and improvement of the model.

In order to conceptualize the role and functionality of virtual objects, some steps of the SSM are adopted to analyze the Virtual Learning Forest. Domain interests surrounding the VLF are described and the situation is conceptualized. The role and activities of virtual objects are specified by creating some modeling diagrams.

**Domain Interests**

Traditional education methods for teaching field skills of measuring and sampling in forest have faced problems of cost, time and space. It is difficult to conduct laboratory exercise in a wide variety of interesting places. Field experiment requires cost for transportation, as if the field of study may be located far away. Even though limitations of cost and field may be resolved, weather and safety may be other limiting factors.

Changes of educational policy or situation stimulate the development of the Virtual Learning Forest system in the school domain. The system consists of a server, virtual forest libraries, a model ontology for tree growth and forest generation, and GIS data of the terrain and forest. The school domain includes teacher, student, system manager, and researcher. Other schools and international institutes cooperate to enhance the learning program and assets of the virtual environment.

School or system manager respond to the requirements of education methodologies according to changes in educational situation, policy, and technical improvements. Requirements from the community determine the direction and content of education in school. Teacher provides educational material to the learning system and sets plans for field experiments. Student use the learning system for learning and developing field skill through the virtual environment. With responses from student and
feedback from other users, researcher evaluates the virtual learning system and learning contents. Results of evaluation are used for improving the learning system and enhancing the learning environment.

**Conceptualization**

The problem specification comes from various actors such as the system manager, teacher, student and school. Student want to understand teaching material fully, and teacher may want to have a new system which can be managed easily. The school may need to spend less money to build systems for teaching student. A new system can be built for virtual field experiments overcoming limitations of real field trips on time and cost.

Investigation of system requirements leads to identifying necessary activities of the system. It identifies flows of main activities to achieve the goal of the system (service Virtual Learning Forest). Serving the VLF requires aggregating related activities like managing the forest ontology, generating the forest, and developing virtual instrument.

**Model Representation**

To specify and characterize virtual objects for the virtual learning forest system, some processes to construct model are needed. They are represented as Unified Modeling Language (UML) diagrams including use case, sequence, and collaboration diagrams.

Fig. 3-5 illustrates a use case diagram of the virtual learning forest system, which gives information of actors and activities in the system. Key elements (actors and use cases) are described below.
- System manager creates virtual forest and virtual instruments and manages them in the learning system.

- Teacher uses the virtual learning system to plan contents of virtual field experiment.

- Student uses the virtual learning system for learning and developing field skills.

- Researcher evaluates the virtual learning system and contributes to develop diverse virtual instrument for enhancing the capability of teaching student.

- Other institute shares the virtual learning system for teaching and cooperate for enriching the virtual learning environment.

A sequence diagram of the virtual forest learning system is shown in Fig. 3-6. It includes subjects, objects and sequences described in the Use Case diagram, and instrument objects and tree object are represented to describe required information and activities. Student and teacher are two subjects in the diagram. Student use instruments in the order of sequences as described in the diagram. The instruments are represented in order of use in a field experiment, and their activities are numbered in according to the order of subject's use.

Two objects, diameter tape and logger tape, communicate with the tree object to get values of measurement by using the instruments. Teacher is involved with two sequences to generate trees in the virtual world and to evaluate measurements by students.

A collaboration diagram (Fig. 3-7), an interaction diagram, describes the structural organization of the objects that send and receive messages in the virtual forest learning system. The sequence diagram and collaboration diagram are semantically equivalent, but they each provide a different aspect. Collaboration diagrams focus on the structure and control patterns of objects and their relations with
other objects, while sequence diagrams highlight the time sequence of activities which are useful in describing Use Case Scenarios.

Students pass the position information of the instruments to locate the instruments (sampling area tool and height measure pole) or to get measurements from the instruments (diameter tape, logger tape, and clinometer). A logger note receives entry values from students. In similar ways, some instruments (diameter tape, logger tape, and clinometer) use position information to get actual location information from the terrain or tree. Teachers generate a forest by passing simulation information to the terrain and tree models, and gets reports of measurements for the student. Teacher gives evaluation result to student and can get responses of student about the learning experiences.

**Abstraction of Objects into Form of Virtual Objects**

A virtual world consists of various virtual objects that represent things (e.g. tree) or concepts (e.g. user's action to target object) that exist in real world. Usually creating a virtual object involves processes of specifying their shape (geometry) and function and generalizing properties and relationships.

In this section, virtual objects in the VLF are identified and defined and then their features are described, and it includes objects of tree, logger tape, diameter tape clinometer, and sampling area tool. It is explained how each object works and how it relates to other processes. And, it is described how the virtual object is created based on the real object and utilized in the virtual world.

**Tree Virtual Object**

It is assumed that longleaf pine is the only tree species in the virtual world in order to simplify and reduce the complexity of real world ecosystem. Longleaf pine is a
pine native to the southeastern United States (Fig. 3-8). It is used for high-value wood products, and it is known that ecosystem of longleaf pine is unique and important to wildlife (Peet and Allard, 1993). A longleaf pine tree typically reaches a height of 30 - 35 m and a diameter of 0.7m. The bark is thick, reddish-brown, and scaly. The leaves are dark green and needle-like, and occur in bundles of three.

A longleaf pine tree geometry model (Fig.3-9) was created with SpeedTree® (SpeedTree, 2002), a commercial 3D tree library. The tree digital content was modified by reallocating shapes and adjusting resolution of textures to refine the quality level of the model.

It consists of trunk, branch and leaf elements, but no root. The trunk geometry is a cone-shaped hollow cylinder, with the cross section being an irregular circular shape. Elements branch and leaves are represented by a rectangular plane and pyramid-shaped plane with highly detailed figure-based texture, so there is no cross-sectional information about them.

The prototype tree model represents an example of a longleaf pine tree with age between 25 - 30 years old. Various sized and aged longleaf pines are generated from the prototype tree model by applying transformations to the size of the prototype model, and the transformation information is randomly developed based on a longleaf pine height-diameter distribution table developed from a natural stand of longleaf pines (Leduc and Goelz, 2009). With the 3D tree digital content, three more different types of tree models are created to apply the discrete LOD to the tree model. The three 2D plane tree models use an image of longleaf pine as a texture. They are set to appear at the range of 0m, 30m, 60m and 90m (Fig. 3-10).
Based on the analysis of tree object, Table 3-1 shows the abstraction of tree virtual object in according to categories of view, control, process and data. The tree object appears in 4 different tree models, the views are controlled by LOD system. A model is generated by forest generation process. It uses a calculation process for tree population. With input of tree age data, tree location and shape are created.

**Clinometer Virtual Object**

A clinometer is an instrument for measuring the angle of a slope. In forestry it is commonly used to measure slope, vertical angles, and tree heights. For measuring tree heights a clinometer has an optical reading lens representing heights in different scales (percent, distance) by converting from the angle measured and knowing the distance from the point of observation to the tree being measured.

There are different types of clinometers for measuring only angle, only height or both. For an example, SUUNTO ([http://www.suunto.com](http://www.suunto.com)) provides various clinometers used in forestry. The SUUNTO PM 5 series presented diverse clinometer models distinguished with types of scales (degree, percent, and distance) and different clinometer body shapes. As an example, the SUUNTO Model PM-5/66 PC provides a scale of angle (degree) on a side window, a percent scale (0 +/- 150 % based on distance to target) at the left side in the optical lens, and a height value scale (1:66, height measured at 66ft distance) at the right side in the optical lens.

Fig. 3-11 shows major parts of the Suunto PM-5 clinometer. The optical pin hole provides two scales for reading height instantly and a cross hairline directing measuring value. On the side of the body there is side window which consists of a circular angle scale, hairline for angle, and labels of left/right scale in pinhole. On the opposite side window a conversion table is provided for providing angle, distance or percent instantly.
To measure the height of a tree (e.g. Suunto PM-5), an observer needs to know the distance from the tree. The observer can read angle or percent of height over the particular distance by aligning the optical pinhole to measuring the angle to the top of the tree. The real height can be calculated by multiplying real distance to the measured height percent value.

\[ H = \tan(\alpha) \times D = \frac{\text{percent}}{100} \times D \]

where, \( \alpha \) is a reading view angle, \( D \) is a distance from tree, and percent is a reading percentage value on clinometer scale.

A clinometer geometry model is developed in 3D, which represents a Suunto PM-5 clinometer (Fig.3-12). It includes a set of geometries such as scale wheel, box frame, pin-hole frame, scale guide pane and pin-hole screen (Fig.3-13). Scale wheel is in a shape of a cylinder, and it has an angle scale on the flat side surface and a bar scale on the curved surface. Zero index of side angle scale leans toward the center of the earth always under the influence of gravity.

Based on the analysis of clinometer object, Table 3-2 shows the abstraction of clinometer virtual object in according to categories of view, control, process and data. The object can utilize a HUD interface to show a measurement with a pinhole with two scales, Model appearance control is conducted within the Tool Bag, a HUD interface, and A Pin Hole HUD interface is utilized to control the measurement window. There are processes for obtaining view angle from the measurement HUD window and for calculating a height with the angle.

**Logger Tape Virtual Object**

A logger tape is an instrument for measuring the distance between objects or length of a object, much like a common household measuring tape. For forestry field
experiment, it is used to measure the distance from tree in feet or meter scale. Generally, the tape is used for measuring short lengths or distance up to 100 feet.

There are different types of logger tape, but they have similar shapes and simple structure. Most of them consist of a metal or plastic box body and metal (nylon or plastic) strip ruler with linear-measurement markings. A self-retracting button is a part of box body. A typical measuring tape is capable of measuring down to 1/32 inch (0.079375 mm), and double-scales rulers are applied such as inch, feet and meter. In Fig. 3-14, a common measuring tape is shown with a description of major parts of tape.

Measuring a distance starts from attaching the end of metal-strip ruler to target object and end by reading the value of the ruler when user stands at target location. In the virtual world, it is complex to implement a sagging effect of tape or represent details of ruler in the small face. Simply, without considering a tape sagging effect, a distance from a target to user location can be calculated like below equation.

\[
D = \sqrt{(TPV - SPV)^2}
\]

where, D is a distance, TPV is a target 3D position, and SPV is a source 3D position.

A geometry model for a logger tape was created in 3D model, for a Stanley Powerlock (Fig. 3-15). It is developed as a set of geometries including tape box body, retracting button and metal-strip ruler.

Based on the analysis of Logger Tape object, Table 3-3 shows the abstraction of Logger Tape virtual object in according to categories of view, control, process and data. A graphic model represents a diameter tape, and a tape strip to target object is created.
as a temporary object within a VWE. Mouse picking action controls the location of the graphic model, and provides the location to distance calculation process.

**Diameter Tape Virtual Object**

A diameter tape (D-tape) is a form of dendrometer which consists of a cloth or metal tape and tape box. It is mainly used to measure diameter of the tree at breast height (DBH). Standard DBH is measured at a height of 4.5 feet above the ground because it is known that it is convenient to measure diameter at the height.

Diameter can be read directly from the scale on the tape because the tape is calibrated in units of 3.14 (PI) inches or centimeters. The measured circumference of tree is transformed in to a diameter by dividing the circumference with PI. The obtained diameter is an approximate value because it is assumed that the tree trunk cross-sections are perfect circles. In Fig. 3-16, a diameter tape is shown with a description of major parts, and there is a figure to show how to read the measurement with diameter tape.

To measure the diameter of a tree, the tape is wrapped around the tree at 4.5 feet above ground, and the diameter of the tree is a value at the tape where the number '0' aligns with the rest of the tape. Although the cross-section at a measured height usually has no exact circular shape, the tape provides a diameter based on an assumption that the tree trunk has a circular shape. The diameter is a calculated value, and as an example it can be represented by below equation.

\[ \text{Dia} = \frac{\text{CF}}{\text{PI}} \]

where, Dia is a diameter, CF is a circumference of tree trunk, and PI is 3.14.

A diameter tape geometry model is created in 3D, and it is shown in Fig 3-17. It is a simple tape model which consists of tape box body, retracting button and tape ruler.
Based on the analysis of Diameter Tape object, Table 3-4 shows the abstraction of Diameter Tape virtual object in according to categories of view, control, process and data. A temporary model, tape strip, is created around the tree truck, and a HUD interface is utilized to show reading measurement of diameter and the measuring height. The model is handled through input onto the Tool Bag HUD interface. Mouse picking action provides location data for three calculation processes for circumference of tree trunk, diameter based on the circumference, and measuring height from ground. They results in a measured diameter and height.

**Sampling Area Virtual Objects : Center pole, Boundary tape, and Tree tag**

A sampling area is an area with a boundary within which tree measurements are taken (usually for all trees within the sampling area). The sampling area is usually designed and planed on paper map or with GIS software in the shape of a grid or circles. There is no magic tool for this work, so many simple tools such as paint markers, maps and GPS are used together.

The location of sampling areas can be located by GPS, and boundaries of the area are marked at the tree trunk by using paint. Plastic tree identification tags are placed on each tree to be measured for use in identification. In Fig.3-18, materials and tools used for assigning a sampling area are shown.

For educational field experiments, circular sampling areas can be determined with a given center location and radius, and the boundary can be marked with paint on the tree. For example, point locations (e.g. \((x,y)=(BPX, BPY)\) ) on the circle boundary can be generated with below equation.
\[( BPX - SACX )^2 + ( BPY - SACY )^2 = RAD^2 \]

where,

BPX, BPY : Values of boundary points

SACX, SACY : X, Y value of sampling area center points

RAD : Radius of sampling area

Similarly, to attach tree tags in the sampling area it is necessary to locate all trees existing inside of the boundary. They can be found with below formula.

\[ X = (TLX, TLY) \quad \text{if} \ ( TLX - SACX )^2 + ( TLY - SACY )^2 - RAD^2 \leq 0 \]

where,

X : Tree location point

TLX, TLY : X, Y value of tree location

SACX, SACY : X, Y value of sampling area center points

RAD : Radius of sampling area

The sampling area tool consists of three tools: center pole, boundary tape and tree identification tag. Area center pole is created in 3D digital content (Fig. 3-19). Boundary tape and tree identification tag are temporary objects, which change their size, shape and contents as the location of area center changes.

Based on the analysis of Sampling Area Tool object, Table 3-5 shows the abstraction of Sampling Area Tool virtual object in according to categories of view, control, process and data. A graphic model, Center Pole is utilized to represent the area center, and temporary models like boundary tape and tree ID tags are created internally. A HUD interface is used for entry of sampling area radius. Two calculation processes refer to the location picked by mouse action, which are calculations for boundary tape
location intersecting with ground surface away at area radius and for tree ID tag location intersecting with tree within sampling area.

**Summary**

This chapter explores ways to abstract and characterize virtual objects in the agriculture and natural resource domain. As an example, the developed VLF system in the previous chapter was analyzed. The roles of objects in the system are captured by adopting a systematic approach, and further details features and functionality of objects are obtained by creating modeling diagrams such as use case diagram, sequence diagram, and collaboration diagram. Virtual objects such as longleaf pine tree, logger tape, diameter tape, clinometer, and sampling area tool are analyzed by abstracting features, functionality, relationships, and data. Analysis of proprietary model structures gave ways to indentify properties related with proprietary structure of model and to distinguish the independent properties among others. And, two model representations of data-centered and model-centered are compared to enhance the data flow structure and virtual object design. A model-centered representation can simplify the data flow, and capture complicated data structure with ontology.

The abstracting and characterizing process can help to indentify the role of elements in the model. The identification information can lead to developing flexible and extendible models. The approach abstracts complicated information about relationships between graphic models, behaviors and processes from a proprietary format.
Table 3-1. Abstraction of Tree virtual object.

<table>
<thead>
<tr>
<th>Category</th>
<th>Items</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>View</td>
<td>Graphic Model</td>
<td>4 different tree models with different detail of appearance</td>
</tr>
<tr>
<td>Control</td>
<td>Model</td>
<td>Forest generation process</td>
</tr>
<tr>
<td></td>
<td>View</td>
<td>LOD system, 4 detail levels</td>
</tr>
<tr>
<td>Process</td>
<td>Calculation</td>
<td>Tree population</td>
</tr>
<tr>
<td>Data</td>
<td>Input</td>
<td>Tree age</td>
</tr>
<tr>
<td></td>
<td>Internal</td>
<td>Model transformation information</td>
</tr>
<tr>
<td></td>
<td>Output</td>
<td>Tree Location</td>
</tr>
</tbody>
</table>

Table 3-2. Abstraction of Clinometer virtual object.

<table>
<thead>
<tr>
<th>Category</th>
<th>Items</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>View</td>
<td>Graphic Model</td>
<td>4 geometry parts, Pin Hole, Scale Wheel, Body, Side Scale</td>
</tr>
<tr>
<td>Control</td>
<td>Model</td>
<td>Activate/inactivate by Tool Bag, a HUD interface</td>
</tr>
<tr>
<td></td>
<td>View</td>
<td>Pin Hole HUD controlled by Ctrl+Up/Down arrow key</td>
</tr>
<tr>
<td>Process</td>
<td>Input</td>
<td>View angle at Pin Hole HUD</td>
</tr>
<tr>
<td>Data</td>
<td>Calculation</td>
<td>H = tan (angle) for scale</td>
</tr>
<tr>
<td></td>
<td>Input</td>
<td>View angle</td>
</tr>
<tr>
<td></td>
<td>Internal</td>
<td>Unit</td>
</tr>
<tr>
<td></td>
<td>Output</td>
<td>Heights for units</td>
</tr>
</tbody>
</table>

Table 3-3. Abstraction of Logger Tape virtual object.

<table>
<thead>
<tr>
<th>Category</th>
<th>Items</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>View</td>
<td>Graphic Model</td>
<td>Tape body</td>
</tr>
<tr>
<td></td>
<td>Temporary Model</td>
<td>Tape strip to target</td>
</tr>
<tr>
<td>Control</td>
<td>HUD Interface</td>
<td>Reading measurement at tape strip</td>
</tr>
<tr>
<td></td>
<td>Model</td>
<td>Activate/inactivate by Tool Bag, a HUD interface</td>
</tr>
<tr>
<td></td>
<td>View</td>
<td>Mouse picking action</td>
</tr>
<tr>
<td>Process</td>
<td>Input</td>
<td>Mouse picking location at target</td>
</tr>
<tr>
<td>Data</td>
<td>Calculation</td>
<td>Distance between target and instrument</td>
</tr>
<tr>
<td></td>
<td>Input</td>
<td>Measurement location</td>
</tr>
<tr>
<td></td>
<td>Internal</td>
<td>Unit</td>
</tr>
<tr>
<td></td>
<td>Output</td>
<td>Distances for units</td>
</tr>
</tbody>
</table>
### Table 3-4. Abstraction of Diameter Tape virtual object.

<table>
<thead>
<tr>
<th>Category</th>
<th>Items</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>View</td>
<td>Graphic Model</td>
<td>Tape body</td>
</tr>
<tr>
<td></td>
<td>Temporary Model</td>
<td>Tape strip around tree trunk</td>
</tr>
<tr>
<td></td>
<td>HUD Interface</td>
<td>Reading measurement at tape strip and height tape</td>
</tr>
<tr>
<td>Control</td>
<td>Model</td>
<td>Activate/inactivate by Tool Bag, a HUD interface</td>
</tr>
<tr>
<td></td>
<td>View</td>
<td>Mouse picking action</td>
</tr>
<tr>
<td>Process</td>
<td>Input</td>
<td>Mouse picking location at target</td>
</tr>
<tr>
<td></td>
<td>Calculation</td>
<td>Circumference of tree trunk</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Diameter with circumference</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Measurement height from ground</td>
</tr>
<tr>
<td>Data</td>
<td>Input</td>
<td>Mouse picking location</td>
</tr>
<tr>
<td></td>
<td>Internal</td>
<td>Circumference</td>
</tr>
<tr>
<td></td>
<td>Output</td>
<td>Diameter, Measuring height</td>
</tr>
</tbody>
</table>

### Table 3-5. Abstraction of Sampling Area Tool virtual object.

<table>
<thead>
<tr>
<th>Category</th>
<th>Items</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>View</td>
<td>Graphic Model</td>
<td>Center Pole</td>
</tr>
<tr>
<td></td>
<td>Temporary Model</td>
<td>Boundary tape, Tree ID tag</td>
</tr>
<tr>
<td></td>
<td>HUD Interface</td>
<td>Boundary radius entry</td>
</tr>
<tr>
<td>Control</td>
<td>Model</td>
<td>Activate/inactivate by Tool Bag, a HUD interface</td>
</tr>
<tr>
<td></td>
<td>View</td>
<td>Mouse picking action</td>
</tr>
<tr>
<td>Process</td>
<td>Input</td>
<td>Mouse picking location at ground</td>
</tr>
<tr>
<td></td>
<td>Calculation</td>
<td>Boundary tape location intersecting with ground away at area radius</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Tree ID tag location intersecting with tree within given area radius</td>
</tr>
<tr>
<td>Data</td>
<td>Input</td>
<td>Area radius, Center location</td>
</tr>
<tr>
<td></td>
<td>Internal</td>
<td>Boundary tape location, Tree ID locations</td>
</tr>
<tr>
<td></td>
<td>Output</td>
<td>Tree IDs</td>
</tr>
</tbody>
</table>
Figure 3-1. Structure of Classes which represent a forest virtual object following a proprietary method used for OpenWonderland format. Colored classes are necessary for representing a module in the VWE.

Figure 3-2. Structure of Classes which represent a HUD object for instrument (diameter) following the required proprietary method used for OpenWonderland format.
Figure 3-3. A client architecture design for data-centered representation of instrument objects. Actions of user and objects are represented as solid lines, and dash lines describes information path.

Figure 3-4. A client architecture design for model-centered representation of instrument objects. Actions of user and objects are represented as lines, and dashes line describes information path.
Figure 3-5. A use case diagram of the Virtual Learning Forest.
Figure 3-6. A sequence diagram of the virtual learning forest system, which focuses on the interaction and behaviors between actors and virtual objects.
Figure 3-7. A collaboration diagram of the virtual learning forest system.
Figure 3-8. Longleaf pine trees (from Google).
Figure 3-9. A longleaf pine tree geometry model created by SketchUp (SketchUp, 2000), a 3D modeling program from Google.
Figure 3-10. Level of detail of the longleaf pine consisting of 4 tree digital resolutions.

Figure 3-11. Major parts of the clinometer (Suunto PM-5 model).
Figure 3-12. Clinometer (Suunto PM5 model) digital content created by SketchUp (SketchUp, 2000), a 3D modeling program from Google.
Figure 3-13. Geometries of 3D clinometer digital content.

Figure 3-14. Major parts of the logger tape and detail of the ruler scale.
Figure 3-15. Logger tape (Stanley Powerlock) geometry model created by SketchUp (SketchUp, 2000), a 3D modeling program from Google.
Figure 3-16. Major parts of diameter tape (JIM-GEM Pocket Diameter Tape) and drawing showing how to read a measurement.

Figure 3-17. Diameter tape digital content created by SketchUp (SketchUp, 2000), a 3D modeling program from Google.
Figure 3-18. Materials and tools used for assigning a sampling area.
Figure 3-19. Sampling area tools (center pole, boundary tape, tree identification tag).
CHAPTER 4
AN ONTOLOGY-BASED APPROACH TO IMPROVING MODEL INTEROPERABILITY

Background

In the previous chapters, the prototype VLF was analyzed to build virtual objects that could have more flexible structure and VWE independent formats compared with traditional development methods based on proprietary products. Through further conceptualization and abstraction of the virtual system and virtual objects, distinct descriptions of virtual objects including their role and functionality can be developed. Beginning with analysis of a proprietary virtual model structures, it distinguished specification data requirements of instruments within the model framework required for specific VWEs. And, comparison of two model implementation techniques showed that the model-centered implementation of virtual object with ontology could contribute to simplify the data flow between virtual object and VWE.

Ontologies has been utilized to capture basic elements of graphic contents for specific graphic applications or graph model formats (Kalogerrakis et al., 2006; Dartigues et al., 2007; Niknam and Kemke, 2011). However, these ontologies were restricted to projecting a graphic model for a specific graphic application onto an ontology. The specific graphic application utilizing a specific graphic engine supports many run-time attributes to increase the ability of the graphic model, but most attributes are dependent to the graphic engine. Thus, the purpose of using ontology was limited to manage graphic models for sharing them for a specific application. The model interoperability is restricted in the specific domain.

A virtual instrument of the VLF is a graphic module rather than a static graphic model, which contains static parts (graphic model) and dynamic parts (behavior and
interaction). To map geometry and behavior of the module into ontology, it is necessary to consider both the graphic and behavioral/interactive aspects of the virtual objects. Interoperation of model in the VWE domain requires sharing virtual objects with different VWEs. But, there are problems of graphic model format and implementation of behavior and interaction.

In this chapter, the various objects in the VLF are analyzed from the standpoint of physical, logical and interactive/behavioral characteristics. Virtual objects' geometry ontologies are developed that represents geometry information. Model behavior ontologies are constructed to expand the functionality of the graphic model. And, it is described how the interactive virtual objects can be represented and implemented in the VWE to improve model interoperability and system architecture. System flexibility and model interoperability are tested through the results of building the same virtual world automatically on two different VWEs, OpenSimulation and OpenWonderland.

**Model Interoperability**

Generally, the term of interoperability is defined as the "ability of two or more systems or components to exchange information and to use the information that has been exchanged" in the IEEE Standard Computer Dictionary (IEEE Computer Society, 2001). It is an important aspect of application development for improving the efficiency of the software system. In the VWE domain, model interoperability can be investigated for two aspects of virtual object, a graphic behavior model and a graphic geometry model. The graphic model behavior describes behavior of the geometry model and interaction between model and user. The behaviors and interactions may be represented as a simulation or process. It requires descriptions based on the related logic and process, and an ontology to manage them independently from the VWE. A
graphic geometry model has a specific file format or description of geometry structure. An exchangeable digital content file format which has an open structure contributes to sharing them.

**Ontology-based Simulation Model Approach**

The purpose of utilizing ontologies for describing a simulation model is to create reusable knowledge that can be shared across different projects and platforms. But, the design and implementation of ontologies are different depending on the problem domain. Miller (Miller et al., 2004) created an ontology for discrete-event modeling by capturing concepts with a taxonomy of model structural characterization (e.g. State-oriented, Event-oriented) and model running mechanism. Although Miller utilized an ontology to represent stochastic models, some researchers (Jurisica et al., 2004; Cuske et al. 2005) built ontologies containing static aspects of entities such as simulation data and simulation governing rule. Beck et al. (2008) utilized an ontology to representing simulation logic and processes using objects that are editable equation representations.

Graphic model ontologies tend to focus on constructing an ontology for a specific data format and sharing models in the same community. For example, Kalogerrakis et al. (2006) created a graphic content ontologies for X3D graphic format to visualize domain knowledge. They generalized X3D graphic elements as graphic content concepts and mapped them into graphic element ontology objects such as a scene and a node. And, Dartigues et al. (2007) created ontologies for integrating computer-aided design (CAD) and computer-aided process planning (CAPP) which are commercial software applications. It focused on data exchange. A domain specific ontology is based on the features of the applications, and a shared ontology is used for mapping the domain specific ontology to other CAD-family software programs. Similarly,
Niknam and Kemke (2011) built a graphic object ontology for a Java 3D graphic scene graph data structure by capturing general graphic concepts such as shape, size and color. Java 3D has no specific file format, so concepts of graphic contents came from characteristics of the graphic model rather than the file structure. Beck (2009) created a virtual world environment which manipulates graphic models of virtual objects by using an ontology as a content database. On the other hand, Parisi et al. (2007) utilized an ontology to create an animation model for CAD and product management software (PDM). The animation model used a generated script for animation.

**Exchangeable Digital Content File Format Approach**

Interoperability of 3D digital content (3D models) has been closely related with particular software packages for creating digital content and with platforms for using the digital content. Digital content creation (DCC) software such as 3ds Max and Maya (www.autodesk.com) has been used to create digital assets for games, virtual worlds for education, social networking, and simulations. DCC software often uses proprietary file formats to code digital assets, and it requires various trans-coding tools to export and import such files to and from other DCC programs. (Earnshaw and Vince, 2001). Trans-coding tools must constantly be rewritten, as the proprietary file formats are changed. Similarly, gaming engines, which utilize the files in their platforms, require work to support diverse digital formats (Eck, 2006). Competition between different DCC software makes it difficult to adapt file format standards. Even when the format is opened source, it may be very complex to read and to develop an algorithm for using the file data.

As use of 3D digital content has expanded, the relation between content and runtime (gaming engine) has become more complex to support, especially with
advanced techniques like physical effects and animation. By separating digital content from runtime (gaming engine) physically and logically, it helps to prevent the content from being embedded in the proprietary runtime engine code and helps to enable generic APIs which can work with many kinds of data (Geroimenko and Chen, 2004; Daly and Brutzman, 2007). But, more complex dynamic behavior is required for interactive applications. More complex control systems need to combine scripted animation, physical simulation and user control. 3D content must be designed for interactivity not only for the interaction with user, but for the interactivity between the different elements of the virtual world.

An interoperable file format, COLLADA (COLLAborative Design Activity) (Arnaud and Barnes, 2006), became an industry standard at SIGGRAPH '05, which is an annual conference on computer graphics sponsored by the ACM (Association for Computing and Machinery). It provides an open standard XML schema for exchanging digital assets between various graphics software applications. COLLADA documents cover a wide range of features including geometry, material, light, effect, camera, animation, physics and controller. It supports mesh-based geometry which is good for handling large, complex dynamic relations between elements rather than other popular formats such as X3D and VRML which utilize specific application-oriented scene-graph data structures. A scene graph, a collection of spatial representation nodes in a graph (tree) structure, uses many active run-time attributes for each 3D rendering engine such as 'switch', 'level of detail', and 'script' which COLLADA format does not define. COLLADA merely defines the necessary data for enabling at any applications.
Recently, the COLLADA format was adopted by many DCC software companies and virtual world platforms such as Second Life, Open Simulator and Open Wonderland because of its interoperability. But the COLLADA format does not focus on supporting simulation applications which requires more complex interaction between elements. Simulation applications need to use intensive controls for user action by handling runtime programs. The open structure of the COLLADA format enables schema extension, and it is possible to define required information as subset functions of the COLLADA format. For example, the immersive education initiative (iEd) created the open 3D/VR file format by extending COLLADA schema for sharing digital assets with diverse virtual world platforms such as Second Life, Open Simulator, and OpenWonderland (Media Grid, 2010). However, the simulation is limited to animation, and it is insufficient to implement simulations requiring complex interactions between user and objects.

**Ontology-based Model and System**

Ontologies used for abstracting data and simulation processes are useful to define complex data and relations, as well as complex interaction and behaviors, even when behaviors result from simulation. And, many virtual objects in the agriculture and natural resources domain could be developed as interactive 3D models.

Information in the ontology is platform independent. By utilizing the ontology and generator, the geometry, interaction and behavior of the models can be converted automatically in the forms that are required for the specific VWE. And, it can be more easily shared than the proprietary, platform specific program code.

In this section, various objects will be analyzed from the point of view of physical, logical and interactive/behavioral characteristics. Virtual object ontologies will be developed that represent both object geometry and behavioral (dynamic) information
with some ontology authoring tools. A graphic geometry model generator is created to building a graphic model by using ontologies, and model behavior ontologies are utilized to create graphic model modules expanding the functionality of the graphic model. And, system flexibility and model interoperability are tested through building the same virtual world on two different VWEs with generated graphic models and model modules.

**Ontology Design**

Ontologies for digital content and module applications are developed for virtual objects including the longleaf pine tree, sampling area tool, and instruments such as diameter tape, logger tape, and clinometer. The model ontology describes geometry information of the objects, while the module application ontology is utilized for representing logical and interactive/behavioral information of the object in the virtual world.

**Tree ontology**

An ontology for representing the 3D digital content of the longleaf pine tree is shown in Fig. 4-1. It is a geometry model ontology that is utilized for generating 3D tree models. The full 3D longleaf pine model (LPTree Model) is an instance of class 3dModel, and three simplified plane models. Plane models consists of planes with a texture of front tree projection, and they are distinguished by the number of planes used for the model.

A full 3D longleaf tree model (LPTree Model) has relations with instances of 'Asset', 'Scenes' and 'Parts'. There are three instances as parts that are 'Leaves', 'Branch', and 'Trunk', and they are related with instances of material and geometries such as 'TrunkMesh'.
Geometry of the parts are represented with properties of 'pointArray', 'vertices', 'triangle', and 'mesh'. The property 'pointArray' is an ontology object containing float arrays, array id, size of array, count of XYZ coordinates. The property 'vertices' contains its id and source float array id. Similarly, the property 'triangles' includes information of triangle count, source id of vertices, and array of vertex order number. The property 'mesh' has id, float array ids, vertices ids, and triangles ids.

'Plane 3D' models have similar structures of ontology objects representing their geometry. The 'TreePlane' is a class representing a part of model geometry, and the 3D plane models, 'LPTree1P Model', 'LPTree2P Model', and 'LPTree3P Model', have different number of plane of geometry to constructing 3D tree. For example, the 'LPTree3P Model' has 3 'TreePlane' objects. The 3D tree models are a set of tree models representing different levels of detail of the tree based on the distance from user avatar. They are utilized for applying discrete levels of detail (LOD) in the virtual world.

A model behavior ontology (Fig. 4-2) of the longleaf pine is an application ontology that is utilized for forming a module to describe relations with other models in the virtual world. It describes 1) how to manipulate (LPTree MovementOperation) the LOD of tree according to the distance between tree and user avatar, and 2) how to calculate the value of the 'DistanceFromAvatar' object which represents the distance with values relative to other object in the virtual world. An instance 'LPTree MovementOperation' of 'MovementOperation' class is utilized to broadcast tree location and the changed avatar location. The 'MovementOperation' class has properties of source object name, target object name, and their coordinates. The values of properties are used to calculate a distance between them.
An instance of class Equation, 'DistanceFromAvatar', expresses a equation to calculate the distance with property value of the 'LPTree MovementOperation'. The equation is represented with operators (Root, Square, Minus) and operands (TreeLocation, AvatarLocation).

**Sampling area tool**

A sampling area tool model ontology is shown in Fig. 4-3. The sampling area tool is a set of three tools, center pole, boundary tape and tree number tag. A center pole is a 3D geometry model which has a stick shape with 2m length and a 5CM x 5CM rectangular intersection. Its assets, scene and parts of geometry are represented in the instance 'CenterPole Model'.

A boundary tape and tree id tag are objects constructed from temporary geometry information that are passed through from other object's activity such as collision with the boundary of the sampling area. For example, the boundary tape may have a different diameter, and it is placed on the ground that has an irregular surface height due to irregularities in the surface terrain. And, tree id tags must be allocated to different numbers of trees according to the sampling area.

An instance 'BoundaryTape' and an instance 'TreeNumberTag' are instantiated from class '2DModel'. The instance 'BoundaryTape' contains instances of 'LineSeg' representing segments of lines composing the lines of the boundary tape. An instance of 'TreeNumberTag' contains instance 'Tag' of 'Box' shape and instance 'TreeNumber' of 'Text'.

A sampling area tool behavior ontology is presented in Fig. 4-4. It describes how the geometries of the three tools (center pole, boundary tape, and tree number tag) cooperate each other.
An instance 'CenterPole Model' is located by information given from instance 'SATool EquipmentOperation' by detecting a user's mouse picking point. It is the center location of the sampling area, and it is utilized to seek points on the ground for generating the boundary tape. An instance 'BTGenerate' contains an equation of an instance 'SeekGroundCollisionPoints', and the equation gives points that place the boundary distance calculated with variable 'SamplingAreaRadius' and variable 'CenterLocation'.

Similarly, an instance 'TagGenerate' utilizes an equation of instance 'SeekTreeArea' for figure tree locations inside of the given sampling area. The equation is formed similar to the equation of the instance 'SeekGroundCollisionPoints', but it facilitates the 'LessThan' operator instead of the 'Equal' operator to figure that the tree locations are located in the region within the boundary points.

Diameter tape

A diameter tape model geometry ontology is presented in Fig. 4-5. The geometries of the diameter tape during diameter measuring activity are represented with a 3D model of the diameter tape and a 2D model of the enlarged tape part for reading measurement on the tape to make it easier for users to read values.

An instance 'DiameterTape Model' of class '3DModel' represents a 3D model with information about asset, scenes and parts. There are three geometry parts, instances of 'RetractButton', 'BoxFrame' and 'StripTape'. And, an instance 'DiameterTape HUD' of 2DModel class represents tapes and rulers on tape with 2D geometries such as boxes and lines. Its properties, 'width' and 'height', contain integer values for HUD window size. It has associations of 'Contains' with 2D geometry objects, 'UpTapeStrip', 'DownTapeStrip', 'Crosshair', and 'UDSacle'. Each object has geometry
element such as 'Point' object that represents X-Y coordinate. For example, 'UpTapeStrip' showing upper diameter tape strip has two 'Point' objects, 'USUpLeft' and 'USDownRight', that contains X-Y coordinates defining left-upside corner and right-downside corner of the tape strip. An Instance 'UDScale' describes an upper tape whose scale starts from 0 and lower tape including scales for measuring values. The scale of the lower tape changes dynamically during measuring activity.

In Fig. 4-6, a diameter tape model behavior ontology is presented. An instance 'DiameterTape Model' of class '3DModel' utilizes 'DiameterTape MovementOperation' which is an instance of class 'MovementOperation' to obtain a location of the 3D diameter tape model that is assumed to be attached to user avatar. The class 'MovementOperation' contains properties such as a 'resource' (e.g. a String object of object name to check the location) and 'location' (e.g. a String object of float values describing X-Y-Z coordinate). An instance 'DTape' detects a location on the tree for measuring diameter using information from instance 'DiameterTape EquipmentOperation' of class 'EquipmentOperation'. The class 'EquipmentOperation' is utilized for representing data created by using the instrument (e.g. mouse-picking position at tree trunk for measuring tree diameter), and providing the data to related objects (e.g. an equation object 'DiameterOfCircumference'). It contains properties which include 'resource' (e.g. a String object of action resource name, 'mouse'), 'location' (e.g. a String object of float values describing X-Y-Z coordinate. The data of 'EquipmentOperation' is referred by an equation object. For example, an equation object, 'DiameterOfCircumference', utilizes 'Circumference' object to calculate a diameter.
In this process, circumference at the target point is obtained by integrating the length of lines that consists of points on the trunk intersecting with an ideal plane generated at the measuring location.

An instance 'DiameterTape HUD' utilizes an equation of instance 'UDScaler'. 'DiameterTape HUD' object has an association relation, 'Update', with the 'UDScaler'. The 'UDScaler' object have an instance of equation, 'DiameterOfCircumference'. The equation calculates diameter of the tree cross-section with variables of 'Circumference' and 'PI ' (3.14). The scales and calculated diameter are represented in the 2D model.

**Logger tape**

A logger tape model geometry ontology is shown in Fig. 4-7. There is a 3D model (LoggerTape Model) and a 2D model (LoggerTape HUD). The geometry of the logger tape during distance measuring activity are represented with a 3D model of the logger tape and a 2D model of the enlarged tape part for reading measurements on the tape at a tiny area in the virtual world.

An instance 'LoggerTape Model' of class 3DModel contains information on asset, scenes and parts. There are three geometry parts, instances of 'RetractButton', 'BoxFrame' and 'StripTape'. And, an instance 'LoggerTape HUD' of '2DModel' class represents tapes and rulers on the tape with 2D geometries such as boxes and lines. An instance 'LRScale' is used for describing scales on the left and right side of the tape (different scales), and both scales of the tape change dynamically during the measuring activity.

In Fig. 4-8 a logger tape model behavior ontology is presented. An instance 'LoggerTape Model' of '3DModel' class obtain the location of the 3D logger tape model that is assumed to be attached to user avatar. An instance 'LTape' detects a location
picked by a mouse-clicking action from information from instance 'LoggerTape EquipmentOperation'.

An instance 'LoggerTap HUD' utilizes an equation of instance 'LRScaler', and the equation calculates a distance from target location from the avatar. The scales and calculated distance are represented in the 2D model.

**Clinometer**

A taxonomy of clinometers entities (Fig.4-9) is designed to represent and store physical and logical information which characterizes a clinometer for realization in the virtual 3D world. It includes information on physical shape (class 3DModel), behavior (class Operation), user interface (class DisplayInterface), and ways to process interaction and behavior (class Process).

Class 3DModel has four subclasses representing physical parts of a clinometer, which are 'BoxBody', 'Pinhole', 'SideScale' and 'ConversionTable'. They are sets of geometric groups forming the geometry of the clinometer. It makes it simple to apply changes in appearance of the 3D model such as changing texture, shape type, and background image. Class 'Operation' represents ways of controlling the instrument. Subclass 'EquipmentProcess' describes how to handle the instrument to get measured values by using it, and has properties specifying action invokers (e.g. mouse wheel up/down for changing clinometer's view angle) and event processers (e.g. update view angle value).

Subclass 'MovementProcess' explains how to change the position of the instrument in some direction (e.g. moving clinometer towards or away from the tree), and describes whether it moves by itself or under control of the user avatar while adjusting the instrument.
Class ‘DisplayInterface’ defines the types of interfaces for displaying information such as enlarged the measuring scale bar and displaying tips for help. Subclass ‘HUD’ is an interface of heads-up-display, and subclass ‘Popup’ is a dialog window. Subclasses of ‘HUD’ class, ‘PinholeScale’ and ‘SideScale’, describe the scale displays for pin hole in the bar-shape and for the side-scale angle.

Class ‘Process’ is a subclass representing the mathematical and functional behaviors of the clinometer and user interface. Subclass Equation describes mathematical expressions of processes. For example, class ‘AngleToHeight’ is an equation calculating tree height based on the measured sight angle, and is used to show the tree height in the clinometer pinhole.

More detailed ontologies for the clinometer are developed based on the above taxonomy. They describe how the clinometer 3D graphic object is represented and how it can be manipulated by operation information.

An ontology for representing a 2D or 3D geometry of the clinometer is shown in Fig. 4-10, and the model geometry ontology is utilized to generate a 3D or 2D model. The 2D model contains several parts including boxes, circles and lines for representing the clinometer box, optical pin-hole, measurement crosshair, and scale. Left and right scales are a set of scale lines iterated with scale type (e.g. degree or percent) and the intervals between lines are based on scale origin location and scale size.

An instance of class ‘3DModel’ represents geometries of the clinometer in 3D, model boundary, and supplement information (e.g. instance Asset) such as creator and unit of geometry. A 3D model includes points, vertices, triangles and mesh geometries for representing parts of the clinometer. A mesh is consists of sets of triangles.
constructed by vertices which are identified from a point array. An instance Scenes defines the boundary of the model, and represents an aggregation map of geometry groups or components that are defined as separated sets of objects.

The application ontology (Fig. 4-11) describes 1) how to update (ClinometerEquipmentOperation) display the scale based on changes of user action (Mousewheelaction), and 2) how to calculate (AngleToHeight) a value (TreeHeight (H)) from result of action change (ex: ViewAngle (alpha)) with distance (Distance) from tree.

And, it also explains 3) how to manipulate (ClinometerMovementOperation) the model according to change of location (AvatarLocation). The ontology is utilized to develop a clinometer model module.

An instance 'AngleToHeight' of class 'Equation' is a mathematical expression constructed with left operand, equal (=) operator and right operand. The equal operator is considered as an assignment operator, not a relation operator. The right operand consists of a 'Multiplier' operator, variable 'Distance' and operator 'Tangent' of variable 'ViewAngle'. The 'TreeHeight' variable is updated from the equation 'AngleToHeight'.

An instance of class 'ClinometerApp', PM5, has an instance of class 'MovementOperation', 'ClinometerMovementOperation, as a properties for defining the behavior of the graphic model according to changes of position (AvatarLocation). The operation manipulates the location of the graphic model in the virtual scene by detecting the location of the avatar wearing the instrument.

**Ontology Implementation**

Every concept involved with an instrument’s geometry, behavior and interaction is formally defined by an object in the ontology. An ontology of an instrument contains concepts such as graphic model, model behavior, process and other concepts specific
to the instrument. A concept contains taxonomic relationships (e.g. a '3DCl
ometerModel' is a member of the class 'Clinometer'), properties (e.g. a 3D model has a particular shape), and association with other concepts (e.g. a clinometer can contain a 2D model, a clinometer contains equipment and/or movement operation for the instrument).

**Authoring tools**

LyraBrowser (Beck, 2008) and ObjectEditor (Beck, 2007) are graphic user interface tools for authoring ontology objects using the engine of the Lyra OMS (ontology management system) (Beck, 2008). Lyra OMS is a server/client environment based on an object database management system for ontology objects. The online editors provide effective ways to collaborate with multiple authors over remote distances. LyraBrowser and ObjectEditor are low-level authoring tools for visualizing and manipulating the ontology as a node-and-link style graph diagram. LyraBrowser is modified with enhanced navigating and manipulating functionalities for manipulating ontology elements by displaying the ontology elements in the shape of a network.

SimulationEditor and EquationEditor (Beck et al, 2008) are higher-level authoring tools using Lyra OMS environment. SimulationEditor is a tool to create a structure diagram of concepts for building a ontology-based simulation of a specific physical system (Kwon et al, 2010b). EquationEditor is a tool used in conjunction with SimulationEditor for specifying the dynamic behavior of a physical system through mathematical equations. SimulationEditor and EquationEditor are used for authoring equation-type processes describing behaviors of model (e.g. calculation of distance by
moving instruments) or relating parts of a model (e.g. calculation of tree height with view angle from scale of the clinometer).

**Building an ontology**

The VLF instrument ontologies which were described in the previous section are implemented with the LyraBrowser. Class 'VLF' is a superclass of instruments, and the instruments have associations with graphic model classes and movement/equipment operation classes. The graphic model concepts represent 2D and 3D graphic model. The operation concepts relate with model behavior such as 'MovementOperation' (it describes the behavior of a graphic model as effected by user (avatar) movement) and 'EquipmentOperation' (describes concepts related with measurement actions with an instrument).

'MovementOperation' concepts are used for describing the position of the instrument which is usually attached to the avatar in the virtual world. It has a property for reference object name. For example, clinometer and distance tape are assumed to be attached to an avatar when activated, so the reference object indicates the avatar. While, 'EquipmentOperation' concept associated with an equation object which is used for calculating measurements of an instrument.

Fig. 4-12 shows the relationship between concepts of instruments in a graph and data table. The graph shows that 'VLF' class has instances of instrument such as 'Clinometer', 'Diameter Tape', and 'Logger Tape'. They are listed in the box of all instances, and one of the instances (Clinometer) is highlighted at top of the list box. 'VLF' class has instrument model classes as subclasses. For example, 'Clinometer Model' class is highlighted at top of the all subclasses box. The arrow between 'VLF' class and the list box describes the relationship between them. The 'Clinometer Model'
class has 4 associations with 2 model class ('VLF DC Clinometer 2D Model', 'VLF DC Clinometer 3D Model'), movement operation ('VLF Clinometer MO') and equipment operation ('VLF Clinometer EO'). In the left property window of 'Clinometer Model' class, names of associations and related concepts are represented.

The 'VLF Digital Content' class is created to represent concepts for a graphic model (Fig. 4-13). It has two subclasses which are 'VLF Digital Content 2D Model' class and 'VLF Digital Content 3D Model' class. 'VLF DC Clinometer 3D Model' class, a subclass of 'VLF Digital Content 3D Model', is a concept representing graphic elements of the clinometer model. It defines a graphic model with elements of asset (model description), scene (scene information of the model) and shape (model geometry information) (Fig.4-14). The shape contains parts of geometry with the part name of the model. For example, 'VLF Clinometer Shape' represents its geometry as groups of box frame (VLF Clinometer Part BoxFrame class), conversion table (VLF Clinometer Part ConversionTable), pin hole (VLF Clinometer Part Pinhole), and scale wheel (VLF Clinometer Part ScaleWheel).

In Fig.4-15, for example, the graph is shown of the relationship between 'VLF Clinometer Shape' class and its subclasses. Each part has associations with geometric elements such as mesh, point array, triangle, and vertices. The mesh concept has properties such as ID, IDs of triangle which consists of the mesh geometry. 'Triangle' concept contains the used vertices group IDs and order of vertices. The order of vertices is an array of integer that represents a vertex id. 'Vertices' concepts contains an id of 'Point array' concept. The 'Point array' concept has an array of float values which is
a member of X-Y-Z 3D coordinate. And, it contains a count of point (total number of coordinate sets) which is used to figure out an id of point from the array.

SimulationEditor is utilized to build an ontology of VLF instrument behavior. The ontology contains a high-level of concepts for each instrument which has processes in the form of equations.

In Fig.4-16, four concepts for each instrument (clinometer, logger tape, diameter tape, and sampling tool) are created and shown in SimulationEditor. They are defined as subclasses of SimulationEditor diagram class, and each class is associated with equation objects.

EquationEditor facilitates building and managing equation objects within a graphic user interface. An equation object is created by assembling operators and operands. Representing an equation expression starts place the Equal (‘=’) operator as a root element which allows two operands at left/right sides. Assembled operators and operands construct a hierarchical structure. A left-side operand of the Equal operator is restricted to symbols (concepts), while any number or symbols are allowed to be utilized in operators. The equation is decomposed in a hierarchical order, and each node (operator and symbol) in the decomposition becomes an ontology object with relationships to other operators and/or symbols.

In the ontology, EquationEditor element class has two subclasses, EquationEditor symbol and EquationEditor equation that contain symbol and/or equation objects. In Fig. 4-17, EquationEditor element class is shown at LyraBrowser. At left window, two subclasses of EquationEditor element class, EquationEditor equation and EquationEditor symbol, are listed as a subclass of Equation element class.
In the middle window, there are subclasses of EquationEditor equation class which is selected in the left window. The VLF Instrument equation class in the subclass list is created and utilized for the VLF project, and on the right window it is shown the related information of the class. An equation object is defined as a subclass of the class. The EquationEditor symbol class has a similar subclass relationship with symbol objects.

In Fig. 4-18, it shows an equation representing a formula to calculate tree height with given symbols of view angle in percentage (e.g. View Angle Percentage) and distance (e.g. Distance) within the Equation tab of the EquationEditor. The equation object has a unique equation ID, 'EQUATION Tree Height from Percentage'. The equation is created in a hierarchical order. The equal operator ('=') is placed, which has two blank slots at left/right side. The symbol (e.g. Tree Height) representing the equation result is set by selecting it among the symbol list, and on the right side blank slot the multiply operator ('X') is assigned. It also has two blank slots at left/right side. On the right side, the symbol, 'Distance' is placed, and at left side blank the divide operator ('/') is set. The denominator of the operator is '100', and the numerator is the symbol, 'View Angle Percentage'. The operators and symbols are stored as objects in the ontology.

A symbol object represent a concept containing a value. It has properties such as Symbol ID, Symbol sign, definition, and unit. Symbols objects are created and managed in the Symbols tab of the EquationEditor (Fig. 4-19). Initial value of the symbol is defined by specifying a source type in the option menu. For example, selecting the Equation option of source type means that the symbol is automatically associated with an equation which uses the symbol at the left-side operand of Equal operator.
Digital Content Generator

A graphic model geometry generator and a graphic model module generator are developed, as programs written in Java. They retrieve information about graphic model and graphic model module from model geometry ontology and model behavior ontology through Lyra.

Graphic model geometry generator

A graphic model geometry generator is an application written in Java that generates a graphic model by using a graphic model geometry ontology. Currently, it generates a graphic model in the COLLADA format that is acceptable in the OpenWonderland and OpenSimulator, though generators could be written for other formats. The COLLADA graphic format of digital content is widely used for representing geometry, relations between geometric elements (e.g. joint), and automated action/behavior of geometry (e.g. wired bone model). Although it has gained much interests as a graphic format because of its flexible and expandable format based on xml tags, it has limited ability to describe interaction and complex behaviors within and between objects.

The generator includes a process to interpret the content of the ontology, and to create a COLLADA format graphic file by manipulating values of the graphic model with data in the ontology. By modifying a property value of the ontology it is possible to generate various models. The COLLADA file format follows the COLLADA specification version 1.4.1 defining 15 library elements to describe graphic content. Elements are animation, animation_clip, camera, controller, geometry, effect, force_field, image, light, material, node, physics_material, physics_model, physics_scene, and visual_scene.
An example of graphic model generation is presented with a clinometer model. Fig. 4-20 shows an example of a geometry model which is stored in the ontology, and it is a scale wheel part of clinometer model. It has a cylindrical shape and contains image textures for ruler.

The geometry model of clinometer was generated with data in the instrument geometry ontology. For example, the Fig. 4-20 showed instances which contain geometry information of clinometer scale wheel. There are 4 instances of mesh, triangle, vertices, and point array. The geometry structure is based on mesh constructed by triangles. A triangle is represented by vertices which is a XYZ coordinates. A point array contains float values which indicate one value of XYZ coordinates.

At Fig. 4-20 (a), instances of 'VLF 3D Mesh' are displayed, which are used for the scale wheel part. The instance is named with the identification number of the mesh after prefix 'VLF M ID'. It has properties of 'VLF MSID' (id number of the mesh), 'VLF P SRC IDS' (id numbers of point array instances referred by the mesh instance), 'VLF V SRC IDS' (id numbers of vertices instances referred by the mesh instance), and 'VLF T SRC IDS' (id numbers of triangle instances referred by the mesh instance). For example, a mesh instance, 'VLF M ID8' has an id number (8), and refers instances of point array (id 14, 15, and 17), an instance of vertices (id 16), and instances of triangle (id 1 and 2). It means that the mesh geometry consists of two triangles which use a vertices constructed with 3 point arrays.

Fig. 4-20 (b) shows a list of triangle instances. The instance is named with the id number of the triangle after prefix 'VLF T ID'. It has properties of 'VLF TID' (id number of the triangle), 'VLF TCOUNT' (count of triangles), 'VLF MID' (id numbers of material),
'VLF VERTEXID' (id number of vertices instance), 'VLF TEXCOORDID' (id numbers of texture coordinate), and 'VLF TPIDS (coordinate order numbers for values in a point array instance). For example, a triangle instance, 'VLF T ID1' has an id number (1), which is built with 48 triangles and a material (Material 13). The triangles are represented with coordinates of vertices (ID16), and the order of vertices are shown in order number array (...3 3 1 1 3 4 4 4, ..).

A list of vertices instances are shown at Fig. 4-20 (c). The instance is named with the id number of the vertices after prefix 'VLF V ID'. It has properties of 'VLF VID' (id of the vertices), 'VLF PSRCID' (id of point array instance representing topology), and 'VLF NSRCID' (id of point array instance representing normal vector). For example, a vertices instance, 'VLF V ID16' has an id (ID16), and its topology is represented with a point array instance (ID14). The normal vector is created with a point array instance (ID15).

At last, a list of point array instances are shown at Fig. 4-20 (d). The instance is named with the id number of the point array after prefix 'VLF P ID'. It has properties of 'VLF SID' (id of the point array source), 'VLF FAID' (id of float values array containing values of coordinates), 'VLF FLOATCOUNT' (count of float values in the float array), 'VLF FLOATARRAY' (array of float values), 'VLF STRIDE' (count of coordinate elements, e.g. 3 for XYZ system), and 'VLF SCOUNT' (count of coordinates which are parsed with the given stride). For example, a point array instance, 'VLF P ID14' has an id (ID14), It contains a float array source (ID23) which has 300 float values (...0.3773541459283267 0 0.0842365764077099 0.4169600892469052,..). They are parsed as a coordinate of XYZ system (stride-3), and there are 100 coordinates.
During the generation, a empty COLLADA format file is created. The, each mesh object of the part in the ontology is collected, and transformed as a 'mesh' node within a 'geometry' node under 'library_geometries' node. Under the ‘mesh’ nodes, data in 'pointArray' object are formed as a source node with id, length of array, and float array. And, data in 'vertices' ontology object are generated into the 'vertices' node under 'mesh' node with its id and source id. Similarly, data in 'triangles' object are formed in the 'triangles' node with order array of vertices consisting the triangles. The result COLLADA file is shown in the Fig. 4-21.

**Graphic model module generator**

The graphic model module, a model package set of digital graphics and program code, is used to enrich functionality of static graphic elements. The program code is required to be written in a specific programming language dictated by the target VWE.

A graphic model module generator was developed in Java, and is specific to the particular target platform. Thus there is a generator for OpenWonderland and a slightly different one for OpenSimulator. The generator creates code from the model application ontology, and the code contains processes that control behaviors of the geometry in responds to user action. And, it includes implicit processes of the model for external features which are not included in the digital content (e.g. a HUD component for measurement indicator of the clinometer).

The code is automatically generated by following the API of a graphic library which is embedded in the virtual environment system. For example, a model module written in Java is utilized for the OpenWonderland VWE, while the OpenSimulator VWE uses modules written in C++ and Linden Script Language (LSL) for Second Life.
A graphic model module is generated for the OpenWonderland VWE. It is a zip file containing a clinometer graphic model generated with the clinometer model geometry ontology and program codes created with the model behavior ontology. The clinometer graphic model is generated in COLLADA.

The generator retrieves model information for the user interface (Class DisplayInterface), geometry (Class 3DModel), behavior (Class Operation) and processes (Class Process) without any restriction from the graphic format of model itself. And, it generates program code matching with the VWE platforms for the user interface (a HUD scale indicator), operation (changing view angle of clinometer by user’s mouse action) and processes (calculating view angle and values of scale ruler) automatically (Fig. 4-22). The model generator creates digital content for the 3D clinometer model in a COLLADA format.

By utilizing the ontology and generator, the geometry, interaction and behavior of the models can be converted automatically in the forms that are required for the specific VWE. The information in the ontology is platform independent, and it can be more easily shared than the proprietary, platform specific program code.

**Model Interoperability**

Many VWEs are developed with proprietary programming language, and different graphic engines are used for different languages to provide the graphic rendering of the system. The proprietary nature of these languages, as well as lack of transparency at the code level, limits the usability of digital content and programming code implementing behavior and process, and limits ability to move models between different VWEs.

For example, the OpenWonderland uses the jMonkeyEngine (http://jmonkeyengine.com) and provides a COLLADA graphic file loader to load the file
format instantly. And, a model module needs to contain program code written in a specific programming language, Java. While, the Second Life (http://secondlife.com) developed a graphic engine with OpenGL (http://www.opengl.org) API, and its specific graphic model is provided to build content in the world. Recently, a COLLADA graphic format model is also allowed to be used in the Second Life. Linden Script language (LSL) is used to control the behavior of objects.

COLLADA graphic format is supported for generating a graphic model, and Java code and LSL code are provided for the OpenWonderland and Second Life (similar VWE such as the OpenSimulator).

Model interoperability was tested by generating and applying a clinometer model and forest which can be used in two different VWEs, the OpenWonderland and the OpenSimulator. The OpenSimulator (http://opensimulator.org) is a virtual world environment based on the reverse-engineered the Second Life server (http://secondlife.com). As an open source multi-platform, multi-user 3D application server, it received much attention because of its compatibility with the client for Second Life, and it has a large user community.

A model geometry generator created a tree and clinometer 3D graphic model in the COLLADA format. Module generators are utilized for generating program code for a forest, tree and clinometer modules for implementing behavior and processes of the models. For the OpenWonderland, the module contains processes for loading a graphic model into the VWE repository, scaling and positioning of the model, and linking between parts of the model. The module is generated in the Java language, compiled and packaged in a jar file . For the OpenSimulator, LSL scripts are generated, and they
are associated with the graphic model manually and utilized in the system without any compiling procedure (scripts are interpreted).

In the module, a function is reserved for detecting interaction with the user interface. And, there is a function to implement a dynamic scale indicator representing measured values converted by calculation functions which are generated from the equations in the behavior ontology. The equation generation is facilitated by a generator application utilizing an equation ontology and object in Lyra (Beck, 2009).

In Fig. 4-23 and 4-24, it is shown that a forest of longleaf pines and a clinometer model are placed and activated on two different virtual world servers, the OpenSimulator VWE and OpenWonderland VWE. The worlds are viewed through the Second Life client program and OpenWonderland embedded viewer, respectively.

The generated 3D models of longleaf pine and clinometer are uploaded to the VWE, and a forest module application is utilized to populate tree objects in the world. A forest of 100 longleaf pines is created on an small island at the OpenSimulator server (Fig.4-23-(a)), and same number of trees are generated in the OpenWonderland server (Fig. 4.23-(b)).

The generated tree modules for both VWE populate a forest of longleaf pines with given population model information. But, terrains used in the test are directly adopted with a simple model created with the VWE's digital content creation tool (for the OpenSimulator) and a simple plain model (for the OpenWonderland).

The generated clinometer model is deployed in both VWEs is shown in Fig. 4.24- (a) (at the OpenSimulator) and Fig. 4.24-(b) (at the OpenWonderland). Behavior (changing view angle) and processes (calculate tree height from angle) of the
climometers are generated in the LSL code and Java code for the OpenSimulator and the OpenWonderland VWE respectively. As a result, it was possible to interoperate 3D models with their behavior and the virtual world in other VWEs and avoid problems with the proprietary languages used on each specific VWE.

**Summary**

This chapter explores ways to improve model interoperability by using an ontology-based approach. The model interoperability in the VWE domain means that a graphic model or graphic module containing processes such as behaviors and interactions can be reused at different VWEs. Ontology may contribute to manage domain knowledge of model behavior and interaction, and geometry of model within forms independent to graphic format and VWEs.

Based on results, in the previous chapter, of virtual objects in the VLF created with traditional proprietary method, ontologies for various virtual objects are developed with the point of view of physical, logical and interactive/behavioral characteristics. The virtual objects includes tree, logger tape, diameter tape, sampling area tool, clinometer. To describe the characteristics of a virtual model, a model geometry ontology and a model behavior ontology are created for each objects. Geometry data in the ontology is represented with elements of mesh-based geometry such vertices, triangle and mesh. A behavior ontology contains concepts of interaction between user and object, and processes (simulation) as a collection of equation objects.

Two generators are developed for creating a graphic geometry file and a graphic module containing graphic model and program code for processes. The graphic geometry generator retrieves geometry data from the ontology, and generates a graphic file for the target VWE. COLLADA file formation is supported currently. The module
generator creates a module of graphic model and program code, which expand the ability to handling the graphic model and controlling related processes implicitly.

System flexibility and model interoperability are tested by building the same virtual world in two different VWEs with generated graphic models and model modules. OpenWonderland and OpenSimulator are the testing VWEs, and the worlds contains virtual tree and virtual clinometer. Graphic geometry models of tree and clinometer are generated in COLLADA file. Modules of tree and clinometer are generated for OpenWonderland, while for OpenSimulator LSL codes are created. It was possible to interoperate 3D models with their behavior and the virtual world in other VWEs, and avoid problems with the proprietary languages used on each specific VWE.

Information in the ontology is platform independent. By utilizing the ontology and generator, the geometry, interaction and behavior of the models can be converted automatically in the forms that are required for the specific VWE. And, it can be more easily shared than the proprietary, platform specific program code.
Figure 4-1. A diagram of the longleaf pine model geometry ontology. Gray solid-lined rectangles are classes, and orange dash-lined rectangles are instances. Blue circles are properties. Arrows show relations.
Figure 4-2. A diagram of the longleaf pine model behavior ontology. Gray solid-lined rectangles are classes, and orange dash-lined rectangles are instances. Blue circles are properties. Arrows show relations.
Figure 4-3. A diagram of sampling area tool model geometry ontology. Gray solid-lined rectangles are classes, and orange dash-lined rectangles are instances. Blue circles are properties. Arrows show relations.
Figure 4-4. A diagram of sampling area tool model behavior ontology. Gray solid-lined rectangles represent classes, and orange dash-lined rectangles mean instances. Blue circles represent properties. Arrows show relations.
Figure 4-5. A diagram of diameter tape model geometry ontology. Gray solid-lined rectangles are classes, and orange dash-lined rectangles are instances. Blue circles are properties. Arrows show relations.
Figure 4-6. A diagram of diameter tape model behavior ontology. Gray solid-lined rectangles are classes, and orange dash-lined rectangles are instances. Blue circles are properties. Arrows show relations.
Figure 4-7. A diagram of logger tape model geometry ontology. Gray solid-lined rectangles are classes, and orange dash-lined rectangles are instances. Blue circles are properties. Arrows show relations.
Figure 4-8. A diagram of logger tape model behavior ontology. Gray solid-lined rectangles represent classes, and orange dash-lined rectangles mean instances. Blue circles represent properties. Arrows show relations.
Figure 4-9. Taxonomy of the virtual object (clinometer) domain ontology.
Figure 4-10. A diagram of clinometer model geometry ontology. Gray solid-lined rectangles are classes, and orange dash-lined rectangles are instances. Blue circles are properties. Arrows show relations.
Figure 4-11. A diagram of clinometer model behavior ontology. Gray solid-lined rectangles represent classes, and orange dash-lined rectangles means instances. Blue circles represent properties. Arrows show relations.
Figure 4-12. An implementation example of an instrument (clinometer) ontology within LyraBrowser.
Figure 4-13. VLF Digital Content class and its subclasses.

Figure 4-14. Taxonomical hierarchy of VLF DC Clinometer 3D Model class.
Figure 4-15. Subclasses and associations of VLF Clinometer Shape class.
Figure 4-16. Example of building a group of model behaviors for VLF instruments with the SimulationEditor, an ontology-based simulation authoring tool.

Figure 4-17. Representation of Equation element class and their subclasses in the LyraBrowser. VLF Instrument Equation class is defined as a subclass of EquationEditor equation class which is subclass of EquationEditor element.
Figure 4-18. An equation object for representing the tree height calculation with view angle in percentage and distance, in equation tab of EquationEditor.

Figure 4-19. An example of defining a symbol (SYMBOL Tree Height) within Symbols tab in the EquationEditor.
Figure 4-20. An example of geometry instances in ontology representing a scale wheel part of an instrument (clinometer) within LyraBrowser.
Figure 4-21. An example of generated COLLADA code for a scale wheel part of clinometer model.
Figure 4.22. An example diagram of Clinometer model module generated from the ontologies.
Figure 4-23. A simple example of 3D model interoperability. Longleaf pine forest model is loaded in virtual world servers: (a) OpenSimulator and (b) OpenWonderland.
Figure 4-24. A simple example of 3D model interoperability. Clinometer model is loaded in virtual world servers: (a) OpenSimulator and (b) OpenWonderland.
CHAPTER 5
CONCLUSIONS, CONTRIBUTIONS, AND FUTURE DIRECTIONS

Conclusions

Presenting a Virtual World in the Agriculture and Natural Resource Domain

The presented work shows that the virtual world environment can be used for educational experiments and simulation. A virtual learning forest (VLF) was developed by using virtual world technology to enhance education experiences without restrictions on time, distance, cost or safety. VLF consists of three scene components (tree, terrain, and forest) and virtual instruments for mensuration including a diameter tape, logger tape and clinometer. Evaluations were conducted including an expert test, performance test, usability test for system design and learning efficiency. The Usability evaluation gave reasonable acceptance and satisfaction for the system, and the result of learning efficiency test showed that students improved their skills by reducing measurement errors.

For forestry education, the VLF shows much potential for offering immersive learning experience that improve on traditional computer-based training. It can offer a laboratory experience that complements real laboratories which may be expensive, dangerous, or inaccessible to students. The developed of a new forestry educational delivery paradigm, such as the VLF, can overcome limitations on cost, time and location.

Abstracting and Characterizing Virtual Objects

The prototype VLF system was analyzed. The roles of objects in the system are captured, and further detailed features and functionality of objects are obtained by creating modeling diagrams such as use case diagram, sequence diagram, and collaboration diagram. Virtual objects such as longleaf pine tree, logger tape, diameter
tape, clinometer, and sampling area tool are analyzed by comparing with real object to abstract feature, functionality, relationships, and data. Analysis of proprietary model structures gave ways to identify properties related with proprietary model structure and to distinguish the independent properties among others. Two model representations, data-centered and model-centered, are compared to represent the data flow structure and virtual object design. A model-centered representation can simplify the data flow, and capture complicated data structure with ontology.

The abstracting and characterizing process helped to identify the role of elements in the model. The identification information leads to developing flexible and extendible models that incorporate complicated relationships between graphic models, behaviors and processes.

**Methodology for Enhancing Model Interoperability by Using An Ontology**

The presented work shows that an ontology-based approach improves model interoperability. The model interoperability in the VWE domain means that a graphic model or graphic module containing processes such as behaviors and interactions can be reused at different VWEs. Ontology contribute to manage domain knowledge of model behavior and interaction, and geometry of model within forms independent of graphic format and VWEs.

Ontologies for various virtual objects are developed with the point of view of physical, logical and interactive/behavioral characteristics. To describe the characteristics of virtual model, a model geometry ontology and a model behavior ontology are created for each objects. Geometry data in the ontology is represented with elements of mesh-based geometry such vertices, triangle and mesh. A behavior ontology contains concepts of interaction between user and object, and processes
(simulation) as a collection of equation objects. Two generators are developed for creating graphic geometry file and a graphic module containing graphic model and program code for processes.

And, system flexibility and model interoperability are tested by building the virtual world on two different VWEs with generated graphic models and model modules. It was possible to interoperate 3D models with their behavior and the virtual world in other VWEs, and avoid problems with the proprietary languages used on each specific VWE.

Information in the ontology is platform independent. By utilizing the ontology and generator, the geometry, interaction and behavior of the models can be converted automatically in the forms that are required for the specific VWE. And, it can be more easily shared than the proprietary, platform specific program code.

Contributions

- A virtual learning forest system for forestry mensuration exercises was built. It can offer experience that complement real laboratories which may be expensive, dangerous, or inaccessible to students. The new forestry educational delivery paradigm, the VLF, can overcome limitations on cost, time and region

- Virtual objects in the domain were analyzed to abstract and categorize model objects. Environment objects and instrument object can be reused in other virtual world

- An ontology for virtual instruments was developed. It is independent of VWE, and it can be used in automatically generating portable and reusable virtual instruments.

Future Directions

Expansion of Model Interoperability

This study tested two VWEs, OpenWonderland and OpenSimulator, which are two popular open VWEs. Model generation supports only limited graphic format and
program code. There are many other VWEs such as Croquet that are utilized for student education with different graphic format and program code. The ontology-based approach to model interoperability can be applied to other VWEs by adopting their requirement into the ontology.

**Evaluation for Contribution of VLF to Real Education**

The presented evaluations focused on the measuring learning efficiency with the VLF. It verified that students could improve their virtual field skill with the system by recursive measurement activities. But, it could not cover the effect of the virtual learning into the real experiments. In future, another evaluation for comparing measurement results in real forest of two groups (a group of students who experienced the VLF versus a group of student who have no experience) may explain the usability of the virtual forest eLearning system for real world education.

**Digital Forest Learning Library**

This study focused on improving a virtual world of longleaf pine ecosystem. It consisted of a kind of tree in the world, but it will be necessary to build more realistic ecosystem with diverse plants to give practical learning experience (Barton, 2008). The ontology-based methodology can be used to generate diverse forest virtual world by providing characteristic forest population and distribution information to the system. The generated forest world can be utilized by other virtual student as a library of virtual forest.

**Ontology Reasoning**

The ontology-based approach did not focus on ontology reasoning that is valuable function of ontologies. It could be used to selecting appropriate forest model and instrument for different field experiments.
APPENDIX
EVALUATION FORM OF VLF SYSTEM

A. These questions are prepared to understand users and user system.

A.1. User’s computer resource

A.1.1. What is the operating system of your computer?

A.1.2. How old is your computer?

A.2. User’s computer ability

A.2.1. How long have you played with 3D game?

A.2.2. How well do you know about Virtual World?

A.2.3. How much have you experienced Virtual Learning Environment?

A.3. User’s background

A.3.1. Did you take a class course related with field experiment?

A.4. User’s expectation
A.4.1. How much do you expect to get positive results on learning after you are instructed about VLF?

A.4.2. How much reality do you expect to experience before you start VLF?

B. These questions are prepared to answer by conducting virtual experiments step by step.

STEP #1. User guide & VFL tool box

B.1. How do you define the helpfulness of VLF user guide?

B.2. How do you define the accessibility of VLF Tool Box?

B.3. How do you define the intuitiveness of VLF Tool Box?

STEP #2. Measuring area tool

B.4. How do you define the usefulness of tool instruction at screen?

B.5. How do you define the correctness of displaying area center and tree number tags?

B.6. How do you define the clearness of area boundary?

B.7. How do you define the intuitiveness of the tool usage?
B.8. How do you define the realism of the tool?

STEP #3. DBH tool

B.9. How do you define the intuitiveness of the tool usage?

B.10. How do you define the realism of the tool design?

B.11. How do you define the usefulness of interface design for reading measurement?

STEP #4. Distance measuring tape tool

B.12. How do you define the usefulness of tool instruction at screen?

B.13. How do you define the intuitiveness of the tool usage?

B.14. How do you define the realism of the tool design?

B.15. How do you define the usefulness of interface design for reading measurement?

STEP #5. Clinometer tool

B.16. How do you define the usefulness of tool instruction at screen?

B.17. How do you define the intuitiveness of the tool usage?
B.18. How do you define the realism of the tool design?

B.19. How do you define the usefulness of interface design for reading measurement?

**STEP #6. Reporting note tool**

B.20. How do you define the usefulness of tool instruction at screen?

B.21. How do you define the intuitiveness of the tool usage?

**STEP #7. Iteration of experiments over target area**

B.22. How do you define the easiness of finding area centers by location tool?

B.23. How do you define your learning proficiency as iterating experiments through areas?

C. *These questions are prepared to get answers from user's experience after completing the virtual experiment test.*

C.1. How do you evaluate the VLF instruction taken before conducting the virtual experiment?
C.2. How long did you take to complete first measurements at first experiment area?

C.3. How long did you take to complete last measurements at last experiment area?

C.4. How much do you define the effect of computer performance?

C.5. How much do you define the easiness of controlling virtual instruments?

C.6. How much difference do you feel between your expectations for VLF and actual implementation?

C.7. How much do you define the completeness of the VLF system?

C.8. How much do you define the realism of the virtual contents?

C.9. How much do you define the value of system supports for understanding the VLF?

C.10. How much are you satisfied with the VLF?


SimFarm. 1993. Maxis Software Inc.


Virtual Newrological Education Centre. 2007. 


BIOGRAPHICAL SKETCH

Yunchul Jung, hailing from Ulsan, Republic of Korea, finished his schooling from Haksung High School. He studied agricultural engineering and acquired a bachelor’s degree from Seoul National University, Seoul. On August 2006, he entered the graduate program at University of Florida. From August 2006 to July 2008, he worked as a Research Assistant at the Department of Agricultural and Biological Engineering, UF, and completed his master's thesis on "An Ontology-based Approach to Simulation with Application to Citrus Water and Nutrient Management" under Dr. Howard W. Beck. After completing his Master of Engineering, he started the doctoral program at UF and was awarded a research assistantship under Dr. Howard W Beck in the area of information technology.