1-1-2010

Synchronous collaborative 3D GIS with agent support

Zheng (Eric) Chang
Ryerson University

Recommended Citation

Chang, Zheng (Eric), "Synchronous collaborative 3D GIS with agent support" (2010). *Theses and dissertations.* Paper 989.

Synchronous Collaborative 3D GIS with Agent Support

by

Zheng (Eric) Chang

Master of Applied Science, Ryerson University, Canada 2005
Master of Applied Science, Wuhan University, P. R. China 1997
Bachelor of Engineering, Wuhan University, P. R. China 1994

A thesis

presented to Ryerson University

in partial fulfilment of the

requirement for the degree of

Doctor of Philosophy in

Civil Engineering

Toronto, Ontario, Canada, 2010

Declaration

I hereby declare that I am the sole author of this thesis.

I authorize Ryerson University to lend this thesis to other institutions or individuals for the purpose of scholarly research.

Zheng Eric Chang
Department of Civil Engineering
Ryerson University

I further authorize Ryerson University to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

Zheng Eric Chang
Department of Civil Engineering
Ryerson University

Synchronous Collaborative 3D GIS with Agent Support

Zheng (Eric) Chang

Abstract

3D geographical information systems (GIS) software tools that support synchronous collaboration efforts among distributed decision-making participants can be very useful in many application areas, such as urban planning, engineering design, disaster and emergency management, and subsurface resources management. Although research has been extensively carried out in related fields such as groupware system, collaborative virtual environments and collaborative industrial design, a comprehensive study has not been found in the 3D GIS field. The scientific goal of this research is to add synchronous collaboration capability to the Internet-enabled 3D GIS environment. More specifically, the objective of this research is to investigate methods and key technologies to design a synchronous collaborative Internet -enabled 3D GIS environment (SC3DGIS).

In this thesis, two basic questions for synchronous collaborative 3D GIS are presented: 1) what are the special functions and usability for SC3DGIS compared to traditional and mainstream 3D GIS applications? 2) What are the special problems in design and development of SC3DGIS when considering it as a specific type of synchronous collaborative system? In answering the above two questions, a conceptual framework is developed to investigate main aspects which play core roles in reflecting the features of SC3DGIS. The more detailed system requirements are further analyzed through a case study. The prototype design adopts two layer structures - the shared 3D environment layer and the agent assistant layer. The shared 3D environment adopts a semi-replicated architecture, while in the agent assistant layer, a multi-agent method is used to solve the complex interactions between users and the shared understanding of all parties in the framework. Two walkthroughs are presented to validate the usability of the prototype.

This results of this research indicate that 1) adding synchronous and collaborative capabilities to a 3D GIS environment can significantly improve the efficiency and satisfaction of decision-making for the geographically-distributed people; 2) the events transferring method and semi-replicated architecture are more suitable for synchronous collaborative 3D GIS than a display imagery transferring method and centralized architecture because of the highly interactive 3D contents; 3) a 3D data model for collaborative purposes require explicit data state presentation, such as color and style; 4) a social collaboration model and its ontology presentation and shared catching method keep the distributed system consistent and understandable; and 5) a multi-agent method sitting on a shared 3D view can be very helpful in assisting group users to carry out more complex communications, such as discussion and making deals.

.

Acknowledgements

First of all, I would like to express my great appreciation and thanks to my supervisor, Dr. Songnian Li, for his valuable advice and ideas as well as constructive suggestions and productive comments on my thesis. I also thank him for providing me the great opportunity to continue my study in the field of GIS and to work with him as a research assistant during my years at Ryerson University. I greatly appreciate his efforts to provide me financial supports throughout the course and my thesis studies. Without his support, I would never have completed this thesis work.

I am very proud to be involved in the GEOIDE project SLM_DFM#15 and would like to extend my thanks to Dr. Donna Kirkwood, Dr. Jacynthe Pouliot and other faculty members and researchers in this project for their help and input to my research.

I also attribute my accomplishment to Dr. Michael A. Chapman, Dr. Jonathan Li, Dr. Ahmed El-Rabbany, Dr. Jinyuan Liu and Dr. Ahmed Shaker and other faculty and staff members in the Department of Civil Engineering for their help and support. Many thanks go to Desmond Rogan, Domenic Valle and other staff who provide technical assistance in solving computer and software problems, and to Mary Neelands, Kim Kritzer and Dianne Mendonca for their administrative support. I also wish to thank to Dr. Vincent Tao and Dr. Gunho Sohn for their support and help.

I wish to thank my fellow graduate students in the Program of Civil Engineering at Ryerson University, Xiaohong Ma, Chengfeng Xiong, Xinbao Chen, Nedim Oren, and Muhammad Butt. My study time with them has made many beautiful and happy memories.

I also wish to express my gratitude to my parents for their unconditional love and support, and to my wife, Weihong Zhan, my son, Cheng Cheng, and my daughter, Yue Yue, for their endless love, patience, and encouragement. Without their help and understanding, I would never have completed this thesis.

This thesis was completed with financial support from GEOIDE, Networks of Centers of Excellence (NCE), NSERC, Ontario Graduate Scholarship (OGS), Ryerson Graduate Scholarships and Awards of the Civil Engineering Department at Ryerson.

Table of Contents

List of Figures

List of Tables

List of Abbreviations and Acronyms

3D FDS	3D Formal Data Structure
ACL	Agent Communicative Language
API	Application Programming Interface
AWT	Abstract Windows Toolkit
CORBA	Common Object Request Broker Architecture
COM	Common Object Model
CSCW	Computer Supported Cooperative Work
CVE	Collaborative Virtual Environments
DBF	Database Format
ESRI	Environment Source Research Institute
EOC	Emergency Operations Center
FIPA	Foundation for Intelligent Physical Agents
JAXB	Java Architecture for XML Binding
JPEG	Joint Photographic Expert Group
GIF	Graphics Interchange Format
GIS	Geographic Information System
GML	Geography Markup Language
GUI	Graphical User Interface
HTML	HyperText Markup language
IP	Internet Protocol
ISIS	Integrated Software for Imagers and Spectrometers

KML	Keyhole Markup Language
LAN	Local Area network
JSDT	Java Shared Data Toolkit
MUD	Multiple User Dimension
MVC	Model-View-Controllor
MDA	Model Driven Architecture
MAS	Multi-agent system
OCX	OLE Control Extension
OGC	Open GIS Consortium
OO3D	Object-oriented 3D GIS model
PDA	Personal Digital Assistant
PNG	Portable Network Graphics
RPC	Remote Procedure Calls
QA	Quality Assurance
RDBMS	Rational Database Management System
SCBM	Synchronous collaboration behaviour model
SDK	Software Development Kit
SQL	Structured Query Language
SSM	Simplified Spatial Model
TCP/IP	Transmission Control Protocol/Internet Protocol
TIN	Triangulated Irregular Network
TEN	Tetrahedral Networks

VRML	Virtual Reality Modeling Language
UDM	Urban Data Model
WCS	Web Coverage Services
WFS	Web Feature Services
WTS	Web Terrain Service
WYSIWIS	What You See Is What I See
X3D	Extensible 3D
XML	Extensible Markup Language
XSL	Extensible Style Language
XTV	X Terminal View

Chapter 1 Introduction

Three-dimensional geographical information systems (3D GIS) are useful in many areas, such as urban planning, environmental monitoring, emergency management, geology and mining, and military. The typical functions of 3D GIS provide spatial query, analysis and visualization to people such as stakeholders, scientists and urban planners for scientific insights and decision-making. To enable these people, who are usually geographically dispersed, to work together, communicate with each other, and exchange and share 3D information for a common task in a timely fashion, a fundamental requirement is to share the 3D models and collaboratively manipulate models during the decision-making process.

From the site selection scenario, for instance, in order to evaluate and locate a new airport in an urban area, people need to share and discuss the proposed 3D airport model, the related 3D city environment, as well as a number of other factors such as land acquisition and construction costs, accessibility and relationship to existing services. These people might be from different stakeholder groups, such as land owners near the proposed sites and environmental activists, and would therefore likely hold diverse viewpoints on its solution. Consequently, solutions are often formulated by groups, such as committees or task forces, in which individuals from diverse backgrounds bring their expertise to the search for solutions and the interpretation of results. For example, considering the specialised knowledge base in geology and earth science especially in the GEOIDE Project[1], the 3D model is not easily understood by people who are not trained in this area. Thus, experts in

[1] This research is part of GEOIDE Project: SLM_DFM#15, http://geotopo3d.scg.ulaval.ca/index.htm

this field need to provide a detailed explanation and even operate the 3D model for demonstration. Sharing the 3D model, 3D environment and the operations will greatly help decision-makers and stakeholders understand the situation and ultimately come to an agreement.

Emergency management also faces similar collaboration requirements. Large and complex emergencies often affect multiple departments and agencies in which data need to be collected and assembled from a variety of locations quickly and under adverse conditions. Part of the role of the Emergency Operations Centers (EOC) is to understand the details of the emergency, order the required response resources, coordinate with agencies (federal, provincial, and local), and determine the immediate actions necessary to deal with the incident. In EOC, there are local first responder experts as well as a network of experts who provide additional support. This second layer of experts is usually remote and separate from the wealth of information and situation awareness provided by the emergency operation systems. Hence, these supporting experts frequently find themselves making decision without the benefit of some critical information that is available locally. Moreover, when major disasters happen, it may be nearly impossible to get all the personnel from their normal locations to the emergency operations center in major metropolitan areas. Major disasters produce nearly instantaneous gridlock and congestion. Key personnel cannot afford to be struck in traffic during the transit time which could be the most critical minutes and hours of the emergency.

The 3D GIS, therefore, might shift toward the next generation by creating a distributed and real-time collaborative 3D GIS environment. The basic requirements of such an environment will include: 1) Internet-based access, 2) shared visualization and shared control of functions and even applications, and 3) synchronous awareness, communications and collaborations among participants. This kind of 3D GIS is termed Synchronous Collaborative 3D GIS (SC3DGIS) in this research.

1.1 Objectives and Research Questions

The scientific goal of this research is to add synchronous collaboration capability to an Internet-enabled 3D GIS environment. More specifically, the objectives of this research are as follows:

1) Analyze and model synchronous and collaborative requirements for a 3D GIS environment, and develop the conceptual framework to identify the core social and technical problems for SC3DGIS.
2) Identify and assess the social factors and characterize the behaviours and relations of the parties involved in the environment.
3) Identify and design (or select) a suitable 3D GIS data model for synchronous collaboration purpose.
4) Design and prototype a synchronous collaborative 3D GIS (SC3DGIS) platform according to the above analysis.

This prototype should have the following features. First, the environment is an **Internet-enabled 3D system**. The users have the Internet access to the 3D system and also can load a Web-based 3D data model. Second, this environment supports **multi-user and real-time collaboration**. Multiple users can work together and carry out a common task in the 3D environment. Third, this environment is also a **GIS-oriented** system in which all types of 3D GIS-related requirements are investigated, especially in the geology field for the GEOIDE Project. Fourth, the environment is **well-structured,** enabling participants/users to work together on planned tasks, make negotiations and deals, and approve proposals with respect to related rules and roles.

The design and development of a SC3DGIS relates to fields of Computer Supported Cooperative Work (CSCW) and 3D Geographic Information System (3DGIS) (see Figure 1-1). Although the systems in both fields have existed for a long time, the SC3DGIS mentioned above is still not seen even in research stages. Rather than developing a framework to fit all kinds of situations, this research focuses on the real-time collaboration within a small group meeting. Within this context, two types of basic questions need to be identified (see Figure 1-1):

1) What are the special functions and usability for SC3DGIS compared to traditional and mainstream 3D GIS systems? Assuming that the 3DGIS has the synchronous collaboration capability, what kind of the systems can be developed through this capability?

2) What are the special problems (e.g. design constrains) in the design and

development of a SC3DGIS when considering it as a specific type of synchronous collaborative system?

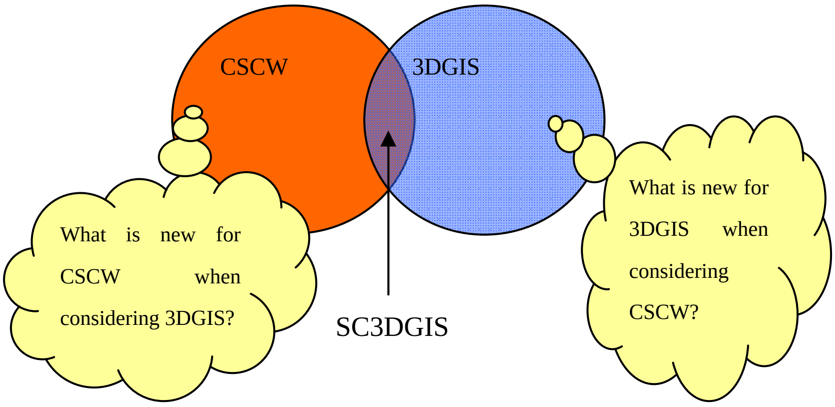

CSCW
3DGIS
What is new for CSCW when considering 3DGIS?
SC3DGIS
What is new for 3DGIS when considering CSCW?

Figure 1-1 Basic questions for SC3DGIS

The first question relates to what SC3DGIS can do if this 3D GIS has synchronous collaborative functions. In other words, this question is about how to look at the framework of SC3DGIS from user's point of view. Usually, this question will be answered through analysing user's requirements in case studies or mock-up use cases.

The second question relates to how to extend current collaborative system (groupware) technology to the 3D GIS field, which has many special features, such as 3D GIS spatial data model and user habits. These features are different from other synchronous collaborative systems, like teleconferencing and collaborative editing systems. Therefore, it is important to consider the unique characteristics of 3D GIS, especially spatial characteristics, when designing SC3DGIS.

More detailed questions will be addressed after the above questions are answered. These questions relate to the design and validation of the SC3DGIS and include:

1) What is the suitable architecture for SC3DGIS?
2) What are the requirements of 3D data support for SC3DGIS?
3) How can the 3D GIS environment in terms of participants, data and services/systems be well organized?
4) How does each client know each other, share the work context and have the same understanding?
5) How are the usability and performance requirements to be met?

1.2 Approach

1.2.1 Research Project Life Cycle

The life cycle of the research project is different from a software development project. For example, the former does not have the specified user requirements including the user interfaces or the detailed business logics which may cost the system designer more time to communicate with the end users. Instead, the requirements of this research project are derived from the assumed user work environments and thus are more static and stable. The objectives of the research project are clearly identified. These differences reduce the iterative process between user requirements and system design for the research project. On the other hand, achieving the research objectives is not just restricted within limited solutions. Instead, the choosing, revising, improving and testing of potential solutions to solve research questions is an iterative procedure and plays very important role in the research project. Bischofberger and Pomberger (1992) (Figure 1-2) introduced an experimental prototype approach in which the requirements definition, prototype and evaluation are iteratively processed.

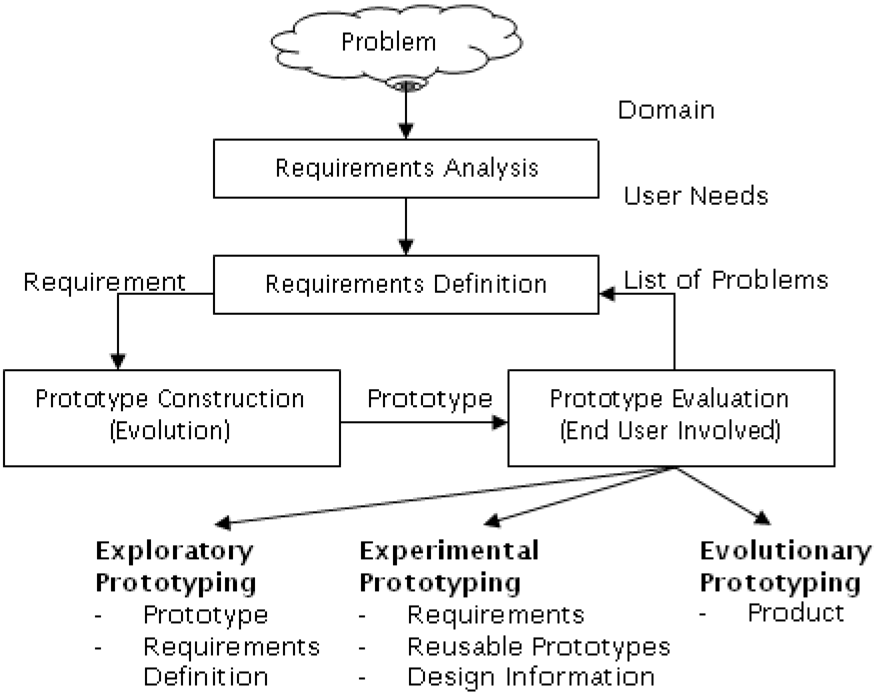


Figure 1-2 Different software prototyping paradigms (Bischofberger and Pomberger 1992)

In this research project, a mixed system development approach involving both an iterative waterfall approach and a prototype approach is adopted. The iterative waterfall approach is used for high level design issues, such as identification of research objectives and questions, high level architecture design, and testing of research results. The prototype approach is used for detailed design, development, and usability and performance testing.

The first stage is the requirements investigation for SC3DGIS (see Figure 1-3), after which the conceptual framework considering GIS criteria is introduced. Two main models are discussed: the social collaboration model and the 3D GIS data model for SC3DGIS. The social collaboration model is used to structure and present the shared 3D GIS environment and the users' behaviours. The 3D GIS data model is to meet the collaborative needs in SC3DGIS. In the third stage, a prototype, GeoLink3D, is designed and implemented. A multi-agent approach is also designed for complex communication and shared understanding. The last stage is to evaluate the models, the design and implementation methods. The evaluation results may provide some feedback to refine the above design and implementation methods.

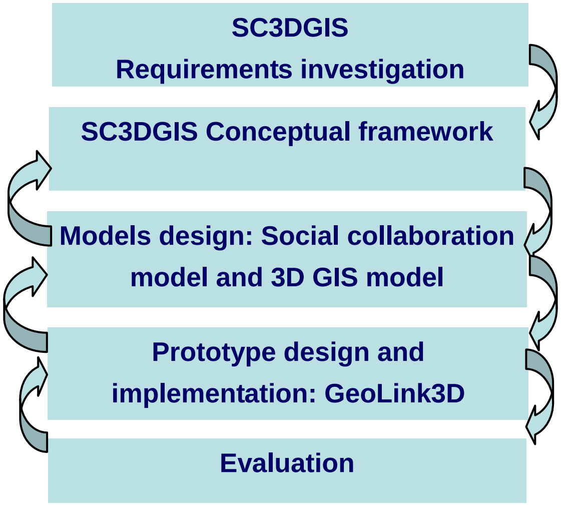

SC3DGIS
Requirements investigation
SC3DGIS Conceptual framework
Models design: Social collaboration model and 3D GIS model
Prototype design and implementation: GeoLink3D
Evaluation

Figure 1-3 Procedure of research

1.2.2 Levels of Abstraction

Abstraction is a cognitive means according to which people concentrate on the essential features of subject of thought and ignore irrelevant details in order to overcome complexity at a specific stage of a problem-solving situation (Hazzan and Kramer 2006). For example, the process, which is also a process of translation and extraction, from requirements collection to conceptual design is an abstraction process. These processes are also classified into different levels, which are termed as levels of abstraction. The higher level abstraction is close to the users' point of view, while the lower level abstraction is close to the developers' point of view (implementation).

In this research project, the research stages (see Section 1.2.1) include three levels of abstractions: requirements collection, conceptual design, and prototype implementation. There is some independency between these levels. For example, the conceptual design and modeling are independent from the system implementation. In other word, any programming language can implement the conceptual design if the performance is not considered. This kind of independency will significantly enhance the reusability of the design in the research stages. In the implementation stage, the system can be divided into different tiers, such as collaborative interfaces, 3D rendering, collaborative components and multicast network. Each tier's design also follows different levels of abstraction.

1.2.3 Requirements Collection

The requirements collection is an important step to answer one of the basic questions in Section 1.1: What can 3DGIS do if it has a collaborative capability? The GIS-specific criteria as the unique features are also involved in the requirements. There are two approaches to collect requirements for SC3DGIS: 1) from literature review and related groupware systems and 2) from case studies and mock-up scenarios.

Since there is no successful SC3DGIS system in existence to provide detailed information about what the SC3DGIS is, the requirements collection comes from the related collaborative 2D GIS research and system developments and other related fields such as groupware systems, collaborative virtual environments, collaborative manufacturing and construction design. However, the requirements from literature reviews or groupware system reviews are too general or may not be suitable for a small group discussion. For example, in collaborative virtual environments, people usually share the same 3D model with different viewpoints. In a small group discussion, people would like to share not only the model but also the same viewpoint, so that they can easily understand each other. Therefore, a case study for a small group discussion and its mock-up scenario is used to obtain more detailed and solid requirements.

The case study selected in this research relates to an urban planning field in which a typical small group decision-making process for a site selection is analysed. A mock-up scenario simulates the above process and meeting rules using a shared 3D environment. Through the scenario, collaboration-related user cases are collected and then become the important evidence of requirements for SC3DGIS. Nevertheless, as it is not possible to collect all the requirements for SC3DGIS, the requirements collected from the above methods include only the typical features and are limited to small group discussion.

1.3 Contributions

This thesis research has focused on addressing technical and social issues in the design and development of a synchronous collaborative 3D GIS, which has been paid relatively little attention in the field over the last few years. The research outcomes are expected to provide better understanding of such systems in terms of the complexity of the technologies involved, system design, and the social and human factors involved. In particular, this thesis makes the following contributions:

- A new conceptual framework, extended from the previous framework developed for 2D systems, describing the important functional, technological and social aspects for designing and developing 3D synchronous collaborative GIS. The framework was developed based on a comprehensive understanding of the special requirements in architecture, technology and 3D data modeling.
- A generic hybrid system architecture, based on event multicasting and message passing mechanisms, for the development and implementation of integrated, synchronously collaborative environments with 3D GIS tools for group decision-making and 3D information sharing.
- Design and implementation of a multi-agent mechanism to facilitate and improve reconciliation of inconsistent perspectives in a real-time collaborative 3D environment.
- A social collaboration model and its application-level ontology representation to provide shared understanding among the users, systems and data within a synchronous collaborative 3D GIS environment.
- Design and development of the first, to the best of my knowledge, workable prototype system that allows real-time sharing of and simple interaction with 3D models over the Internet.
- Identification of the main elements for the presentation of a generic 3D data model and

the criteria for the requirements of real-time collaborative 3D GIS. The criteria and the analysis results, such as coordinate system consideration and the comparing of the GML, X3D, KML and GRID, also benefit the 3D GIS community.

1.4 Organization

This thesis is organized into ten chapters. Chapter 1 presents an introduction of research objectives, research questions, methodology and contributions. Chapter 2 reviews the current work in collaborative GIS and related collaborative functions in other fields. Chapter 3 provides technological foundations in collaborative system design. Chapter 4 investigates the conceptual framework for SC3DGIS through a case study and requirements analysis. Chapter 5 presents a detailed investigation for the social collaborative model. Chapter 6 selects and designs a data model for SC3DGIS based on the specific criteria. Chapter 7 details the system design and prototype of the shared 3D GIS environment. Chapter 8 details the system design and prototype of multi-agent support. Chapter 9 introduces the user experience, usability and evaluation to the prototype. Chapter 10 presents the conclusions, discussions and recommendations for future work.

Chapter 2 Collaborative GIS and Related Work Review

From Chapter 1, it is known that SC3DGIS is a mixed research topic which involves CSCW and 3D GIS. The previous research and related work in these fields are critical to this research project. Through the review, the conceptual framework, current architecture for collaborative system, collaborative functions and key technological issues which could affect the design of SC3DGIS will be presented.

In this chapter, the basic concept of collaborative GIS and the classification of CSCW are first introduced. Then, the conceptual framework for collaborative GIS is presented and followed by the implementations of several typical collaborative GIS in 2D. Next, the 3D GIS and Internet-enabled 3D visualization systems are also reviewed. Finally, the observations of collaborative functions from other fields, such as computer conferencing and instant messaging systems, social networking sites and co-authoring systems, are presented.

2.1 Concept and Classification of Collaborative GIS

"***To work jointly with others or together especially in an intellectual endeavour***" **–** the above is the definition of collaboration in American Heritage® Dictionary of the English Language.

In a GIS community, there are several terms that describe this kind of group intellectual work such as **collaborative geographical information systems** (CGIS) (Churcher and Churcher 1996, Jones et al. 1997, Li and Coleman 2003, Balram and Dragicevic 2006),

geocollaboration (MacEachren and Brewer 2004, Cai 2005), or **group spatial decision support** (GSDS) (Armstrong and Densham 1995, Nyerges 1999, MacEachren 2001, Jankowski and Nyerges 2001). MacEachren et al. (2004) consider computer-supported geocollaboration to involve a committed effort on the part of two or more people to use geospatial information technologies to collectively frame and address a task involving geospatial information. In order to maintain consistency, the term **collaborative GIS** is used in this thesis.

Collaborative GIS closely relates to the concept of computer-supported cooperative work (CSCW). CSCW, through its technological implementation – groupware, supports group of people engaged in a common task (or goal) and provides an interface to a shared environment (Ellis et al. 1991). Groupware has been defined as “technology that communicates and organizes unpredictable information allowing dynamic groups to interact across time and space” (Cameron et al. 1995).

In both academia and industry, various groupware prototypes and products have emerged and provided particular functionalities to users. Each groupware system is designed to support a particular cooperative work situation or a particular range of cooperative work situations. Although cooperative work settings are very diverse in terms of task, duration, group, organizational context and culture (Hinssen 1998), they are usually classified as one of the four situations according to the temporal and spatial dimensions (Johansen 1988, Dix 1996): 1) same time (synchronous) and same place (co-located); 2) different time (asynchronous) and same place; 3) different time and different place (distributed); and 4) same time and different place. Table 2-1 shows the time-place matrix and the instances of the typical systems. The research carried in this thesis project falls into the last category.

Table 2-1 Time-place matrix (Johansen 1998, Dix 1996)

		Time	
		Same time	Different time
Place	Same place	Face-to-face (tabletop displays, meeting support tools)	Asynchronous interaction (project scheduling, coordination tools, shift work systems)
	Different place	Synchronous distributed (shared editors, video-and audio-conferencing tools)	Asynchronous distributed (email, newsgroups)

2.2 Conceptual Framework

The goal of the conceptual framework is to delineate important technological, social, and cognitive parameters that must be considered to extend, or reinvent geoinformation technologies to support work by groups. MacEachren and Brewer (2004) described a conceptual framework for geocollaboration activities in which human-centered factors such as problem context, collaboration tasks, and perspective commonality, and system-oriented factors such as temporal context, interaction characteristics, and tools to mediate group work, are considered important aspects for developing geocollaboration environments, testing their usability, and understanding their use and usefulness.

Nyerges and Jankowski (Nyerges 1999, Jankowski and Nyerges 2001) were concerned about the development and/or integration of group-based GIS technology with other computer technologies to facilitate group problem-solving, scientific visualization, and decision-making which have an inherently geographical character. They used the enhanced adaptive structuration theory (EAST) and its enhanced successor, the EAST-2 framework, to describe significant issues for characterizing group decision-making (Jankowski and Nyerges 2001, 2003). EAST (EAST-2) consists of eight constructs and is grouped according to convening, process, and outcome categories to communicate their roles in decision situations (Figure 2-1).

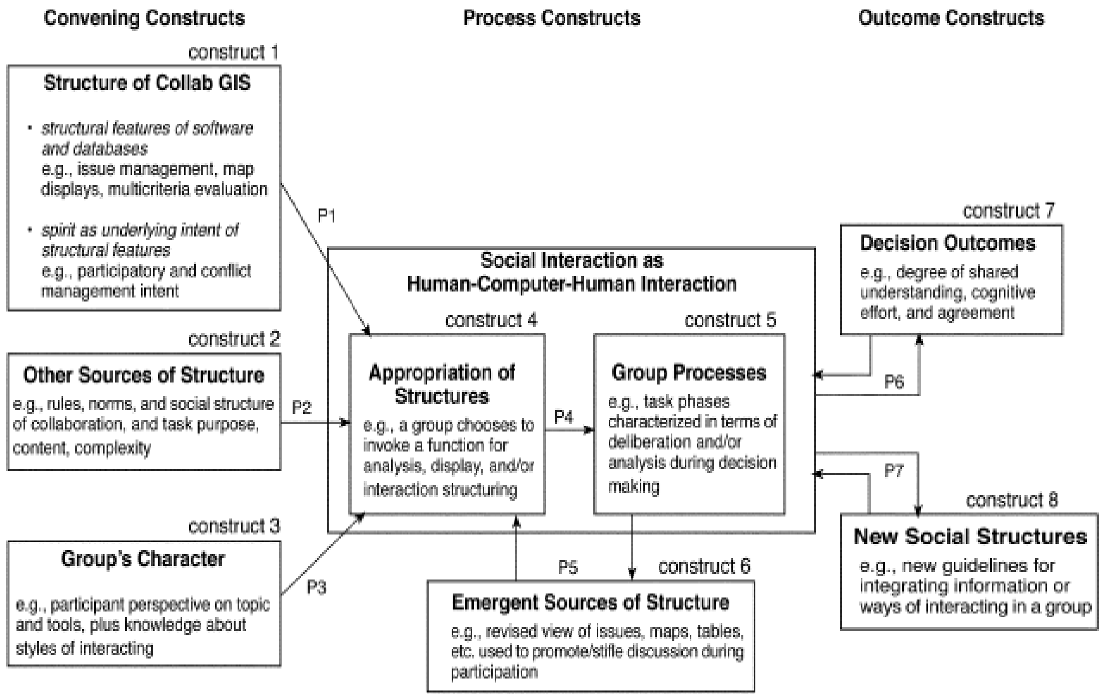

Convening Constructs
Process Constructs
Outcome Constructs
construct 1
Structure of Collab GIS
• structural features of software and databases
e.g., issue management, map displays, multicriteria evaluation
• spirit as underlying intent of structural features
e.g., participatory and conflict management intent
construct 2
Other Sources of Structure
e.g., rules, norms, and social structure of collaboration, and task purpose, content, complexity
construct 3
Group's Character
e.g., participant perspective on topic and tools, plus knowledge about styles of interacting
P1
P2
P3
Social Interaction as
Human-Computer-Human Interaction
construct 4
Appropriation of Structures
e.g., a group chooses to invoke a function for analysis, display, and/or interaction structuring
P4
construct 5
Group Processes
e.g., task phases characterized in terms of deliberation and/or analysis during decision making
P5
construct 6
Emergent Sources of Structure
e.g., revised view of issues, maps, tables, etc. used to promote/stifle discussion during participation
construct 7
Decision Outcomes
e.g., degree of shared understanding, cognitive effort, and agreement
P6
P7
construct 8
New Social Structures
e.g., new guidelines for integrating information or ways of interacting in a group

Figure 2-1 EAST (Jankowski and Nyerges 2001 2003) framework provides a conceptual map for understanding a group decision support situation.

The convening constructs articulate what is important in setting up a decision task: the organizations to be represented, the people from those organizations who are to participate, and the information technology that can be made available. These construts correspond to social-institutional influence, group participant influence and participatory GIS influence.

The process constructs include dynamics of invoking decision aids, management of decision tasks from phase to phase, and emergence of information structures, such as maps, models and databases. The outcome constructs and associated aspects include direct outcomes related to the specific decision task, and the social relations created, evolved, and/or destroyed when the task is completed.

Dragicevic and Balram (2004) made more specific efforts in structuring and managing distributed planning processes with a Web-based collaborative spatial Delphi framework, in which remote users can share maps, annotation and text comments at the same or different times. Cognitive mapping, the Delphi procedure, and consensus approaches are the main components integrated to structure a shared virtual space for problem-solving and planning. They also presented a collaborative modeling framework with agent UML (AUML) to solve more complex interactions between natural and human systems (Balram and Dragicevic 2006).

Gong and Lin (2006) also developed a concept framework for collaborative virtual geographic environment (CVGE), a distributed geographical world. They argued that geocollaboration is involved in four components: geographic environment, geo-tasks, task-related geo-problems, and multi-participants. Multi-participants mediated by collaborative tools act in unison to accomplish geo-tasks and solve geo-problems contained within a geographic environment. The system is based on client-server architecture. A case study and a prototype system were designed and developed by Java, Java3D, and VRML to explore the methodologies of collaborative spatial planning of silt dam systems.

The factors of the conceptual framework for collaborative GIS are mainly drawn from two aspects: social/institutional aspect and technological aspect. The former relates to social structures such as rules, policy, social norm and domain characteristics. The latter relates to technological supports to the former aspect. The technological supports may include current IT technologies, GIS software and decision-making tools. However, these factors may change significantly according to the specific domain characteristics and research focuses. For example, both EAST framework and collaborative spatial Delphi framework work on similar group decision-making situations, but the factors of the conceptual framework are different.

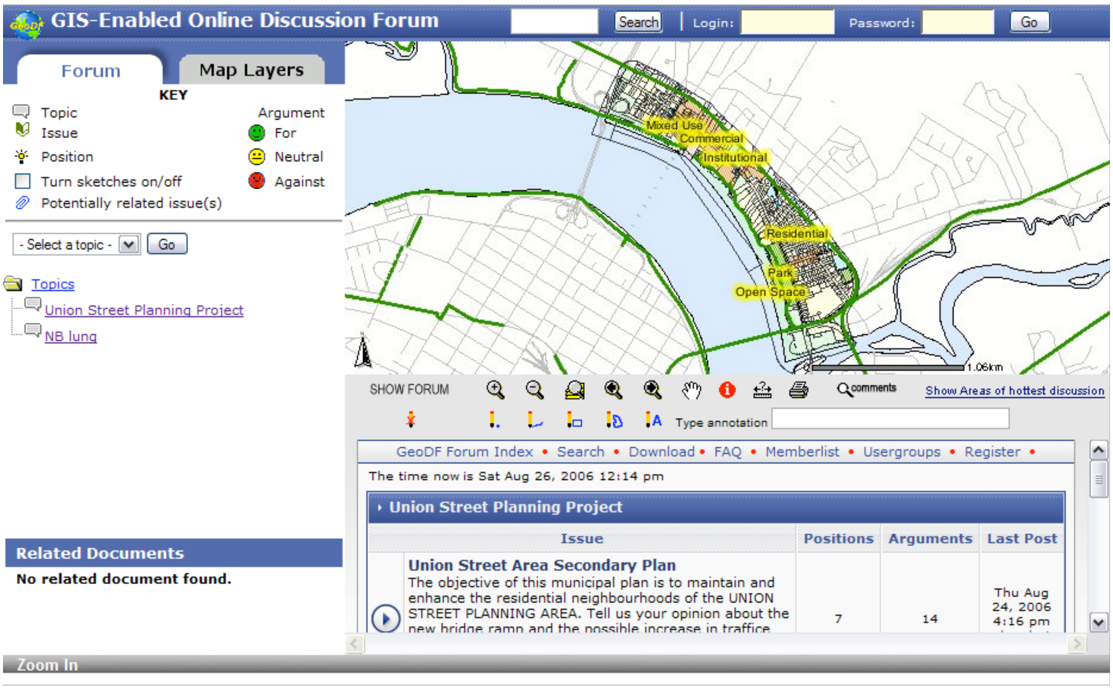
Moreover, although these factors are based on real work contexts, the related technological supports such as collaborative software tools are not available. In other word, these conceptual frameworks cannot provide guidelines for the design and development of the collaborative software. There exists a gap between the conceptual framework and collaborative system design. Compared to the above, the conceptual framework of SC3DGIS has unique features: focusing on real-time 3D GIS collaboration, scoping in small-group meetings, and bridging the gap between the conceptual framework and implementation.

2.3 Collaborative GIS Development

Although collaborative GIS can be classified into four types in terms of time and space according to Section 2.1, there are some other frequently used terms in the GIS fields such as: Public Participant GIS (PPGIS) (different time or same time, different place), Multimodel Collaborative GIS (same time, same place) and Synchronous Collaborative GIS (same time, different place).

2.3.1 PPGIS

PPGIS is used to help neighbourhood community groups and individuals use mapping and spatial analysis in the participatory process in a variety of areas, including community development, neighbourhood revitalization, environmental justice and natural resources management (Wong and Chua 2001). PPGIS application tools may include features, such as email/feedback forms, sketching and annotation, map attachments, geo-referenced comments and an online discussion forum. Tang et al. (2005) selected several Web-based PPGIS applications for evaluation. The evaluation criteria include: 1) enable experts to play the facilitators role, 2) exchange of views, 3) documentation and sharing of evolution of ideas, 4) showing decisions in context to related decisions, and 5) effective communication of spatial context. In order to meet all the criteria, Tang also developed a GIS-enabled online discussion forum (GeoDF) based on ArcIMS and phpBB (see Figure 2-2).



No related document found.

The time now is Sat Aug 26, 2006 12:14 pm

Union Street Planning Project

Issue	Positions	Arguments	Last Post
Union Street Area Secondary Plan The objective of this municipal plan is to maintain and enhance the residential neighbourhoods of the UNION STREET PLANNING AREA. Tell us your opinion about the new bridge ramp and the possible increase in traffice	7	14	Thu Aug 24, 2006 4:16 pm

Figure 2-2 Main user interface of GeoDF (Tang et al. 2005)

At the same time, GeoVPMS was also developed by Li et al. (2007) with open source software. In Geo-Forum, the PPGIS main features include geo-map oriented online communication, notification, feedback and discussion. For example, users who may be geographically dispersed can get online information and have discussions using a shared online map and send feedback in a nearly real-time pattern.

Rinner (2001) proposed and developed a GIS-based argumentation map, Argoomap, which supports online discussion and planning based on Google Map. The concept of argumentation maps provides explicit links between arguments and the geographic objects they refer to. These geo-argumentative relations not only allow for cartographic representation of arguments, but also support the querying of both space and discussion. ArgooMap was customized as ParticipatoryGIS.com by Boroushaki and Malczewski (2009) to address simultaneously deliberative and analytic dimensions of spatial decision-making and planning in an integrated and cohesive fashion. ParticipatoryGIS.com uses the server-side architecture approach to Web-based GIS. It employs HTML, CSS and JavaScript on the client-side and a combination of PHP scripting language and a MySQL database on the ParticipatoryGIS.com server. The Google Maps server provides the map and Google Maps API.

2.3.2 Multimodel Collaborative GIS

A multimodel collaborative GIS mediates distributed thinking and decision-making through the use of large-screen displays supporting multi-user, natural interaction in a same-time, same-place group work environment. MacEachren et al. (2003) developed two systems, HI-SPACE and DAVE_G, in which users can make use of hand gestures as a mechanism for specifying display locations. HI-SPACE adopts a drafting table as the display device, while DAVE_G adopts an electronic white board (see Figure 2-3). Both systems can track the hand position and identify individual gesture poses (e.g., two fingers extended), and understand speech/gesture requests for most commonly used map display functions such as “show a map of population within Pennsylvania”, “zoom here”, “highlight these features”, “make a one-mile buffer around these features”.

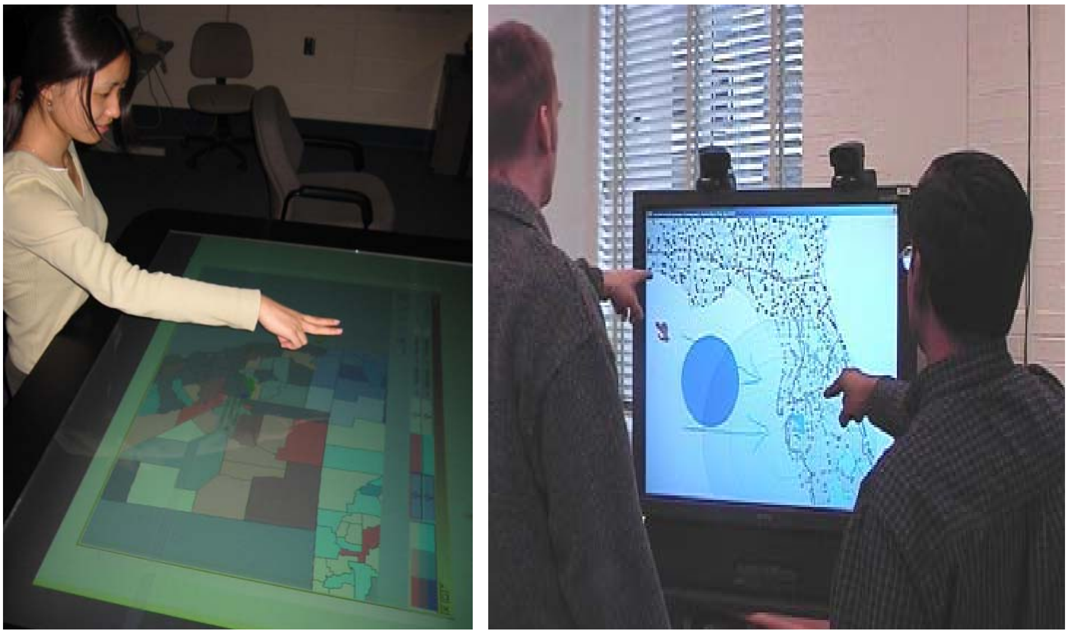
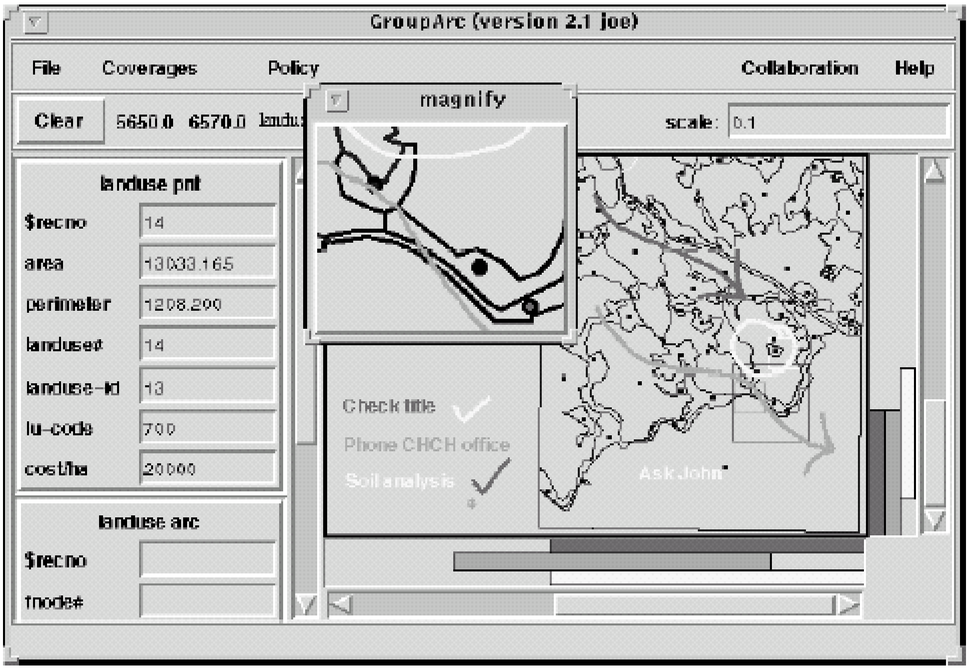


Figure 2-3 Gesture interface, HI-SPACE (left); Two-person, gesture-speech interface, DAVE_G (right)

2.3.3 Synchronous Collaborative GIS

Synchronous collaborative GIS (SCGIS) focuses on users' collaborations in a GIS environment with same time and different place pattern. The difference between SCGIS and PPGIS lies in the synchronized time pattern. In SCGIS, the GIS environment can be synchronized to all clients while in PPGIS, the GIS clients do not have the synchronized GIS environment. For example, in GeoDF, although users can concentrate on the same topic and area of interest (AOI), the clients' GIS environments, such as map extension and scale, are different, while in the SCGIS, all the clients may be forced to have the same GIS environments. The early efforts on synchronous collaborative GIS, e.g., GroupArc (Churcher and Churcher 1996), Habanero (Chabert et al. 1998) and Toucan Navigate (InfoPatterns 2007) focused primarily on combining GIS with CSCW software (groupware).

GroupArc (see Figure 2-4) was initially developed to explore the potential of lightweight CSCW browsers for GIS applications. It was written in Tcl/Tk language (Ousterhout 1994), runs on Unix, Macintosh and Windows platforms and uses GroupKit. GroupKit (Roseman et al. 1992) is a toolkit for building a general class of collaborative applications and includes a number of awareness widgets for use in GroupKit based applications.



scale: 0.1

landuse pnt

$recno 14

area 13033.165

perimeter 1208.290

landuse# 14

landuse-id 13

lu-code 700

cost/ha 20000

landuse arc

$recno

fnode#

Figure 2-4 Interface of GroupArc (Churcher et al. 1996)

When GroupArc is running, GroupKit manages the registration of conference participants (who may enter or leave at any time) and communication between the GroupArc replicas on individual participant’s workstations (see Figure 2-5).

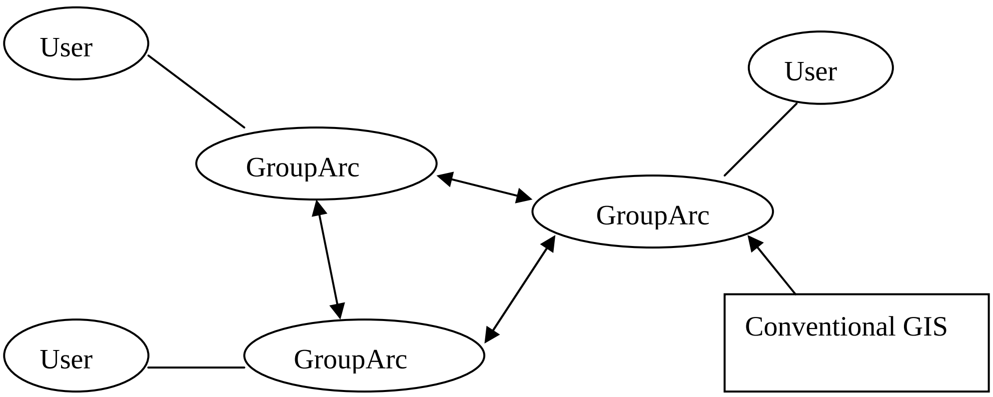


Figure 2-5 Users communicate through GroupArc (Churcher et al. 1999)

GroupArc allows physically separated users concurrently to browse and annotate GIS data in a cooperative way. Each participant must have Internet access and also their own copy of the GroupKit and GroupArc software. The participants can therefore stay in different buildings, cities or out in the field and all participate in the discussion (Churcher et al. 1999). Therefore, GroupArc is a replicated collaborative GIS system.

Habanero, developed by the National Center for Supercomputing Application at the University of Illinois at Urbana-Champaign, is a collaborative framework and environment containing a set of applications. Through Habanero one can interact with other people on the Internet using a variety of applications that share state and events. Habanero is written in Java and runs under any operating system that supports Java 2 and JINI v1.0 (Chabert 1998).

The Habanero client, server and applications provide the necessary environment to create collaborative workspaces and virtual communities. The server hosts sessions and connects the clients that interact with the sessions using a variety of applications called Hablets (see

Figure 2-6 and 2-7). Sessions can be recorded, persistent, access-restricted and even anonymous. The Habanero client provides the interface to define, list, create, join and interact with a session. The client provides session information, user identification, a notification mechanism, record and replay capabilities, security, a list of active users and tools, an address book and the capability to easily create session templates.

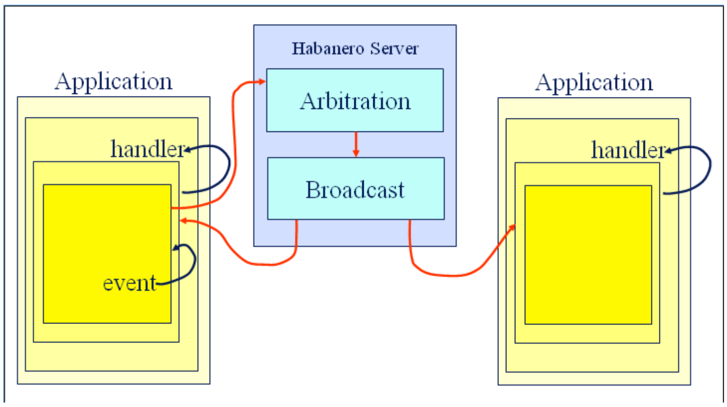


Figure 2-6 Habanero's Event interception (Chabert 1998)

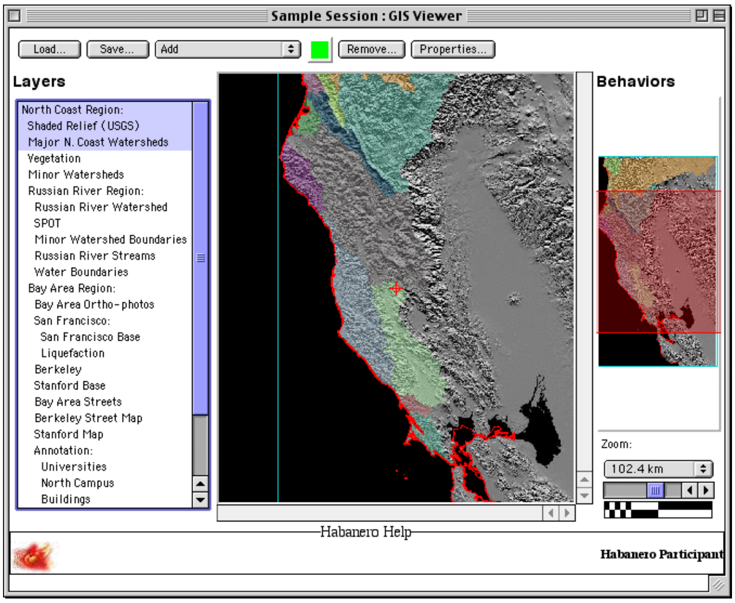


Figure 2-7 GIS view Habanero (Chabert 1998)

The above two prototypes illustrate the typical methods for synchronous collaborative GIS, a commercial GIS system with a groupware extension (e.g., GroupArc: Arcinfo + GroupKit) or a groupware environment embedding a GIS component (e.g., Groove Framework and Toucan, Habanero and GIS viewer). However, Cai (2005) argues that these two approaches are equally bad strategies for geo-collaborative applications because of heavy dependency on one or more particular CSCW, GIS, or other involved software systems, which causes difficulties in terms of deployment, maintenance and adaptation.

GeoLink (Chang and Li 2007), a synchronous collaborative 2D GIS platform, is an early version of Geolink3D prototype. It adopted a replicated architecture. The client is a rich or thick client (see reference in 3.2.2) with full GIS functionalities. The real-time collaboration is relying on a pseudo-layer in every client where every mouse event or key board event can be captured and sent to other clients. The pseudo-layer is a glass pane in a Frame above the GIS layer (See Figure 2-8). Through consistency mechanism, all the clients can work together and at the same time keep the whole system in consistency.

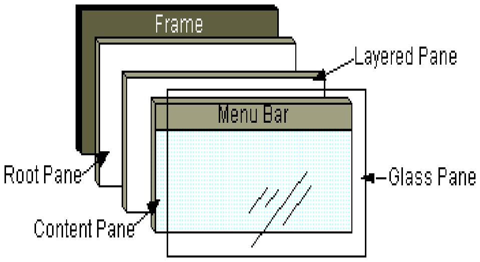


Figure 2-8 Glass Pane method used in GeoLink2D

In summary, the design and development of synchronous collaborative GIS has to rely on the progress of the development of the CSCW (groupware) system because much of the work with respect to architecture, consistency and multi-user interface is similar.

2.3.4 3D GIS Systems

Several state-of-the-art 3D GIS systems, such as 3D Analyst of ArcGIS from ESRI Inc. and Image VirtualGIS from EADAS Inc. are active and widely used in current GIS communities. Their main functions are to provide 3D GIS data presentation (visualization) and simple 3D GIS data analysis and query. Web Terrain Service (WTS) (OGC 2003) proposed by the Open Geospatial Consortium (OGC) is an open standard for Web based 3D GIS to visualize terrain data. Some researches also used Virtual Reality Modeling Language (VRML) to visualize 3D GIS data in a Web-based environment. Although some of these systems or tools provide Web-based visualization capability or GIS related query functions or even true 3D data model, no system or prototype is found to support real-time collaboration in a 3D GIS environment.

The main commercial 3D GISs are still single user-based, standalone systems, such as 3D Analyst of ArcView from ESRI Inc., Image VirtualGIS from EADAS Inc., GeoMedia Terrain from Intergraph Inc., and PAMAP model of PAMAPGIS from PCIGEOMATICS (Zlatanovaa et al. 2002). These systems are neither Web-based nor synchronously collaborated. While VRML together with georeferencing considerations, GeoVRML, has been investigated for Web-based geospatial 3D data visualization and analysis (Rhyne 1999, Huang et al. 2001), most of these studies neither involved real-time collaboration activities nor provided any distributed and multiple users 3D framework. Gong and Lin (2006) described a collaborative virtual geographic environment (CVGE) in a 3D distributed geographical world. The system framework was based on client-server architecture. A case study and a prototype system were designed and developed with Java, Java3D, and VRML to explore the methodologies of collaborative spatial planning of silt dam systems. However, the client-server architecture is limited in handling real-time collaborative computing. Stock and Bishop (2006) developed an envisioning system to study community values and interactions with their landscape (3D models) in a workshop environment. Some virtual reality technologies and personal digital assistant (PDA) devices are used in this system.

With the emerging of various 3D "digital earth", for example Google Earth 3D with KML and Microsoft’s 3D Virtual Earth, the 3D systems with some collaboration functions also appear, mostly as extension or add-on tools of the core and “heavy” system/server. Examples can be also found in TerraExplorer (Skyline 2006), Leica Virtual Explorer (Leica 2007) and Unype (Unype 2007). Through sending connection information, the users can get a shared 3D view using these systems. TerraExplorer supports ESRI shapefile format, Oracle spatial, ArcSDE and Map Feature Service. Leica Virtual Explorer supports ESRI shapefile format and 3D Terrain data. A more interesting system comes from Unype with which users can experience a multi-user Google Earth 3D (Skype 2007). It is worth mentioning that ESRI shapefile and Map Feature Service data (OGC Simple Feature format) provide z value to point, line and polygon to present 3D surface model. More detailed technical discussion can be found in Chapter 6.

Close examination of these 3D systems provides a number of observations that are useful to the research. First, these collaboration tools are not stand-alone systems and always work with a heavy core system which normally includes a collaboration server handling the synchronous collaboration like Toucan Navigate and a global dataset server providing 3D data and images like Google Earth 3D. This kind of "heavy" architecture hinders the system extendibility and scalability. It is difficult to extend these systems to meet application-specific requirements related to collaborative 3D GIS. Second, the main 3D GIS systems do not support true 3D models. Almost all of them are 2.5D systems which focus primarily on large scale (city level) data models. Third, not all of these systems have extension or add-on tools for real-time, synchronous sharing of 3D views. For example, both Google Earth and Microsoft Virtual Earth only allow asynchronous sharing of maps and 3D views through emailed links.

2.4 Observations of Collaboration in Other Fields

In order to better understand the collaborative work, this section presents the observations of collaborative activities in different fields. Some of the systems may not be considered as a collaborative system. However, they do include support (functions) for collaborative activities.

Computer conferencing

Microsoft NetMeeting (Microsoft 2000) is a typical computer conferencing system in the early years. Any application can be shared by all the participants through this system, which is called “shared application”. In the shared application, the host who launches an application can grant his own operation to others so that other users can not only remotely watch the view but also operate the system. WebEx (CISCO 2008), E TEAM (NC4 2008) and remote desktop control offer similar functions using the same shared application architecture. These kinds of systems usually have a video camera, microphone and text chat to facilitate communications among users. Such tools can also be seen in Instant Messaging systems.

Instant Messaging Systems

Windows Live Messenger, Yahoo Messenger, Google Talk, and Tencent QQ (Tencent 2008) are the typical and the most popular instant messaging services in the world. Users can be instantly connected with each other through text, voice or video. They can also share the personal files, photos, and videos while talking and even play games and share applications. These kinds of services are really synchronous collaborative systems according to collaboration classification as shown in Table 2-1.

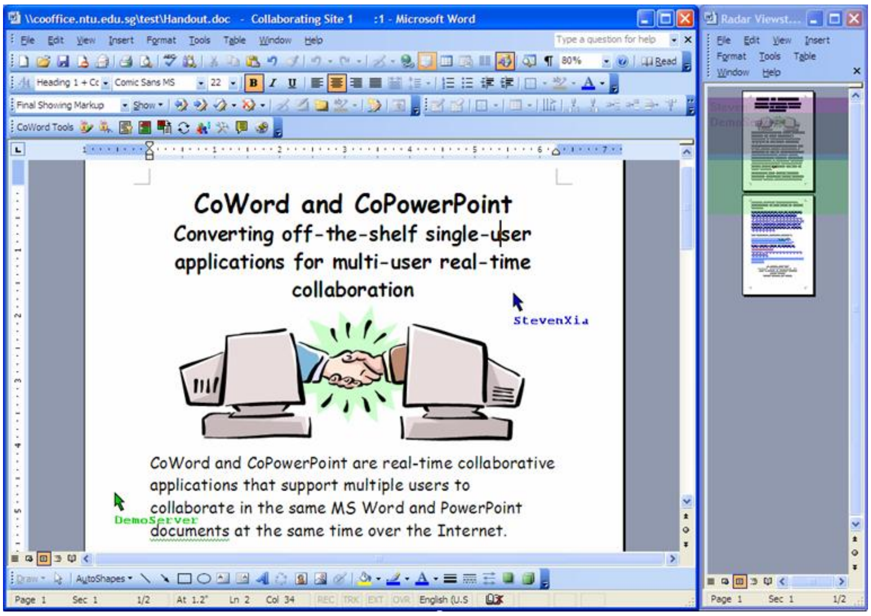
Social Networking Site (SNS)

Recently, social network sites such as Friendster, Facebook and MySpace, as well as content-sharing sites that also offer social networking functionality (including YouTube, Flickr, Upcoming, del.icio.us, Last.fm, and 43 Things) have captured the attention of millions of users. A social network site is also termed a social networking service or virtual community in which individuals can: (1) construct a public or semi-public profile within a bounded system, (2) articulate a list of other users with whom they share a connection, and (3) view and traverse their list of connections and those made by others within the system (Boyd and Ellison 2007). SNSs usually offer the following basic functions: network of friends listings (showing a person's "inner circle"), person surfing, private messaging, discussion forums or communities, events management, blogging, commenting (sometimes as endorsements on people's profiles), and media uploading (Breslin et al. 2007).

Co-authoring System

Co-authoring systems such as Microsoft Office Grove (GROVE 2008), CoWord and Google Docs (Google 2008) help collaborated users edit the same MS Word document anytime (the same time or different times) over the Internet (Figure 2-9). Changes made to the document are immediately displayed for all users to see.

Office Groove 2007 is a collaboration software program that helps teams work together dynamically, even if team members work for different organizations, work remotely, or work offline. CoWord is powered by GCE (Generic Collaboration Engine) (Xia et al. 2004). GCE integrates a comprehensive collection of collaborative technologies and can be used to convert existing and new single-user applications into advanced collaboration tools without making any change to the source code of the original application.



CoWord and CoPowerPoint

Converting off-the-shelf single-user applications for multi-user real-time collaboration

CoWord and CoPowerPoint are real-time collaborative applications that support multiple users to collaborate in the same MS Word and PowerPoint documents at the same time over the Internet.

Figure 2-9 CoWord Interface (CoWord 2008)

Concurrent Versions System

A Concurrent Versions System (CVS), also known as version control system keeps track of all work and all changes in a set of files, and allows several developers (different space and different time or close to the same time) to collaborate. CVS uses client-server architecture: a server stores the current version(s) of a project and its history, and clients connect to the server in order to "check out" a complete copy of the project, work on this copy and then later "check in" their changes. Typically, the client and server are connected over a LAN or over the Internet. Clients can also compare versions, request a complete history of changes, or check out an historical snapshot of the project as of a given date or as of a revision number.

Collaborative Virtual Environments

Collaborative virtual environments (CVE) have been developed for a wide range of applications. The main applications include virtual reality and augmented reality, multiple player video game and battlefield simulation. The typical platforms and tools include DIVE, MASSIVE-3, Secondlife, Counter-strike and War Warcraft (Jeffery 2008). These CVEs focus on creating 3D visualization environments that can be applied in many scientific and engineering fields in order to avoid the use of physical prototypes, to train people in high-risk situations, and to interpret real or simulated results (Theotisto and Fairen 2005). They are either a computer-based, distributed virtual space, or a group of spaces in which people can meet and interact with each other, with agents or virtual objects. Most of the current CVEs provide a highly interactive virtual environment in which each user is presented with the same virtual world in the same way from different viewpoints.

Collaborative manufacturing and construction design

Collaborative manufacturing and construction design (e.g., Adaptive Media Envision3D, Autodesk Streamline, Alibre Design and CollabCAD) focus on real-time collaboration support for 3D CAD (Fuh and Li 2005). Using these collaborative design environments, multiple designers can efficiently work together to discuss and modify the evolving 3D CAD output in both co-located and distributed situations. Because a lot of 3D data modifications are involved in this kind of scenario, keeping consistency is a major challenge and thus causes a much stricter consistent environment than virtual reality and augmented reality.

Summary

The typical collaborative functions in different kinds of systems are summarized in Table

2-2.

Table 2-2 Summary of groupware systems

Groupware Category	**System Cases**	**Type**	**Collaborative Functions Descriptions**
Computer conferencing and instant messaging system	WebEx, ETeam, MSN, Yahoo message, QQ message	Synchronous Distributed	• Setting up sessions, audio, video, shared application connections • Adding late joiners, more than 2-way connections • Integration of other media (phone, video and text chat) • Providing text-based computer-mediated discussions between users. • Each letter or sentence that is typed is immediately observable on the screens of other users, which facilitates rapid turn taking in discussions.
Social Networking Site	MySpace, Facebook	Asynchronous Distributed	• Network of friends listing and searching • Private messaging • Discussion forums or communities • Events management • Blogging • Commenting
Co-authoring systems	Microsoft GROVE, CoWord	Synchronous Distributed	• Different phases of authoring, e.g., brainstorming, doing research, planning, writing, and reviewing • Simultaneous document editing and/or sequential document editing • Annotations, versions and revisions • Communication between authors about the document or the authoring process • Coordination of the authoring process

Concurrent Versions System	CVSNT, OpenCVS	Asynchronous Distributed	• Keeping track of all work and all changes • Comparing versions • Request a complete history of changes • Checking out a historical snapshot of the project
Collaborative virtual environments, Multiplayer games, virtual worlds	DIVE, MASSIVE-3, Counter-strike	Synchronous Distributed	• Creating a virtual place populated with avatars that can navigate and interact with other people and objects in the environment • Persistent places • Containment and tracking of objects • User extensible • Shared video and audio, selective groupings of users • Multi-user text chat, multi-user dungeon (MUD)
Collaborative manufacturing and construction design	Adaptive Media Envision3D, Autodesk Streamline	Synchronous Distributed	• Simultaneous 3D editing • Shared 3D environment • Communication between authors

Chapter 3 Technological Foundations

Collaborating systems have special features in comparison to other information systems. While these features differ from each other in terms of interfaces, architecture and work context, they may, however, share similar technical foundations. This section presents an overview of key technologies especially in multiple user interfaces, architecture and the consistency issue. Some of these technologies will be selected and used in the design and development of the prototype system described in Chapter 9.

3.1 Awareness and Collaborative Interface

In the collaborative system, the graphic user interface (GUI) is very different from other systems. The main difference is that the user can be aware of other users' activities through the GUI. The collaborative GUI usually plays an important role in coordinating the tasks, avoiding conflicts and sharing activities. Typical collaborative GUIs include shared view, radar view and telepointer. Because there are no special developing tools to develop these kinds of GUIs, special technologies and even special system architectures are required to support them. In the following sections, the basic concept of awareness is first introduced, after which multiple user interfaces such as shared view, radar view, and telepointers are delineated. The section ends with a consideration of geographical features.

3.1.1 Awareness

The conception of awareness involves states of knowledge as well as dynamic processes of perception and action. Awareness, especially for workspace awareness, is the up-to-the-moment understanding of another person's interaction with a shared workspace (Gutwin and Greenberg 1996). Awareness involves knowledge about where others are working, what they are doing, and what they are going to do next. Four basic characteristics can be observed from prior work on awareness (Adams et al. 1995, Norman 1993, Endsley 1995).

1. Awareness is knowledge about the state of an environment bounded in time and space.
2. Environments change over time, so awareness is knowledge that must be maintained and kept up to date.
3. People interact with and explore the environment, and the maintenance of awareness is accomplished through this interaction.
4. Awareness is a secondary goal in the task – that is, the overall goal is not simply to maintain awareness but to complete some tasks in the environment.

Gutwin and Greenberg (2002) presented a descriptive framework of workspace awareness for a real-time groupware. The framework tried to answer three questions: 1) what the elements of workspace awareness are; 2) what mechanisms are used to maintain it, and 3) when it is used in collaborative work situations. These three parts correspond to three tasks that the groupware designer must undertake in supporting workspace awareness, namely: understand what information will be provided, determine how the knowledge will be gathered, and determine when and where the knowledge will be used (see Figure 3-1). Detailed workspace awareness (WA) elements are presented in Table 3-1.

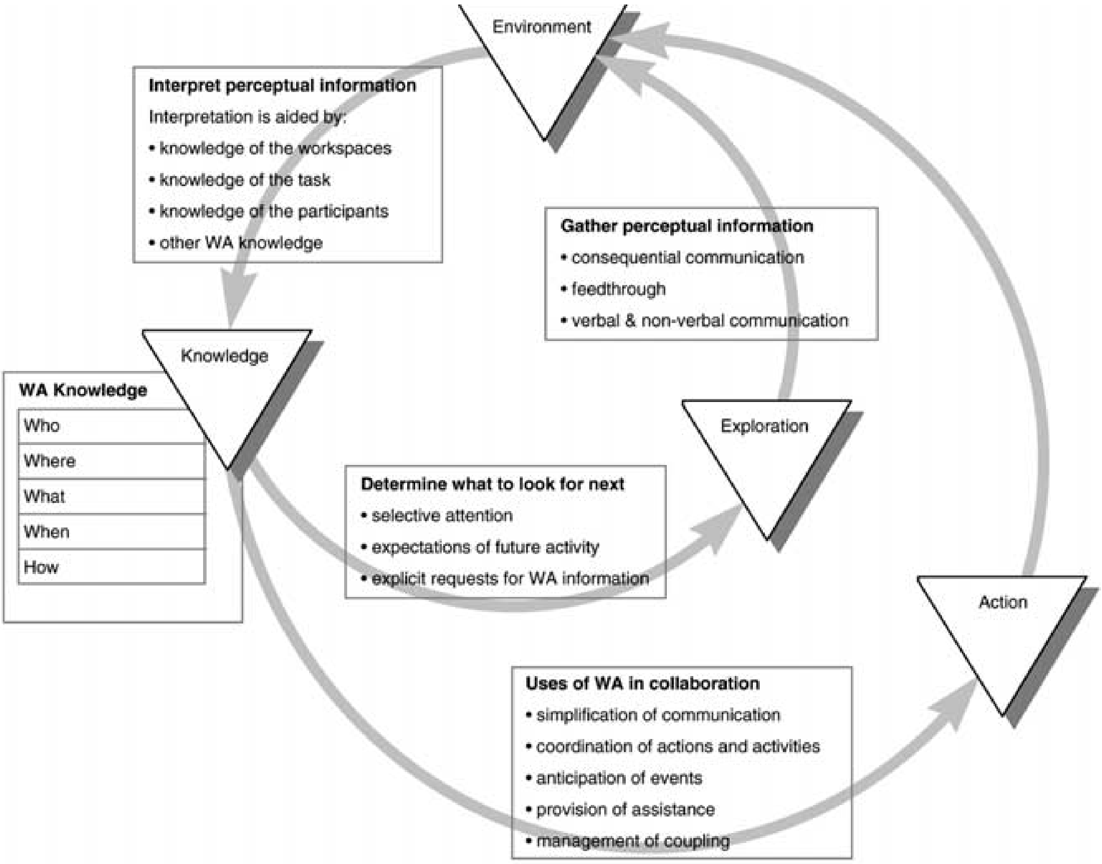

Environment
Interpret perceptual information
Interpretation is aided by:
• knowledge of the workspaces
• knowledge of the task
• knowledge of the participants
• other WA knowledge
Gather perceptual information
• consequential communication
• feedthrough
• verbal & non-verbal communication
Knowledge
WA Knowledge
Who
Where
What
When
How
Exploration
Determine what to look for next
• selective attention
• expectations of future activity
• explicit requests for WA information
Action
Uses of WA in collaboration
• simplification of communication
• coordination of actions and activities
• anticipation of events
• provision of assistance
• management of coupling

Figure 3-1 Work space awareness framework (Gutwin and Greenberg 2002)

Based on the framework, the authors gave more examples to the design of groupware interfaces or collaborative wedges. Typical examples/wedges include: participant list, telepointers, view rectangles, radar view, multiple views, shared views, avatars, action indicators, location indicator and multi-user scrollbars. The main purpose of these collaborative interfaces is to facilitate user awareness.

Table 3-1 Elements of workspace awareness relating to the present (Gutwin and Greenberg 2002)

Category	Element	Specific Questions
Who	Presence	Is anyone in the workspace?
	Identity	Who is participating and who is that?
	Authorship	Who is doing that?
What	Action	What are they doing?
	Intention	What goal is that action part of?
	Artifact	What object are they working on?
Where	Location	Where are they working?
	Gaze	Where are they looking?
	View	Where can they see?
	Reach	Where can they reach?

3.1.2 Shared View

Synchronous collaboration usually requires a shared environment in which users (or participants) who join in the same session could share the same view and be aware of other users' activities. The shared view can be classified into three scenarios: strict shared view (WYSIWIS, What You See Is What I See), relaxed shared view (relaxed WYSIWIS) and hybrid shared view (private and public views). Figure 3-2 shows the differences among the three scenarios.

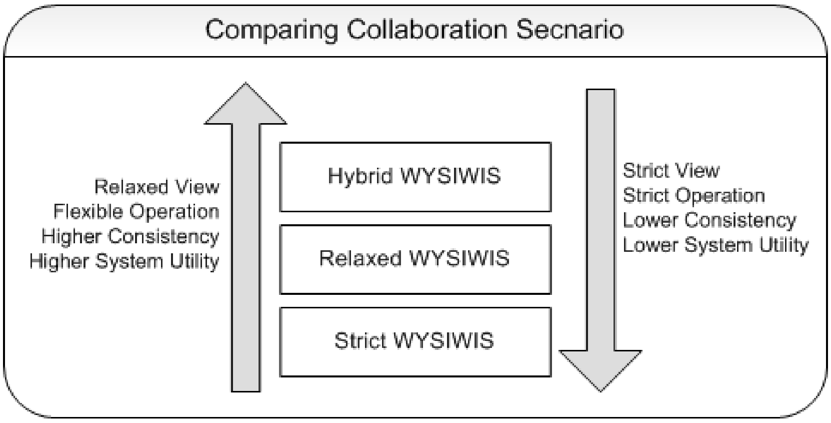


Figure 3-2 Comparing Collaborative Scenarios

In strict WYSIWIS scenario, which can be seen in NetMeeting, all the participants have a strict duplicated view. All the operations including mouse, keyboard and any other inputs on one workspace will be distributed to all other participants' workspace, and therefore will be seen by these participants. All the workspaces keep the same environment states. Because of the strictly-replicated environment, keeping the workspace in consistency is a challenging issue. Some locking methods may be used to solve inconsistency problems. In this scenario, usually only one user can operate the workspace at a time, which causes inefficiency of the utilities of the workspace. In the area of 3D virtual workspace, strict WYSIWIS is not in common due to its inefficiency as different users want to have theirs own 3D views.

In relaxed WYSIWIS scenario, which can be seen in different multi-player video 3D games, participants share the same view environment but with different viewpoints. The collaboration among participants is implemented through so-called avatars, 3D pointers or agents. Although there are different terms, they have the same purpose: these avatars provide their own viewpoints and collaboratively work with each other. For example, through moving and rotating an avatar, the user can adjust the viewpoints and obtain its own views (private view). Further, a complicated avatar can change the contents of a 3D world and even work on a task together with other avatars. This scenario is often used in 3D virtual environment.

Hybrid WYSIWIS, which can be seen in collaborative manufacturing and construction design, is a mix of both strict WYSIWIS and relaxed WYSIWIS. The main method is to provide the participants with both a shared (public) view and a private view at the same time. The public view provides collaboration functions, while private view is used for private tasks. In the private view, all the avatars can be seen as the representatives of the participants. Clicking on one avatar will change the public view to the avatar's view. This scenario is often used in collaborative industry and construction design in which multi-users need not only finish their own part, but also get reference from their partners.

3.1.3 Radar View

Radar view is a popular technique for providing this awareness information in shared representations. The radar view displays an overview map of other participants' area of interest, especially in collaborative GIS. Views on the radar are represented as coloured transparent rectangles and are used to convey awareness information about where the operator is currently working in the map area.

The radar view (Figure 3-3) shows each user's viewpoint and cursor position, the individual sign and light markers are to the right, and a legend appears below the map (Schafer et al. 2003).

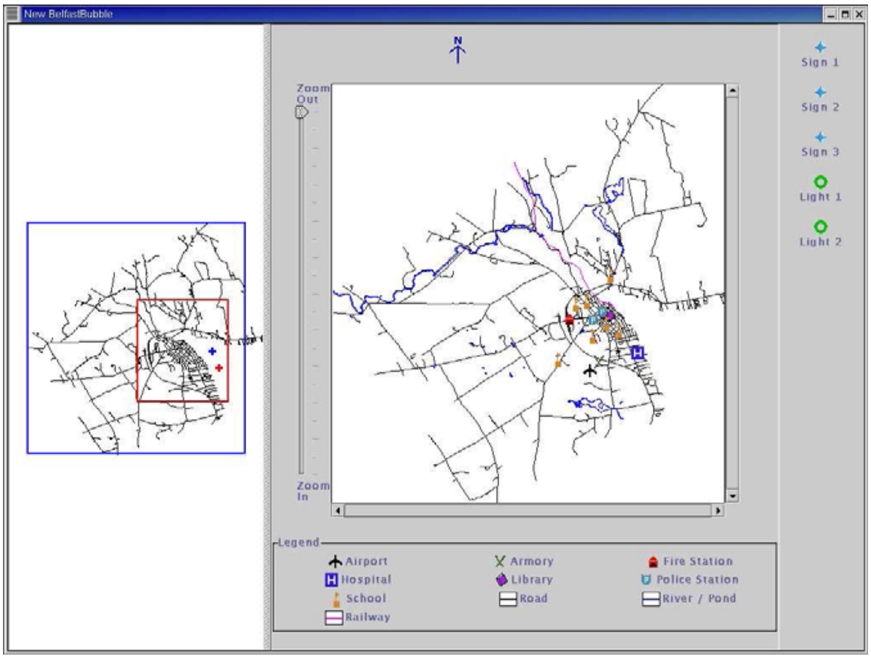


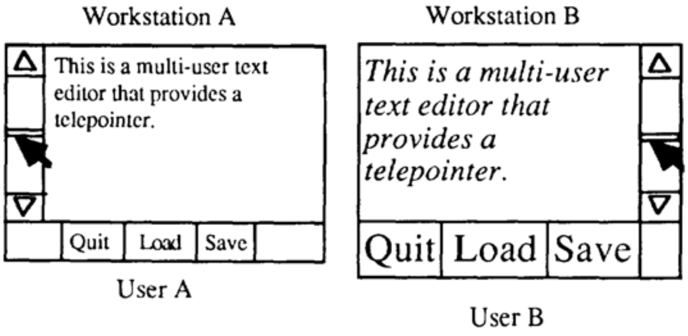
Figure 3-3 Radar view (Schafer et al. 2003)

3.1.4 Telepointer

Telepointers are the presentations of local users' mouse cursors displayed on remote participants' screens in a real-time groupware system. As an important element of groupware interface, telepointers are able to provide a variety of group awareness information including presence, location and activity. Xia, et al. (2005) divided the telepointer into several types: window coordinate-associated, widget-associated, and object-associated.

A window coordinate-associated telepointer is associated with the coordinates in the shared window space. It is usually used in a strict WYSIWIS view mode in which each object is placed at the same position in the shared window, and the same coordinates point to the same object at all sites, so the window coordinates are sufficient for a telepointer to locate any objects in the shared window.

A widget-associated telepointer is associated with identifiers of a window widget, and provided with the relative position inside the widget space. It is usually used in a relaxed WYSIWIS view model. For example, in Smart Telepointer (Rodham and Olsen 1994), the telepointer's reference parameters include a path in the widget tree from the root to the leaf-level widget which contains the telepointer and relative position information within the leaf-level widget space (see Figures 3-4 and 3-5).



This is a multi-user text editor that provides a telepointer.

This is a multi-user text editor that provides a telepointer.

Figure 3-4 Widget-associated telepointers: Smart Telepointer (Rodham and Olsen 1994)

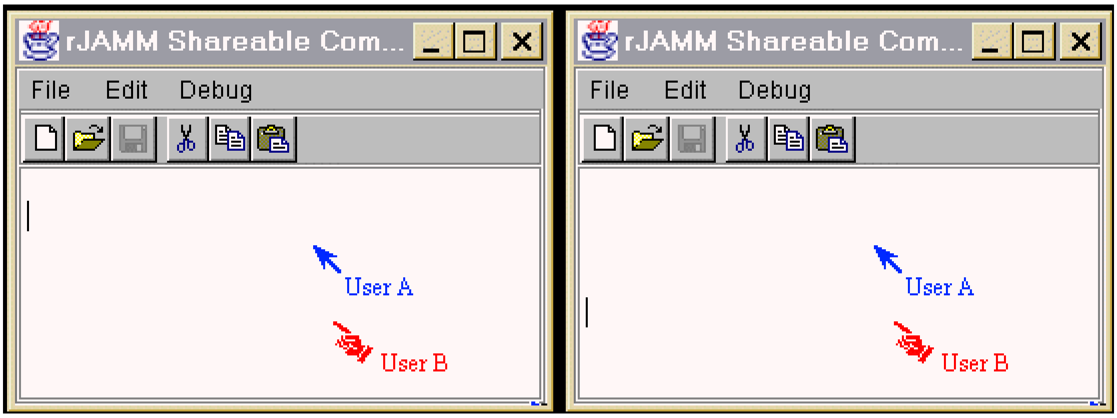


Figure 3-5 Telepointers from Flexible JAMM (Begole et al. 1999)

A object-associated telepointer associates a telepointer with the reference object rather than its position (Xia et al. 2005). It is able to correctly point to the reference object in the face of dynamic content and view changes caused by both concurrent and sequential editing operations. It is usually used for real-time collaborative document editing systems due to dynamic content and view changes (Figure 3-6).

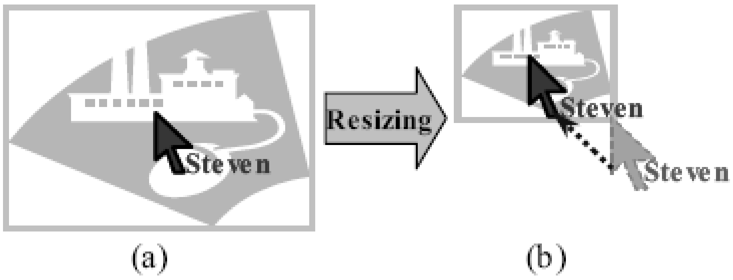


Figure 3-6 Object-associated telepointer shows (a) The initial state; (b) The state after executing a resize operation: the telepointer remains inside the picture (Xia et al. 2005).

3.2 Architecture Considerations

Collaborative systems are typically classified into two types in term of architecture: centralized architecture and replicated architecture. If the rendering component on the client is considered for the graphic related system, the system is classified into thick, median and thin clients. Since SC3DGIS is related to the 3D graphic system and synchronous collaboration, the above issues have to be considered together.

3.2.1 Centralized Architecture vs. Replicated Architecture

When considering collaboration functions among multi-users, system designers often make a choice from centralized, replicated or hybrid architecture to handle distributed environments.

In a centralized architecture, only one instance of application runs on a central server. The server is responsible for controlling all input and output to the distributed end-users. A sequence of events generated by an end user's interaction are collected and sent to the central server. The output of the shared application must be broadcasted to all participating users for visualization and consequent actions (see Figure 3-7).

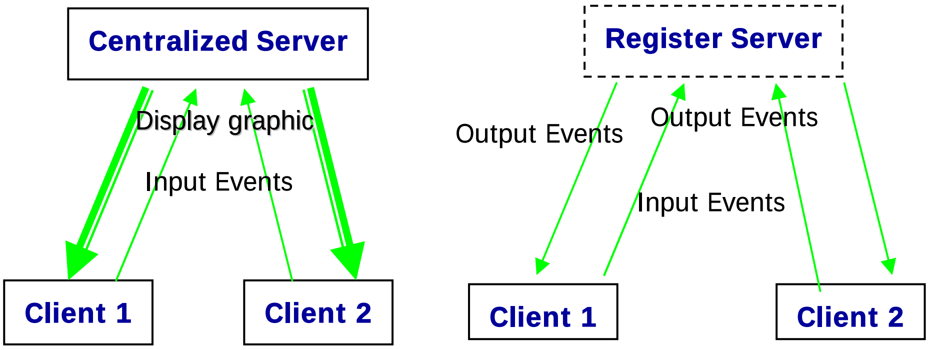


Figure 3-7 Centralized architecture (left) and replicated architecture (right)

Examples of applications based on the centralized architecture are NetMeeting, SharedX (Garfinkel et al. 1994), X TerminalView (XTV) (Begole et al. 1999) and commercial collaborative software such as WebEx and ETeam. An important advantage of a centralized architecture is that it guarantees consistency of shared data which makes the shared application easier. The host of the application can share its application with other participants just through sending its interface images to other participants and other call-back and transformation mechanisms. The disadvantage is that it requires higher bandwidth to distribute displaying information to all end users, strict WYSIWIS interface, less responsive to user input, and less fault tolerance (Suthers 2001). In a replicated architecture, the entire application is installed and run on each client's machine, with some means of synchronization provided between them (Dewan 1999). Both the input events and the graphical outputs are processed locally. As opposed to a centralized architecture, a replicated architecture requires lower bandwidth because output is only locally transmitted. This increases performance and scalability. This method is often used for systems with complex rendering requirements, such as 3D graphics, video games and virtual worlds, in which hundreds of frames are drawn every second. Examples of applications based on this architecture are GroupKit, DistEdit, Counter Striker and Battlefield (series). The disadvantage is the increased complexity in handling data sources. Because there are multiple copies of shared data in a replicated architecture, it is generally expensive to keep the state of shared data replicas synchronized.

Applications are decomposed by many components. Some components may require to be shared while others may require to be replicated according to functionalities. Therefore a hybrid or semi-replicated architecture is introduced by Dewan (1999). Hybrid architecture does not represent a "pure" new category but represent the composition of centralized and replicated architecture.

3.2.2 Thin Client or Thick Client

A generic 3D system primarily consists of 3D data model and a rendering system based on the 3D model. It can be divided into several sub-components: 3D data repository, 3D data transportation, 3D internal data structure, 3D rendering and APIs, and 3D high level user interfaces (see Figure 3-8).

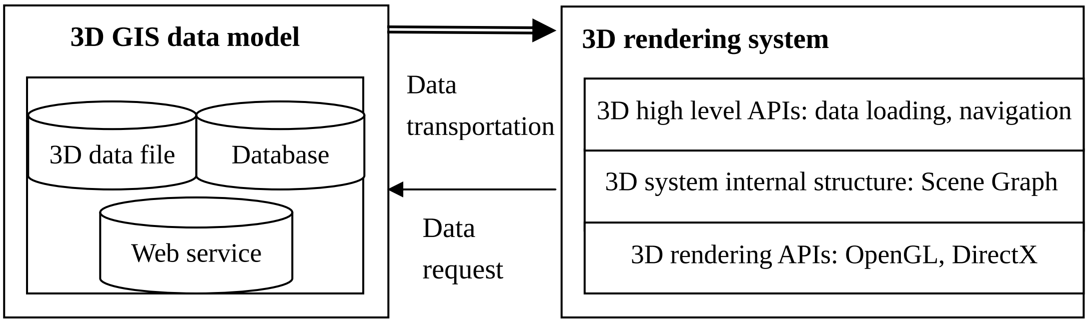

3D GIS data model
3D data file
Database
Web service
Data
transportation
Data
request
3D rendering system
3D high level APIs: data loading, navigation
3D system internal structure: Scene Graph
3D rendering APIs: OpenGL, DirectX

Figure 3-8 Genetic 3D system architecture

3D data repository stores 3D data model outside the 3D rendering system. Usually, there are three approaches to manage 3D data: 3D data files, 3D database and Web service. The data request message is sent from the client to the data repository. 3D data transportation responds to the request and sends the 3D data to the 3D rendering system. The data may be coded with XML and transferred by streams in the Internet environment. After the data streams reach the 3D rendering system, they will be encoded and reconstructed to the original format. Through 3D high level APIs, such as Java3D or Open Scene Graph, 3D data is loaded into a 3D scene graph (compared to an external 3D data model). Lastly, low-level rendering APIs like OpenGL or DirectX will render the data on the output devices. This last step could be transparent for 3D GIS application developers. The developers just need to handle the 3D data with 3D high-level APIs, like Java 3D and Xith3D. Although the developers could handle OpenGL directly and build internal structure for themselves, this method makes the development procedure more complex and expensive.

As far as Web-based visualization is concerned, clients are classified into thick, medium, and thin (see Figure 3-9) according to the embedded render and display element generator schemes. Both render and display element generator schemes may be embedded in the client side. The thin clients just like a simple Web browser showing common images. All the render and other schemes are run on the server side. The medium clients have embedded rendering plug-ins. The advantage of thick clients is that the client is free to realize any schemes including very complex visualization and interaction schemes while the drawback is the need for a special Web browse plug-in which implements these schemes (Altmaier and Kolbe 2004). The decision regarding which client model is appropriate for which application depends on the specific scenario. For example, the highly interactive 3D system which requires frequently and quickly changed views will prefer a thick client model. The commercial 3D systems, like Google Earth 3D and Active Worlds, use the thick client model. Users need to download the client system before running it.

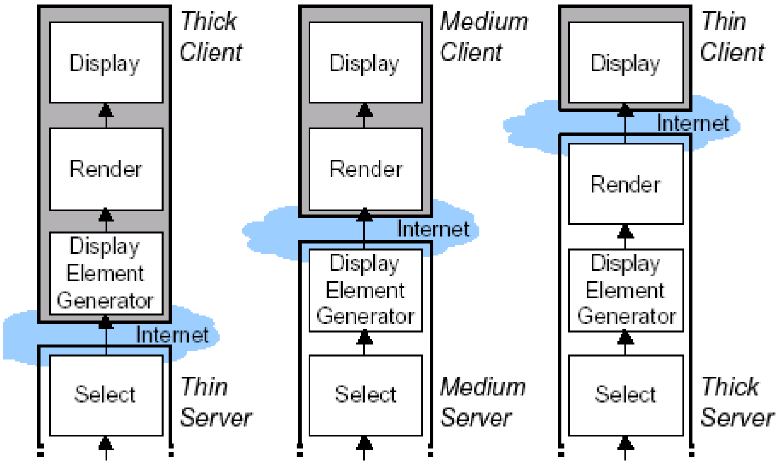


Figure 3-9 Different balancing schemes with visualization tasks between client and server (Altmaier and Kolbe 2004)

3.3 Consistency

The synchronous collaborative system that contains a shared workspace function will maintain some kind of consistency between two or more clients. Usually, there are two strategies to deal with inconsistencies, avoiding inconsistency or allowing its emergence. From a technological perspective, strategies to avoid inconsistency generally increase response and notification time, and reduce the availability of the client for actions, compared to strategies that simply allow for inconsistency (Hofte 1998).

There are two ways for groupware systems to avoid inconsistency: the ordering method and the locking method. The ordering method is a method to avoid inconsistencies through accepting actions that may cause inconsistency but postponing their execution to a moment that will not cause the emergence of inconsistency. When users take actions, these actions always follow the same ordering. The locking method does not accept or deny actions that may cause inconsistency.

Floor Control and Latecoming are special terms in collaborative system to keep system in consistency at a user's interface level. Floor control refers to the management of interaction among participants in meetings. Myers et al. (2001) gives a classification to distinguish the different floor control polices based on the procedure of obtaining a floor control: assigning control, requesting control, and releasing control. By combining these release and request mechanisms, all of the existing floor control policies can be constructed. The term latecoming is used to denote a process which allows latecomers to join and participate in an ongoing session. Although there are many mechanisms to handle floor control and latercoming, the best fit mechanism relies highly on the user scenarios and the work procedures.

3.4 Multicast Protocol and Network

With today's technology it is possible to afford the "cost" of making a unicast connection with everyone who wants to see your Web page. However, if audio and video streams are to be sent, both of which require a huge amount of bandwidth, to hundreds or perhaps thousands of recipients through establishing a separate unicast connection with each of the recipients, the sending computer and the network would collapse.

Collaborative application, therefore, requires point-to-multipoint and multipoint-to-multipoint communication, i.e., multicast communication. A number of multicast protocols, which range from large scale network infrastructure such as Distributed Interactive Simulation (DIS) to simulated multipoint protocols based on TCP/IP such as Multicast Backbone (MBONE), have been developed, which are all application dependent. Most of the collaborative systems have their own network to support multicast protocols, such as conferencing system, WebEx with MediaTone Network, and the groupware tool, groupKit. Fortunately, there is always an abstract layer above the transportation and network layer to provide an interface for the specific CSCW application so that the application can be independent from network protocol and also can be transparent for the application developers.

Chapter 4 Design of Conceptual Framework

The conceptual framework in this research is used to investigate main aspects which play core roles to reflect and identify the features of SC3DGIS. The conceptual framework is similar to the user requirement analysis. However, the main goal of the former is to identify and delineate important technological and social parameters, which should be considered to extend, or reinvent, geoinformation technologies to support small group work, while the requirement analysis focuses on more detailed users' requirements and behaviours analysis based on specific use cases. Since there is not currently a SC3DGIS in existence, a virtual group meeting scenario is simulated according to a mimic synchronous collaborative 3D environment, in which the geographically dispersed people can be aware of others' work, share a 3D view, discuss a geography-related issue and work on a common task. Based on this study, the general directions for the research questions in Section 2.1 will be identified.

4.1 Collaboration in Group Meeting

Site selection in urban planning has typical scenarios, group meetings, to involve different people from different places working on a common goal. A group meeting for the site selection plays an important role in the exchange of opinions, discussion of solutions and making of decisions. In this research, a group meeting is used to analyze how the people collaboratively work together in a shared work place referred to as a meeting room. The minutes of the group meeting are investigated to figure out the meeting's processes, rules and people's activities.

Because the minutes did not describe any collaboration in the 3D GIS environment, the simulation for a virtual group meeting in a shared 3D GIS environment is developed based on the processes, rules and people's activities in the group meeting for site selection. Further, this virtual simulation and the prototype are also designed for geological scientists in the GEOIDE Project.

4.1.1 Group Meeting Scenario - Site Selection

The investigation of alternative site selection for facilities, such as airports, highways, and heavy industry, is a complex task that involves cooperation among scientists, stakeholders and people from government agencies with complementary expertises. A typical investigation comprises a thorough examination of all pertinent factors, such as existing and foreseen land use planning, characteristics of wind and weather conditions, operational and safety aspects, interference with the existing road network, earthworks, construction and expropriation costs, as well as environmental and socio-economic impacts.

Context of site selection

This case study is based on the Wood River Region Airport Site Selection and Feasibility Study (Friedman Memorial Airport 2006). The study used forecasts of aviation demand developed from data through year 2002, as presented in the 2004 Master Plan Study, to identify the minimum facility requirements for an airport at a new site (see Figure 4-1). The primary minimum requirements for a new airport which guided the study are: compliance with all applicable Federal Aviation Administration (FAA) design and safety standards;

ability to provide reliable all-weather service via an Instrument Landing System (ILS); be able to accommodate current known aviation demand as well as offer the flexibility to accommodate future demands as they arise, for decades.

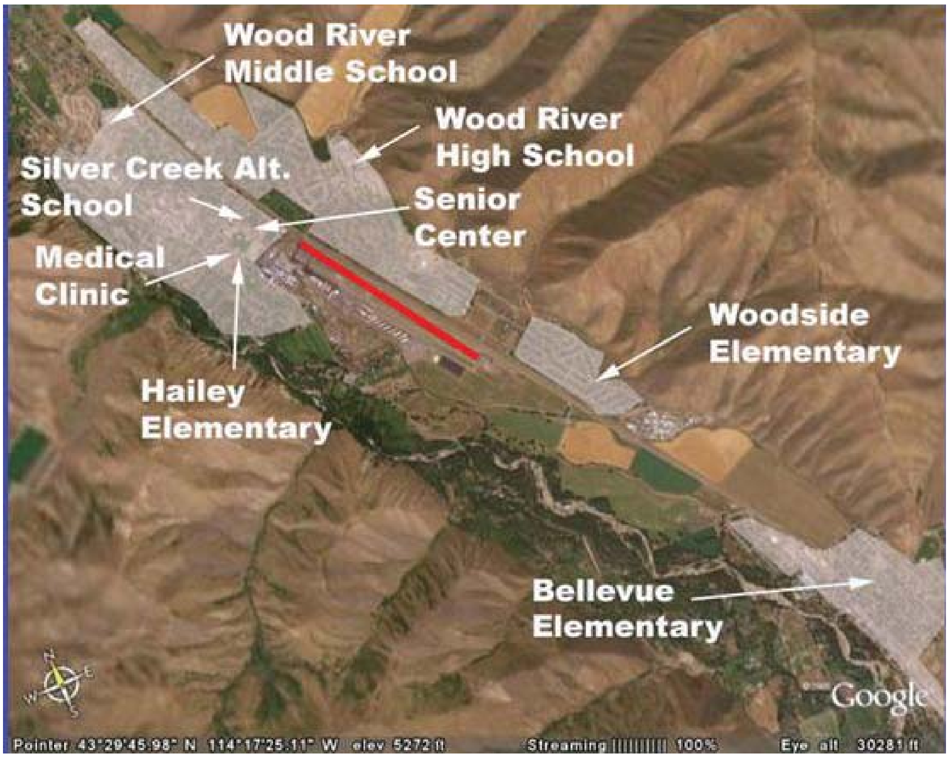


Figure 4-1 One location alternative for a new airport (Friedman Memorial Airport 2006)

Site selection process

The site selection process involves a series of data collection and evaluation activities that becomes more specific with each subsequent step of the site selection process. The process usually involves the following steps: 1) confirm readiness, 2) develop the work plan, 3) conduct search for sites, 4) evaluate long list, and 5) evaluate short list/recommend site(s) (GSA USA 2007). Figure 4-2 shows the detailed workflow of the process.

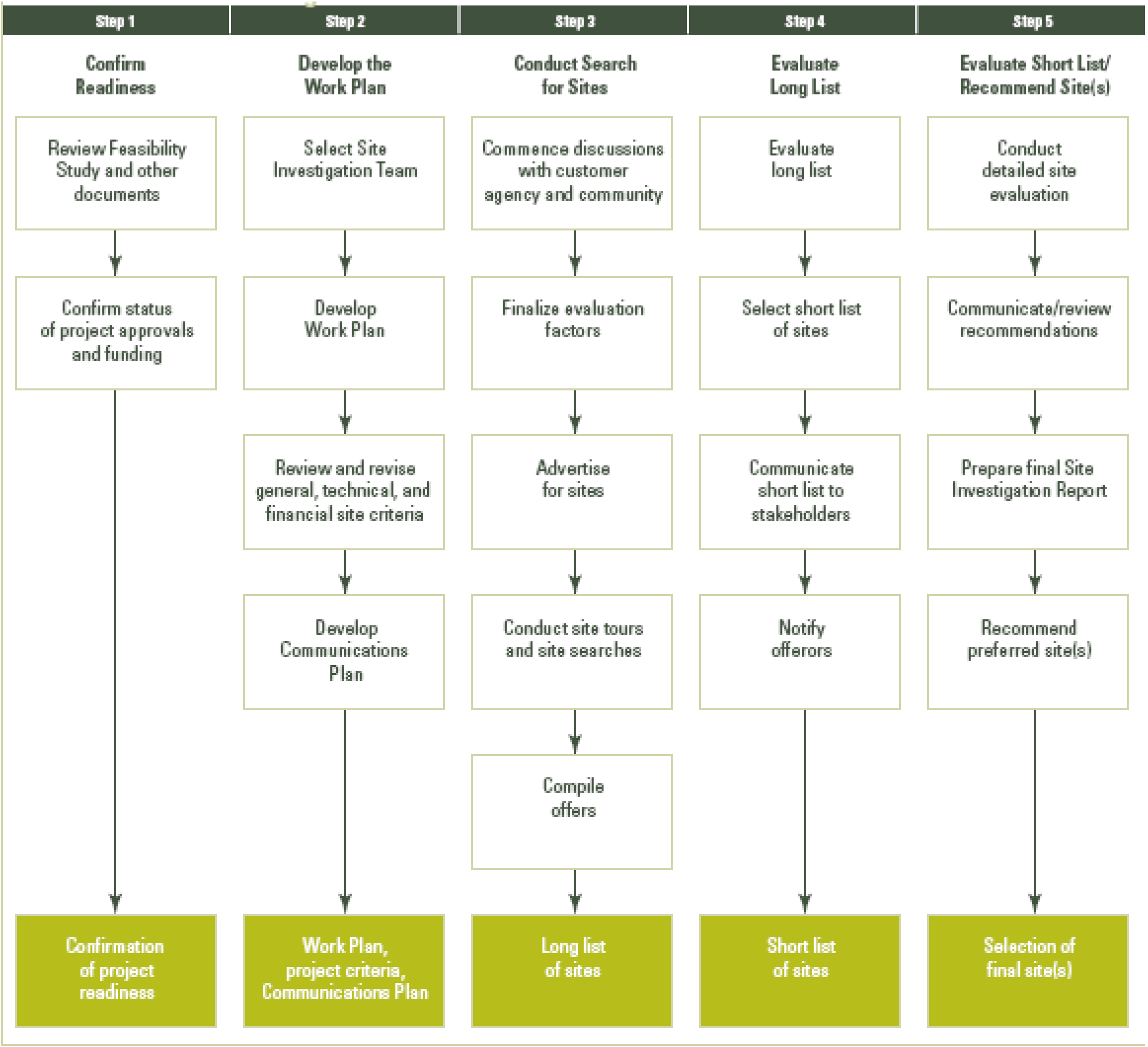

Step 1
Confirm Readiness
Review Feasibility Study and other documents
Confirm status of project approvals and funding
Confirmation of project readiness
Step 2
Develop the Work Plan
Select Site Investigation Team
Develop Work Plan
Review and revise general, technical, and financial site criteria
Develop Communications Plan
Work Plan, project criteria, Communications Plan
Step 3
Conduct Search for Sites
Commence discussions with customer agency and community
Finalize evaluation factors
Advertise for sites
Conduct site tours and site searches
Compile offers
Long list of sites
Step 4
Evaluate Long List
Evaluate long list
Select short list of sites
Communicate short list to stakeholders
Notify offerors
Short list of sites
Step 5
Evaluate Short List/ Recommend Site(s)
Conduct detailed site evaluation
Communicate/review recommendations
Prepare final Site Investigation Report
Recommend preferred site(s)
Selection of final site(s)

Figure 4-2 Site selection process (GSA USA 2007)

The study identified 16 potential sites within a reasonable (by industry standards) geographic proximity to the primary service area. A number of critical evaluation criteria were applied to the initial candidate sites in order to screen the list to a preferred short list for more detailed evaluation. Three candidate sites, one in Blaine County, one in Lincoln County, and one in Camas County, were selected for detailed, comparative evaluation. In addition to the involvement of an Airport Site Selection Advisory Committee in the screening and selection process, the FMAA Board sought and received substantial input from the interested public. This public input, along with technical documentation, was used by the FMAA Board in arriving at a preferred location for a new airport, which is a site located along State Highway (SH) 75 in Southern Blaine County.

Participant and stakeholder involvement

Stakeholder involvement was a significant element of this planning process. The involvement process included a 25-member Site Selection Advisory Committee, public information workshops, presentations at FMAA meetings, project documents available in public places, direct presentations by FMAA and staff members to government entities and special interest groups, and the development of a Web page to provide easy access to information and to give and receive public input. It is the firm belief of the FMAA that thorough and detailed dissemination of information with ample opportunity for public input is critical to the success of planning processes.

Group meeting process

The group meeting is often held in nearly every step (see Figure 4-2) when some decisions need to be made such as establishing site selection criteria (Step 2), commencing discussions with customer agency and community (Step 3), selecting short list (Step4) and conducting detailed site evaluation (Step 5). Usually there are two kinds of group meetings in the process: regular meetings for internal and invited people, and special meetings for public hearing. Appendix 1 shows the minutes and transcripts of a regular meeting, held on July 7th 2005. The public hearing meeting of the Friedman Memorial Airport Authority held on September 28th, 2005 can be seen in Wood River Region Airport Site Selection and

Feasibility Study.

A group meeting in urban planning is a typical use case of synchronous collaboration. In a regular meeting, for example, participants will communicate with each other, share common work context towards the same understanding on a specific topic, and finally make a decision. The participants may assume different roles. The chair of the meeting is in charge of the meeting and coordinates the speakers' activities. The speakers are in control of the microphone and demo equipment. The audience just watches the meeting and may provide the feedback. A group meeting is also the most frequently used methods for decision makers.

The whole process of a group meeting is usually as follows. First, the chair will provide the agenda of the meeting for approval. The agenda could be a proposal of the alternatives and related criteria of the site selection. Related materials such as reports, images, and proposals will be sent to all the participants. The previous minutes will be approved, if any exists. The issues listed in the agenda will be discussed one by one. During this meeting, the participants can present their opinions and demonstrate their ideas through tools, such as PowerPoint or GIS tools. Some scoring process may be processed to screen the criteria and alternatives. The decision can be made after the screening process.

4.1.2 Virtual 3D GIS Environment Simulation

Based on the process of such a group meeting, a mock-up group meeting is simulated under a collaborative and distributed 3D GIS environment. In this mock-up meeting, the participants will discuss the alternatives and criteria, and screen the criteria. These participants are geographically dispersed parties, such as experts in specific fields, stakeholders and government agencies. They can be aware of others activities and work on the same goal through accessing the collaborative 3D GIS environment. The detailed mock-up group meeting includes the following steps:

- Setting up shared 3D GIS Environment
  1) Register session
  2) Set up shared 3D GIS view
  3) Set up floor controls
  4) Set up shared data source
- Joining the environment
  1) Join the session and shared view
  2) Accept the floor rules
  3) Download shared data
- Approving Agenda (Topics)
  1) Chair summits agenda to all the participants for approval and gives detailed introduction to the agenda.
  2) Participants may give comments to the agenda.
  3) Chair may accept the comments and make a change to the agenda, and submit the agenda to everyone for approval again.
  4) Repeat Step 2 and 3, until the agenda is approved.
  5) Participants approve the agenda.

- Discussing and demonstrating in the 3D environment
    1) There are two mechanisms which manage the participant’s discussion: presentation style and discussion style.
    2) In the presentation style, the Chair gives the participant the right to present, show 3D GIS view and upload data to everyone. For example, upon a participant’s request for presentation, the chair allows the request. The participant obtains the right to control the microphone, shared 3D GIS view and video channel, and starts to present to all the participants within a limited time.
    3) In the discussion style, instead of being assigned the right by the Chair, every participant has an equal chance to give a presentation if he/she can “catch” the right to control the floor. After one presenter finishes his/her work, the first one who catches the right can present.
    4) The participants’ opinions may be adopted and become part of the decisions.
    5) A voting system may be used to screen the alternatives and criteria (Optional).
    6) Close the meeting

The above mock-up group meeting process seems very simple and straightforward. However, this kind of system is different from traditional client-server or browse-server system in the following aspects:

- Participants’ finding and registration of interesting topics.
- Workflow mechanism. Workflow is used to plan, assign and implement the tasks. The workflow is a sort of asynchronous collaboration.

- Negotiation mechanism. Negotiation is needed to handle discussion in the decision making process.
- Multiple users' awareness in goal, tasks and behaviours. Users can share not only the data, but also the operations and 3D GIS views supporting what-you-see-is-what-I-see (WYSIWIS) or relaxed WYSIWIS.
- Multicast-based message transportation routines. For example, in the agenda approval step or issue discussion step, the clients need a real-time message exchange process. The agenda or 3D view operations need to be transported to all the clients and the clients' revision about the agenda need to be transported to Chair and other clients. It is a multicast transportation routine which differs from routine between the client and the server.

Moreover, in this simulation, multiple users' discussion and collaboration are based on a shared 3D GIS environment. A presenter usually presents his/her opinion and argument through demonstrating and operating the 3D model and environment for other participants. In order to understand the presenter's opinion and context, all the participants should have the same viewpoint to the 3D model and share the operation of the 3D environment. This shared 3D GIS environment also needs the shared data sources and tools to coordinate the participants' behaviours and operations so that the multiple users' operations to the shared 3D GIS environment will not be interrupted.

Realization of the above mimic-simulation relates not only to a technical issue but also to social issues, such as people's collaboration. The next section will present a conceptual framework to guide and scope the related issues.

4.2 Conceptual Framework of SC3DGIS

Based on the above scenario, the conceptual framework of SC3DGIS can be divided into three aspects: social, GIS and technological (see Figure 4-3). The social aspect focuses on collaboration and cooperation among users. The 3D GIS aspect relates to the special criteria derived from 3D GIS requirements, which differentiates SC3DGIS from other collaborative systems or groupware systems. The technical aspect gives more attention to the main technological challenges to implement the SC3DGIS.

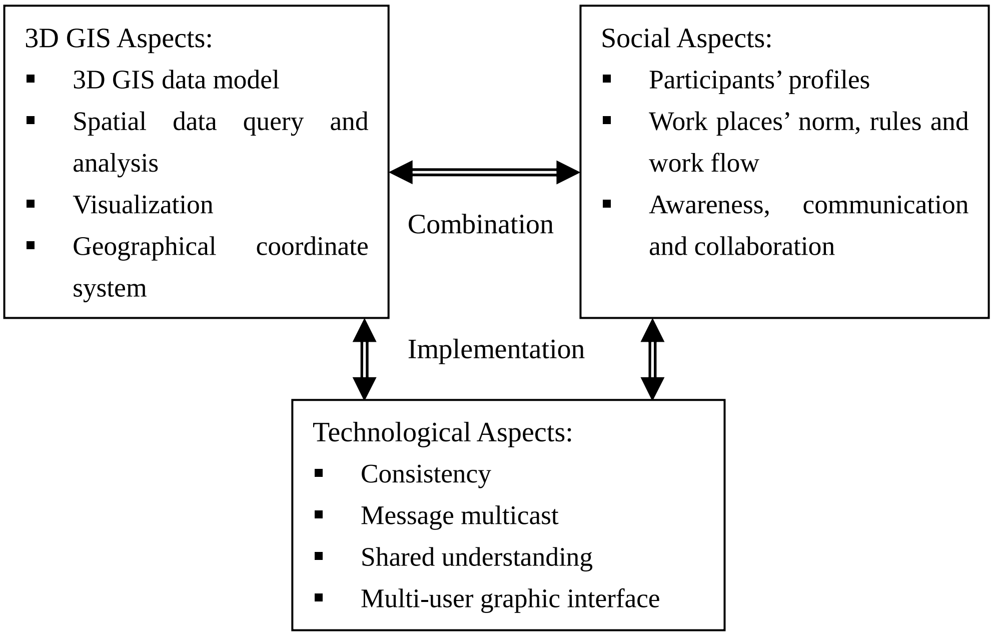


Figure 4-3 Conceptual Framework in SC3DGIS

4.2.1 Social Aspect

The social aspect relates to how the participants work with other participants. The core elements may include:

- Participant’s profile, role and relations with each other
- Workplace’s norm, rules and operation protocols
- Participant’s awareness, communication and interaction with each other
- Workflow and negotiation processes

A participant’s profile usually presents the user’s name, location, title, origination and role. Different roles have different privileges to access and operate the system. The workplace is also an important factor for users to work together. Users will follow the same norm, rules and work protocols in the same workplace.

Awareness, communication and interaction present three levels of collaboration. According to Section 3.1.1, awareness involves the understanding of another person’s interaction with the shared workspace. The typical functionalities include shared participants’ profiles and shared environment, such as shared view, radar view and telepointer (Section 3.1). Communication involves the information exchange between participants. The information may include events, messages, text chat, radio and video conferencing. Interaction involves not only information exchange but also decision-making and negotiation processes. This level of collaboration is more complex than awareness and communication since this kind of process involves multiple information exchange and shared understanding.

4.2.2 3D GIS Aspect

The 3D GIS aspect specifies the special features in SC3DGIS. There features such as collaborative virtual environments, multiplayer games and virtual worlds differentiate SC3DGIS from other groupware systems. The main features are:

- 3D Spatial data model in the geology field
- Spatial data query and analysis
- Cartographic visualization
- Geographical coordinate system

The spatial data model is the main difference between GIS and other systems. This kind of data model describes both location information and characteristics of spatial attributes such as roads, land parcels, and vegetation stands on the earth's surface. In order to specify the special characteristics of GIS, several core elements, such as a geographic coordinate system, geometry, attribute, topology relations, and styles, are identified. These elements can be presented into one data model, like the GIS **theme model**, which is also called the **Feature model** in GML. This kind of model is used to present layer-based themes of the GIS map (see Chapter 6 for further discussions).

Spatial data query and analysis include some typical GIS algorithms, for example, the shortest path analysis, buffer query and utility networking analysis. This query and analysis is based on the geometry, topology and attributes information from the spatial data model.

The special characteristics of GIS in visualization is about cartographic symbolization, which describes how mapmakers display a spatial feature in a map or digital map with cartographic issues (map elements) such as symbol, color, text and type of map, title and scale. Rather than integrating these elements into GIS data mode, current GIS platforms such as ArcGIS configure them on the fly within the system when the data model is loaded. However, the same content could be visualised in different ways according to a user's habit and the purpose of the map theme. The method separates the visualization from its content and causes problems for SC3DGIS because all users' visualizations need to be in consistency.

A geographical coordinate system, such as that found in traditional GIS, is also a core element for SC3DGIS. Although other systems like collaborative virtual environments and video games have coordinate systems, SC3DGIS and the supported data model are based on an Earth-oriented spatial reference system.

4.2.3 Technological Aspect

The technical aspect provides technological support or tools to implement the social aspects and GIS requirements. These requirements, such as users' awareness, communication and interaction, result in technology differences compared to traditional desktop, client-server or browse-server systems, such as OpenLayers and Mapserver. The core challenges for SC3DGIS are:

- Consistency

- Message multicasting
- Shared understanding
- Multi-user graphic interface

The consistency issue is about how to keep the distributed clients in consistency. Usually the geographically dispersed users work on their own distributed clients. There are potential conflicts when the users operate the system in their clients. A mechanism such as floor control is needed to coordinate potential conflicts.

Message multicasting is also required to handle client-to-client message transferring. This direct communication between clients is different from typical client-server architecture. The developers for SC3DGIS have to give more consideration to the network layer design, which is transparent for traditional client-server system development.

A shared understanding issue is about how to make all the parties, such as the messages, different clients and their behaviours, and data sources, understandable to each other. This is the foundation for collaboration in a distributed system.

A multi-user graphic user interface (GUI) is also a challenge. Development tools such as C++ or Java are not designed for collaboration in mind. Special technological efforts have to be made to design multi-user GUI for 3D GIS purposes.

4.3 Summary

The conceptual framework is presented to describe the social, GIS functional and technological aspects which play core roles to reflect and identify the features of SC3DGIS. A comprehensive understanding of the special requirements in architecture, technology and data model is obtained and presented.

SC3DGIS needs to meet two fundamental requirements: 1) providing a Web-enabled environment and real-time collaboration capability, and 2) satisfying a set of 3D GIS requirements. In the first type of requirement, the data is accessed and transmitted over the Internet. Multiple users can share the 3D view, model, data source and directory. Multiple users are also synchronously aware of the existence of others through collaborative tools such as participant list, radar views, telepointers, multiuser scrollbars and aviators. At the same time, complex communication and interaction are required to carry out tasks. The second type of requirement includes a 3D GIS data model for SC3DGIS and some 3D GIS functions necessary for collaborative visualization and exploration of 3D data represented based on the 3D model. The 3D GIS data model not only requires GIS content representation, such as themes, 3D model objects, attributes, geometry and primitives, but also state representation, such as model styles and state. The latter is used for collaboration purposes, so that the distributed multiple users have the same state of 3D model and maintain the system consistency.

Moreover, the distributed parties including participants, collaborative systems and shared data need to be understandable to each other. The shared understanding of the knowledge needs to be presented by a common understandable format. In summary, the SC3DGIS should have the following features:

1. An **Internet-enabled 3D system environment**. Users must have Internet access to the 3D system and also have the capability to upload/load their own 3D data through the Internet.
2. An environment that supports **multi-user and real-time collaboration**. Multiple users can work together and carry out a common task in a 3D environment. Specifically, multiple users can share the 3D view, model, data source and be aware of each other's operations and behaviours. Multiple users are synchronously aware of each others' existence through collaborative tools such as participant list, radar views, telepointers, multiuser scrollbars and aviators. Multiple users' communication and complex interactions are also supported.
3. The environment is **well structured** such that participants/users can work together on planned tasks, make negotiations and deals, and approve proposals with respect to related rules and roles. Complex interaction among participants might be required in this situation. The structured environment is based on the collaborative capabilities of the system.
4. This environment is also a **GIS-oriented** system in which all kinds of 3D GIS-related requirements are needed. The core is the GIS-based 3D data model, spatial data query and analysis. The GIS user habits and work patterns are also concerned.

The next chapter (Chapter 5) will provide a more detailed investigation into the social aspects of how the distributed users work together through collaborative features. Chapter 6 will discuss the 3D GIS data model used for SC3DGIS. Following these two chapters, the technological aspect will be addressed through designing and implementing the prototype of SC3DGIS.

Chapter 5 Behaviours Analysis and Social Collaboration Model

Users will invariably face difficulties in understanding each other when they work together in a geographically dispersed manner. These difficulties mainly arise from a lack of shared awareness and understanding of each other. Thus, the distributed environment, work context and user behaviours need to be understandable and shareable by all the parties. This chapter will analyze and describe the behaviours of all the parties such as users, system and even data based on use cases. These behaviours are then described in the Social Collaboration Model (SCM) and presented as an ontology for shared understanding.

5.1 Behaviours Analysis from Group Meeting Case Study

The users' behaviours are obtained and described in the following use-case methods. A use case is a description of a system's behaviour as it responds to a request that originates from outside of that system. The use-case technique is used in software and system engineering to capture the functional requirements of a system. In this research, the use cases mainly relate to the collaborative functions within a shared 3D view based on a group meeting case study. The typical use cases based on conceptual proceedings in the mock-up group meeting scenario are presented as follows. More detailed use cases presented by UML can be found in Appendix 2.

Search and registration use case (Set environments)

1. Chair sets up group meeting topics
2. Participants join in the group meeting according to the topics

3. Chair sets up the group meeting environments (Floor Control, 3D environments, data sources, and agenda)
4. Participants join the environments and accept the agenda

Discussing and approving issues use case (Negotiation process)

1. Chair sends issues, for example the agenda, to every participant for approval. The discussion process will follow specific interaction protocols such as FIPA-Request and FIPA-Contract-Net.
2. Participant responds to the agenda by sending a not-understood or a refuse message to achieve the rational effect of the communicative act, or also an agree message to communicate the agreement to perform (possibly in the future) the communicative act.
3. Participant performs the action and, finally, must respond with an inform message of the result of the action (eventually just that the action has been done) or with a failure if something goes wrong.
4. Chair receives all the messages and decides if the agenda is approved.

Sharing operations in 3D GIS view use case (Shared 3D GIS view, shared operations)

1. The chair gives the 3D view operation privilege to one of the participants according to the floor control mechanism (presentation style, for example)

2. This participant interacts with the 3D GIS view (for example, navigate the 3D GIS view, adds data source and makes annotations on the 3D GIS view) to make his/her presentation or argument.
3. All the participants who register on the topics have the same operations on their environments.
4. After a certain time elapses, for example, five minutes, this participant has to end his demonstration.
5. Any participant can apply for the operation privilege.
6. The chair then assigns the right to a participant who is the first one to apply for this privilege.

Handling shared data source use case (Shared data source)

1. The meeting chair sets up the shared data sources, for example by uploading data files to the shared system or HTTP servers and configuring shared Web services.
2. Other participants update the data source so that the data files uploaded from the chair can be downloaded into local hard drives.
3. One participant may share/upload his own data to the system.
4. Chair may delete the shared data source.
5. The participant who has shared his data may delete the shared data.

Compositing and automating tasks (For latecomers)

All the operations and tasks performed by the participants will be recorded in a log file using encoded format such as XML. The latecomers are the participants who are late for the group meeting.

1. The latecomer applies to registration (**Finding and registration use case**) to the group meeting session.
2. The Chair sends the latecomer current environments such as, all participants' profiles, session information, 3D states, the log file recording previous operations and discussed issues in the group meeting.
3. The latecomer receives the environment state, composites the previous operations, and automates these operations to the current environment.

5.2 Social Collaboration Model

The social collaboration model (SCM) is a schematic/conceptual description of a set of elements and behaviours amongst users, systems and data based on behaviours analysis and a group meeting scenario. The main purpose of SCM is to provide a shared understanding among users, systems and data for the SC3DGIS environment.

Since the concept of SCM in this research is borrowed from some social models such as virtual organization (VO) and social networking, a review of these models is presented first. The design of SCM is discussed afterward, followed by an application ontology presentation.

5.2.1 Observations of Social Models

A social model is borrowed from social science to solve social-related problems. Virtual organization (VO) and social networking service (SNS) are two related areas.

A virtual organisation (VO) is defined as a geographically distributed organization whose members are bound by a long-term common interest or goal, and they communicate and coordinate their work through information technology (Ahuja 1998). Social models appeared and focused on different aspects of VO. Tjortjis et al. (2002) gave a review of some social models such as “models of virtuality”, “VO life cycle model” and “dynamic decision style” and proposed their own one for Distributed Software Maintenance Teams (DSMT). These models tried to describe specific type of social organization through limited variables such as structure, communication, processes and lifespan. However, these models are usually limited to an email-based communication structure, are too generic, and lack technical detail.

Social network sites (SNS) mentioned in Section 2.4.2 offer social networking functionalities including a network of friends’ listings, person surfing and private messaging. There are some abstract social models found to describe the social networking community. For example, Mika (2005) partitioned the social networks model into three disjoint sets corresponding to the set of actors (users), the set of concepts (tags, keywords) and the set of objects annotated (bookmarks and photos) with a tripartite graph. Zhdanova et al. (2007) used the same model to measure the closeness between entities within Web communities and SNS. Some ontology-based models describe users, such as VCard (Internet Mail

Consortium 2008), Friend of a Friend (FOAF) (Brickley 2008), and resources (documents and photos), such as Rich Site Summary (RSS) (Attitude Group 2008). Although these models are simple, they provide clues to solve problems such as shared understanding. For example, ontology and semantics technology are used to solve shared understanding and boundary breaking through Web Services. A multi-agent approach is used to implement these solutions to change Web services into semantics Web service (SWS) (Bryson et al. 2003).

Compared to a social networking model, which describes the static contents such as “what are the elements”, the SCM model helps the distributed users share a common understanding not only about “who they are” but also “what they are doing” and “how they work together”.

5.2.2 SCM: Relations and Elements

SCM is an abstracted presentation of the interactions and elements for participants to work together on a 3D environment in a mimic small group meeting. Figure 5-1 shows a diagram to describe the main elements and relations of two geographically dispersed participants who collaborate with each other in a group meeting. More participants can be applied into the same pattern.

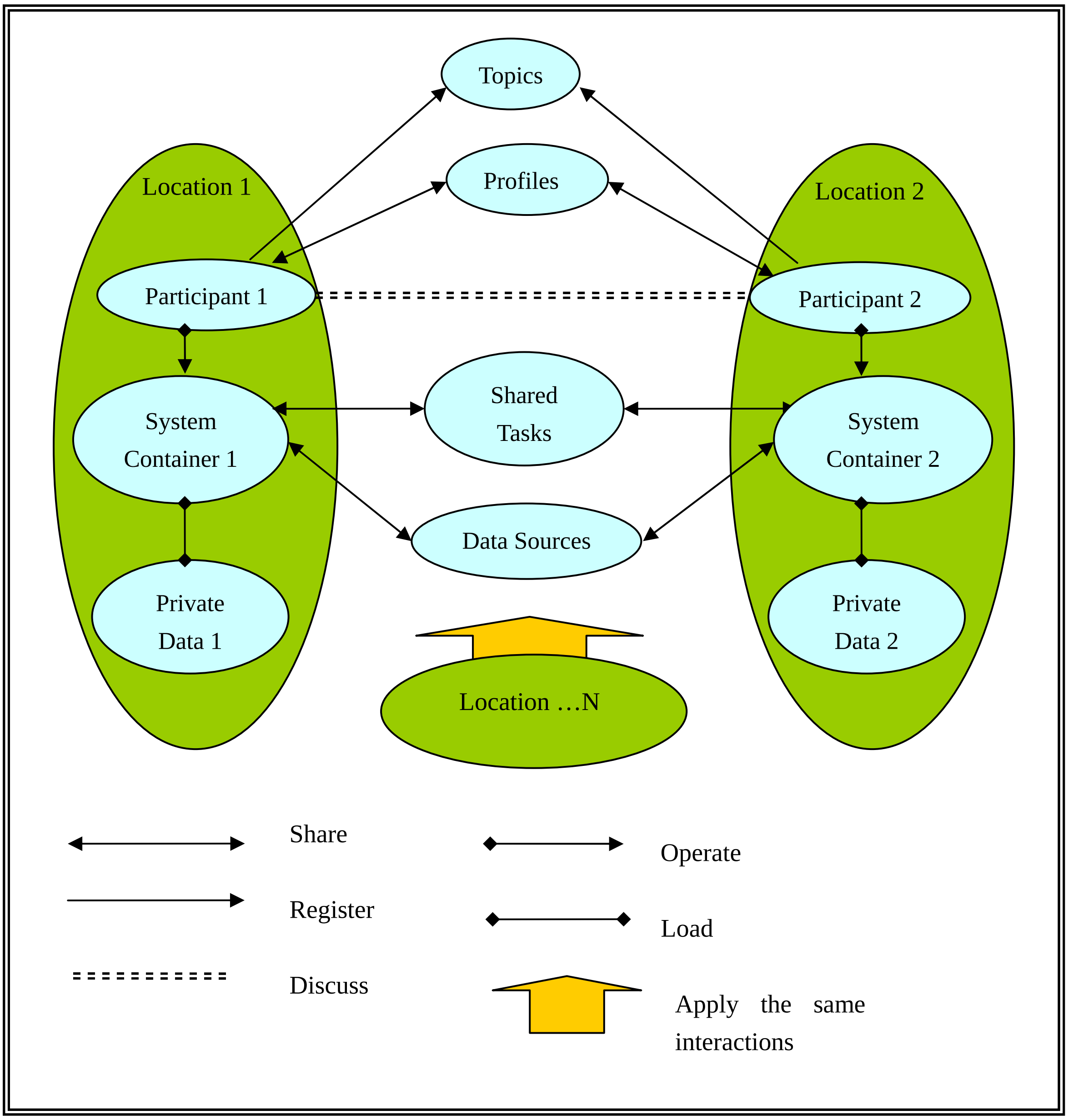


Figure 5-1 Elements and relations diagram for Social Collaborative Model

According to the diagram, participants, systems, and data sources play different roles to carry out the collaborative tasks. For example, when one participant operates the 3D system, such as zooming in/out or rotating the 3D model, his/her operations are shared with other participants. These operations are called shared tasks. In order to avoid the conflicts of multiple participants' operations to the system, different participants may be granted different privileges or roles, called profiles. Participants also need to upload the data source for sharing, which is called the shared data source. Some complex communications, such as making deals and discussions, are also required among participants.

Based on the above description, there are five main elements in SCM: **Topic, Participant, Systems Container, Shared task and Data source**. As well, every element has two attributes: Action and Property. Action defines “what they are doing” and property defines “who they are”. The detailed elements are described as follows:

Topic identifies the topic that is discussed by the participants in a group meeting. Participants can search for a topic and join a group meeting session through the topic identification. All the activities will relate to the topic.

Participant represents the people who find the topic interesting and attend the group meeting. Participants play different roles and therefore have different privileges to operate a shared system/service container. The participants’ roles and privileges also present the structure of the organization. The collaborated work load can be carried out through this structure.

System Container presents a place where a system platform or service is contained. In this container, participants can share the same operation and rendering model, even the whole system platform. The system platform, for example, could be a 3DGIS environment, Web-based GIS service or a GIS application.

Shared Task describes the tasks performed and shared by the participants. These tasks could be a simple operation or a set of operations to the system. Although other elements have their own tasks (private tasks), shared tasks are specified as the tasks known and shared for all participants. Private tasks can be changed to shared tasks, if necessary. For example, when a task is required to be used by other participants, it will become a shared task.

Data source describes the data format, database connection information and the locations of the data sources when the data need to be loaded in the system. The data source includes a private data source and a shared data source. The private data source can be uploaded and becomes the shared data source so that every participant can use it. Table 5-1 describes the main properties and actions of every element.

Table 5-1 SCM elements and the related properties and actions

Elements	Property	Action	Comments
Topic	Topic Name	Create topic Delete topic	It could be an identification to group meetings. People who are interested in this topic will attend this meeting.
Participant	Profiles	Register topic Deregister topic Discuss with other participants	The profile includes register name, title, role and access right. Role (Chair, Presenter, Audience) Access Right (Operation, Mediate, Private and Public)
System container	System Identity	Navigate 3D view (Zoom in, Zoom out, Rotate and Pan) Query functions Analysis functions	Only presents shared tasks.
Shared Task	Task Name, Parameters and Owners	Similar with System containers and data source	
Data source	Data Sources Identity Format or Metadata	Load data Remove data Receive data from remote peers Send data to remote peers	Includes private data source and shared data source.

5.3 Ontology Presentation

Ontology is an explicit specification of a conceptualization (Gruber 1993). Specifically, ontology is a formal explicit description of concepts in a domain of discourse (concepts or classes), properties (slots or roles) and restrictions on properties. Jasper and Uschold (1999) identified three major areas of uses for ontology: 1) to assist in communication between human beings, 2) to achieve interoperability (communication) among software systems, and 3) to improve the design and the quality of software systems.

According to different purposes, ontologies are also classified into difference levels such as top-level, domain-level, task-level and application-level (Guarino 1997). Because the purpose of using ontology to present SCM is to share the common understanding of the group meeting in the SC3DGIS environment among distributed people and software agents, The SCM ontology can be seen as an application-level ontology.

In developing an ontology, the primary task is to define the main elements, such as terms (vocabulary), discourses (classes), properties and constrains of properties. A detailed approach can be seen in a technical report from Noy and McGuinness (2001). Figure 5-2 shows a conceptual diagram of an application-specific ontology derived from SCM. These concepts, together with their properties, will be coded and encoded as contents of Agent Communication Language (ACL) messages (see Section 8.4). A detailed XML encoded ontology can be found in Appendix 3.

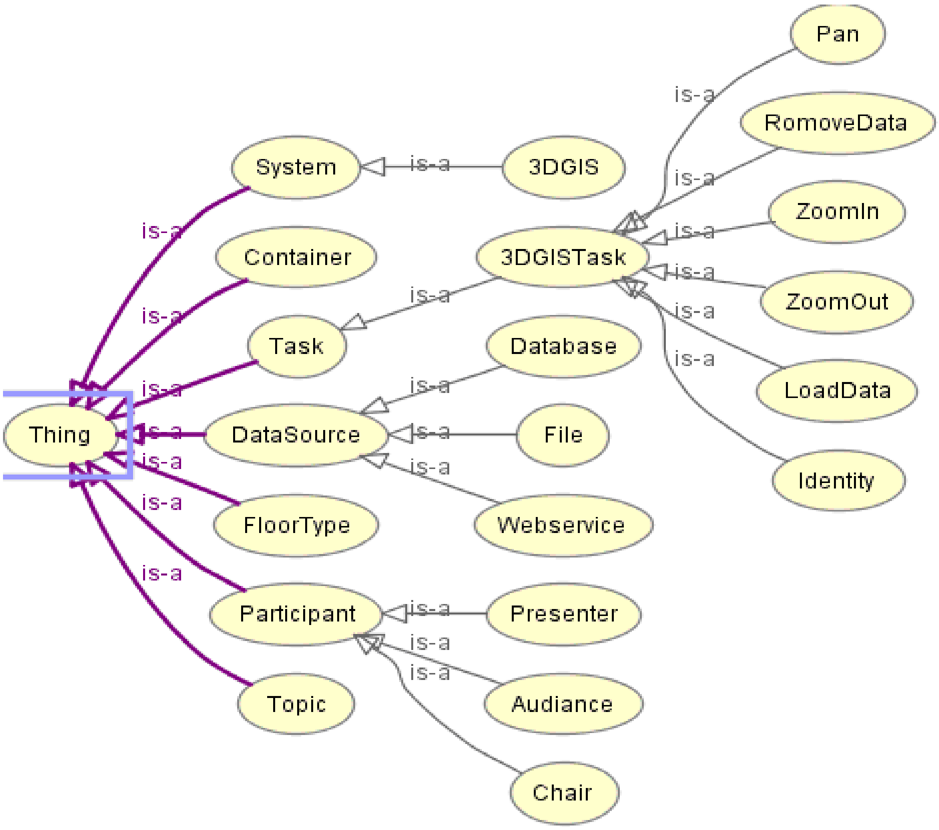


Figure 5-2 Conceptual diagram of the application-specific ontology for SCM

5.4 Summary

In this Chapter, a social collaboration model and its ontology presentation have been developed and presented. This not only provides guidelines for the SC3DGIS development in this research, but also serves as an abstract model for any collaborative and distributed graphics system, such as a collaborative 2D GIS, to share the system status. The social collaboration model is presented as an application-level ontology for shared understanding and further design and development of the SC3DGIS. The ontology is first presented as an XML presentation. It is then is transferred via Java classes to the shared 3D view system and agent support system so that each client can understand the other.

Chapter 6 3D Data Analysis, Modeling and Selection

The 3D GIS data model[2] for collaborative purposes has special requirements, as the special functions of collaborative 3D GIS differ from single-user-based 3D GIS. The main difference lies in the consistency requirements in which the 3D model needs to be visualized synchronously and consistently in a distributed manner. In particular, the geographically dispersed users need to see the same 3D model with the same states such as viewpoint, styles and colors, in their distributed 3D environments. There are two approaches to achieve a collaborative 3D GIS data model: creating an entirely new data model or selecting/modifying a current 3D data model to fit into the requirements.

Unlike the 2D GIS model, there is no dominant and commonly-accepted 3D GIS data model in traditional 3D GIS platforms. Although a lot of early research has been conducted to study the design and development of 3D GIS models such as Triangulated Irregular Network (TIN), grid DEM, 3D Formal Data Structure (3D FDS) (Molenaar 1990, Rikkers et al. 1993), Tetrahedral Network (TEN) (Pilouk 1996), Simplified Spatial Model (SSM) (Zlatanova 2000), Urban Data model (UDM) (Coors 2003), and the Object-oriented 3D GIS model (OO3D) (Shi et al. 2003) (The related literatures, reviews and evaluations can be found in the papers from Zlatanova et al. (2000) and Shi et al. (2003)), many of these models focused

[2] **3D data model vs. 3D data format**
In this thesis, 3D GIS model refers to the presentation of real 3D objects from a real world point of view, while 3D GIS format refers to 3D data from a software point of view. For example, when a 3D model, e.g., TEN, is designed and developed, the term “model” is used. However, when TEN is loaded into a 3D system or imported and output to other systems, the term “format” is used.

on how to present topologic relations among 3D objects. Successfully used cases in a 3D GIS system are not found. GML (OGC 2006) is known as an open GIS data standard for the GIS community and has many application extensions[3], like 3D City Models (Kolbe 2006). However, it has some limitations when used in collaborative 3D systems. These limitations will be discussed in detail later in this thesis.

While more 3D data models appear in non-GIS fields, such as virtual reality (VR) and augmented reality (AR), 3D Video games, manufacturing and construction design, these models, for example 3DS Max, AutoCad DXF, VRML/X3D, Vivid, Wavefront OBJ, and RenderWare RWX, focus on 3D visualization effects and interaction. Specific requirements in the GIS field, such as topology and semantics, are not emphasized in these models. Hence, these non-GIS 3D data models could be alternatives or have extendable potential to be used for collaborative 3D GIS data model.

The main purpose of this chapter is to design/select a suitable 3D GIS model for a synchronous collaborative 3D GIS environment (SC3DGIS). In this thesis, rather than being limited to 3D GIS data models, the typical 3D data models mentioned above (GIS and non-GIS) are reviewed by introducing the 3D data model elements which describe the core 3D model features. By taking these core 3D model features as well as the requirements of the SC3DGIS platform into consideration, the criteria for SC3DGIS data model are then derived. According to these criteria, an "ideal" 3D GIS data model is described to meet

[3] http://en.wikipedia.org/wiki/GML_Application_Schemas/

requirements in both 3D GIS feature and collaboration. However, the criteria for the data model may be affected by other factors, such as how many users are in the user community and how easy it is implemented. Instead of designing a new data model, in the real development, a suitable 3D GIS model is selected from current commonly used 3D data models based on the above criteria and the "ideal" 3D GIS model description.

6.1 3D Data Model Elements

Generally speaking, a 3D data model can be represented by the elements of geometry, topology, semantics, 3D model states, and styles. Together, these elements describe a comprehensive 3D objects set. Geometry focuses on presenting the 3D primitives and objects, such as points, lines, polygons and cylinders. Topology focuses on the relations among the 3D geometric objects. Semantics focuses on the annotation, attributes and the more complex domain ontology of the 3D data model. The 3D model states and styles usually are not mentioned in content-based 3D models because they represent the run-time features and can be varied over time. Figure 6-1 shows the simple relations among the elements in two 3D objects.

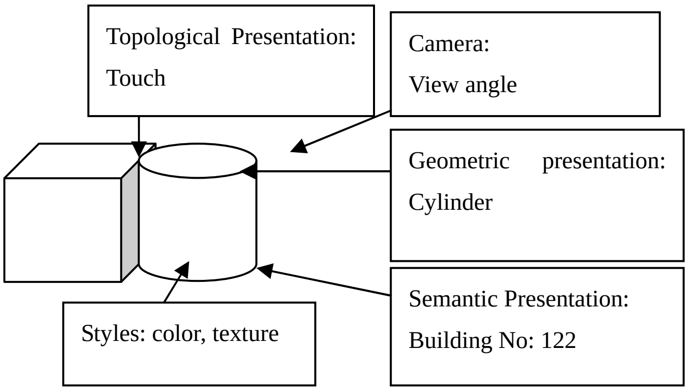

Topological Presentation: Touch
Camera: View angle
Geometric presentation: Cylinder
Semantic Presentation: Building No: 122
Styles: color, texture

Figure 6-1 Diagram of 3D model elements

Geometry: 2.5D model vs. true 3D model

There are several different definitions of 3D objects. Larrivee et al. (2005) semantically identify the differences between 3D objects and the 3D universe. From a 3D universe point of view, a 3D shape is a point, line or polygon that stores x-, y-, and z-coordinates as part of its geometry, while from a 3D object point of view, a 3D shape as a solid has length, breadth, and depth. The latter point of view, defining a 3D shape as a solid, avoids confusion between the number of dimensions of the universe and those of the object. However, a 3D surface model which is composed of 3D objects as polygons is considered a 3D model by most 3D systems. In this research, although 0D, 1D, 2D and 3D objects are defined as point, line, polygon and solid, a 3D surface model and 3D solid model are all called 3D models. A surface model represents the surface, e.g., the boundary of the object, not its volume (like an infinitesimally thin eggshell). A solid model defines the volume of the object represented (like a rock).

Early 3D GIS data models, such as TIN and GRID, are called 2.5D models. For example, the GRID DEM is a sampled array of elevations (z) that are at regularly spaced intervals in the x and y directions. TIN is a set of adjacent, non-overlapping triangles with x, y coordinates and z vertical elevations for their vertices, together with a topological relationship between the triangles and their adjacent neighbors. The disadvantage of these two models is their limitation in presenting complex 3D geometries such as building windows, pipelines and geological objects. For example, in the GEOIDE project, when the geoscientists make a decision based on the geological situation, they must accurately characterize very complex

3D geological systems that are made up of a multitude of linked objects such as rock strata, faults, folds and fractures. The solid model on the opposite has the advantage of being able to present complex 3D objects and carry out complex 3D data analysis such as petroleum reservoir characterization or groundwater contamination, stereo block diagrams, and geo-object cut-away.

In order to present more complex 3D geometries, several 3D models comprising both a surface model and/or a solid model are introduced in 3D GIS, such as 3D FDS, TEN, SSM, OO3D and GML. The early Google Earth's KML (Google 2006) model is also a 2.5D model. The basic 3D object consists of geometry primitives, features and styles. Geometry primitives are 2D primitives such as point, line string, polygon and the aggregation of multiple geometries. Given an altitude, the 2D object will be extruded to simple 3D object. Starting from KML 2.1, through the COLLADA interchange file format (Khronos Group 2006), Google Earth has the ability to present complex 3D geometry.

Primitives: simple primitives vs. complex primitives

A single 3D object can be composed of primitives and aggregations. For example in SSM and GML models, every basic primitive has the three dimensional coordinates of x, y and z. These basic primitives can be 0-dimensional, such as a point, 1-dimensional, such as an edge, 2-dimensional, such as a surface, 3-dimensional, such as tetrahedron, or all of them. The higher dimensional primitives can be composed of primitives from the next lower dimension of primitives.

Primitives composed of other primitives are defined as complex primitives; otherwise, primitives are defined as simple primitives. As we know, GML and SSM are all complex primitives, while KML and X3D are the simple primitive models.

Coordinate System

GIS-related data has two types of coordinate systems: the absolute coordinate system and the relative (local) coordinate system. The relative coordinate system can be easily transferred to the absolute coordinate system given several reference points. In GIS, these reference points are always part of the metadata. However, when the data is a 3D data (model) from other fields such as 3DS Max (Autodesk) or Maya, most of them are based on relative coordinate systems and do not have reference to the real world since they are not geo-referenced data.

The solution for transferring a non-geo-referenced 3D data to a geo-referenced 3D data is to utilize affine transformation presented by 4×4 matrix, which includes rotation, scaling, and translation transformation. Any 3D coordinate system transformation can be made through this matrix. The matrix can be obtained through three corresponding (pair) points between the two coordinate systems (see Appendix 4 for detailed coordination transformation and usage example in Java3D).

Some of the 3D data formats directly store the matrix or the parameters (scale, translation and rotation) presented in matrix into the data files. For example, X3D just stores box objects with a size parameter (origin of 0, 0, 0) and applies transformation parameters into the box so that when the objects are displayed, they can be moved to a correct geographical position with transformation parameters. Using this method, any model can be geo-referenced by adding geo-related nodes. In fact, X3D (Web3D 2010), which includes all the features of GeoVRML (GeoVRML Working Group 2008) in presenting geo-reference related information, such as GeoLocation and GeoOrigin, makes it possible to present geographic information.

KML also has a similar method when COLLADA is introduced into KML. COLLADA defines the 3D model without a geo-reference and KML moves the model to the correct location, scale and rotation on the earth with spatial-related tags, such as Location, Orientation and Scale. The advantage of the method is that it is not necessary to give geographic coordinates to every point of the model (compared to the GML model). This opens a door for all 3D graphic models to put their non-geography model into the real geographical world.

Topology

Topology describes spatial relationships between adjacent objects. Topology among geometric objects causes more interests on GIS researchers. In GIS, topology can be implemented explicitly through data structure. For example, ESRI’s coverage data format explicitly stores topological relationships among neighboring polygons in the Arc Attribute Table (AAT) by storing the adjacent polygon IDs in the LPoly and RPoly fields. Adjacent lines are connected through nodes, and this information is stored in the arc-node table. For the TEN model, the arc-node relationship is stated in the ARC table; the TRIANGLE table contains the tetrahedron-triangle-edge link. Similar methods for 3D topology can be found in GML specification and CityGML, which is more complicated than its 2D counterpart. The edges are defined by a start and an end node, and surfaces are bounded by an outer ring, which is an ordered set of edges. Nevertheless, the topology could be obtained implicitly through on-the-fly computation. The model does not store any topological relations; rather the relations can be calculated based on 3D geometry in a runtime environment. The 9-intersection method (Egenhofer and Franzosa 1991), Voronoi-based method (Chen et al. 2004) and other methods are introduced by many researchers in an attempt to solve this problem.

Semantics

Semantics focuses on the annotation, attributes and more complex domain ontology of the 3D GIS data model. The semantics representation plays a very important role in the GIS data model because more information, including geometric and topological information, can be presented by semantics information. Current 3D systems mainly represent 3D semantics with appearance, color and texture. In 2D GIS, a series of attributes used to annotate the 2D objects semantics together with geometry objects are presented as a Feature object, for example GML and OpenGIS Simple feature. Although many 3D models, such as KML and X3D, do not have this explicit presentation, if these 3D models are presented in XML, they can be easily presented the same as the Feature object.

State and Styles

A 3D model state usually describes how the users look and navigate the 3D model in a 3D system through a camera. The camera "captures" the 3D view from its current location, orientation and perspective. The visualization system then draws that view on the user's display device. The style features, such as color, line style and texture, represent how the 3D model looks. State and style can be presented in a 3D model as a fixed status, such as in KML and X3D, but they also can be saved and configured in 3D systems and shown on the fly, such as in GML.

Summary

Table 6-1 summarizes the main 3D geometry models from geometry primitives, topology, XML presentation, thematic presentation, state/style and standards. Although these models are designed for different purposes, they share the same elements in presenting the real 3D world. In all the models, OpenGL and DirectX are lower-level rendering APIs which may support higher-level 3D data models for rendering. The higher-level models (such as SSM and GML) use the lower-level APIs (such as OpenGL or DirectX) to render in the 3D system. As for the thematic presentation, many models, such as OpenGL or X3D, do not give a special definition in this regard, but that does not mean they do not support thematic presentation. Thematic presentation may appear in the object-oriented system design and database design.

Table 6-1 Summary of 3D geometry model

3D models	Geometry Primitives	Topology	XML	Thematic	Status/ Style	Standard
DEM/Grid	Point	No	No	No	No	OpenGIS
3D FDS	node, arc, face and edge		No	Feature object	No	No
TEN	tetrahedron, triangle, arc and node	Partial	No	No	No	No
SSM,	Node and face	No	No	No	No	No
OO3D	node, segment and triangle	Partial	No	No	No	No
GML, CityGML	Complex 3-dimensional geometric primitives	Yes	Yes	Feature object	No	OpenGIS
KML	point, lineString, lineRing, polygon and model	No	Yes	Yes	Yes	OpenGIS
VRML/X3D	box, cone, cylinder, sphere, indexedFaceSet.	No	Yes	No	Yes	Web3D
OpenGL	Points, lines, polygon, quads, quad-strip, etc	No	No	No	No	OpenGL Architecture Review Board
DirectX	Vertices and triangle	No	No	No	No	NO

Oracle Spatial 11g also provides solid 3D data support and allows for some 3D operations, such as within_distance and any_interact. However, few rendering systems directly support

Oracle Spatial data. When database data, like Oracle spatial data, for instance, need to be loaded into a 3D rendering system, these data are usually transformed into other formats such as X3D or GML format.

Next several sections will identify if these elements meet the requirements of SC3DGIS. The criteria of SC3DGIS are discussed as well.

6.2 Criteria for SC3DGIS Data Model

The criteria of data model are established based on the requirements of a SC3DGIS environment. The design objective of SC3DGIS is to help geographically dispersed experts and non-experts share a common understanding, 3D model and environment in order to collaboratively work together. According to the proposed SC3DGIS features mentioned in Section 4.3, these criteria include several aspects such as Web and Internet access, GIS requirements, collaboration capability, extendibility and interoperability, and implementation. A detailed requirements description is as follows:

- Meet Web and Internet-enabled environment. Because the data is accessed and transmitted over the Internet, the proposed 3D GIS model should keep it as simple and light-weight as possible. XML-based presentation is important for Internet-based applications because most data transmitted on the Internet is encoded by XML. There are also tools such as XSL to transform data format from one to another.
- Meet the 3D GIS requirement. The core GIS data model related requirements should be included, such as a geo-referenced coordinate system, geometry, topology and

semantic information. It should be a true 3D model (surface model and solid model).

- Meet real-time collaboration environment. The proposed model should have the capability to represent model status and styles, which keeps the model state consistent for all clients. The model state includes point of the view (camera position, look angle) and rendering styles (color, fog, shade and lights). The proposed model also should have the capability to represent interactive and dynamic actions such as touch sensor and animation.
- Extensibility and Interoperability. The proposed model is extensible not only to a specific field such as geology, utility and even biology, but also to the new elements, such as those which are user-defined. The proposed model is also transparent to system design, which means the data model design is separated from the system design and rendering tools. The proposed model is able to exchange with other 3D data standards and should have the potentials to become a standard.
- Implementation. The model should be widely used and supported by different 3D systems and rendering API. Not only the 3D views but also the loaders are needed to support the model. Because some applications have their own 3D view/rendering system, they just need a loader to load the model.

Topology consideration

Explicit representation of 3D topology is used in many 3D GIS model design such as GML and TEN. However, the representation ignores implicit relationships and has obvious disadvantages such as the complexity of maintaining, collecting, and constructing the structure, as well as the inefficiency in a Web-based environment. For example, if there is an overlap relationship between two 3D objects, the bounding volume of the two objects will be overlapped in the 3D space for identifying the relationship.

Thus, the collaborative 3D GIS model will not store topologic information; instead, the topology among objects can be calculated on the fly through a bounding volume related algorithm. Since the 3D model is objected-oriented, only the relations among objects are considered. The relationships of primitives inside the objects are hidden and ignored. This method will again greatly reduce the complexity of the data structure and increase the performance of the 3D GIS at the application level.

Coordinate system consideration

Most GIS related data models use an absolute coordinate system only, for example GML and SSM in which every object is presented with a true ground coordinate system. However, many non-GIS related 3D data models use a relative/local coordinate system, for example, 3DS Max. These models are usually built by a 3D model builder software such as AutoDesk 3ds Max (AutoDesk 2009) and SketchUp (Google 2009).

In considering integrating these non-GIS related 3D models into 3D GIS systems, it is necessary to use both coordinate systems. Successful examples of this are KML and X3D. Through introducing COLLADA into KML, any non-geographical 3D model can be smoothly integrated into Google Earth 3D.

Primitives consideration

Primitives are the basic elements in presenting 3D objects. However, there exist two kinds of over-killed problems: over-killed encoding and over-killed compositing.

Over-killed encoding means that although very small parts of the primitives are used, the whole primitives have to be encoded in the system. For instance, for a 3D pipeline object, primitives point and line are enough to present a pipeline. If a 3D system is designed for a 3D pipeline system, only primitives point and line are necessary. However, from the system designer's point of view, even if just two primitives are used, the whole primitives set, for example in GML, have to be encoded in the 3D system. In this case, GML is over-killed for a 3D pipeline system.

Over-killed compositing usually happens in the complex primitives. For instance, in GML, when higher dimension primitives such as tetrahedron are used, 3D primitives have to be presented by 2-dimensional primitives (surfaces), and 2-dimensional primitives then have to be presented by 1-dimensional primitives (lines), and so on. However, the proposed 3D system may not have the requirements to check the primitives under 2-dimensional primitives (polygons, lines and points). Therefore the presentation for the complex primitives is over-killed.

The solution for these two over-killed problems is to keep the primitives as simple as possible while keeping the primitives extendable to composite all kinds of objects.

Interoperation consideration

The method to solve interoperability could involve the use of XML and related technologies. The combination of the facts that the XML syntax is text based and fairly straightforward to parse has lead to the emergence of XML as the premiere data interchange format when cross-platform interoperability is required.

Because of the extensibility and structured nature of XML used for communication between different systems, communication not only between internal computing systems but also external systems (such as vendors, customers and partners) can be implemented using the common technology regardless of the platforms and technologies used for each independent system, which makes it easier for information sharing and system interoperability and knowledge transfer between different computing teams. From one source of XML-based information one can format and distribute it via a multitude of different channels with minimal effort. Through the use of extensible style language (XSL), developers can easily separate content from formatting instructions.

6.3 Proposal of Data Model for SC3DGIS

According to the criteria discussed in the previous section, the proposed 3D GIS data model could have the following elements:

1. Geometric primitives: Point, Line, Polygon and Box
2. Geometric object: composed of geometric primitives and composite of several geometric objects

3. Geometric styles property: composed of properties describing appearance, color, texture and other legend information.
4. Attribute object, semantic information: For example, the owner’s name, gender, address and so on are represented as attribute objects. The feature model in GML can be borrowed in the proposed 3D GIS mode. A feature is composed of one 3D geometric object and one attributes information annotating the 3D geometry.
5. Model object: composed of geometric objects, attribute objects and state objects. A model object also supports aggregation of model objects. For example, the complex building (a model object) consists of parts (also model objects), and these parts again are a composition of a main part (model object), chimneys (model object) and balconies (model object).
6. Model state object: composed of the camera parameters representing how the user look and navigate the 3D model in the system. This component help keep the replicated clients consistent in a collaboration environment.
7. Theme object: composed of the same model objects representing the same theme. For example, the building theme, a theme object, will include a collection of building objects. A building object will be one of the model objects.
8. Supports both local coordinate system and absolute coordinate system.

Figure 6-2 illustrates the relations among the above elements. From this UML-based diagram, we can see the following detailed composition relations: a theme object is composed of 3D model objects. A 3D model object is composed of geometry objects, model states and attribute objects. A geometry object is composed of styles and abstract primitive object. The abstract geometric primitive is extended to Point, Line, Polygon and Box primitives. The Point, Line and Polygon have X, Y and Z values present 3D surface objects. The Box primitive specifies a rectangular parallelepiped box, in which the extents of X, Y, Z axes are specified respectively. A similar definition can refer to X3D specification (Web3D-BOX 2008). The Box primitive is the simplest method to present 3D solid objects. In other words, the Box primitive can present any solid objects. In theory, Point, Line, Polygon and Box can present any 3D surface model and 3D solid model. That is the reason why only these four primitives are proposed to present Geometric objects.

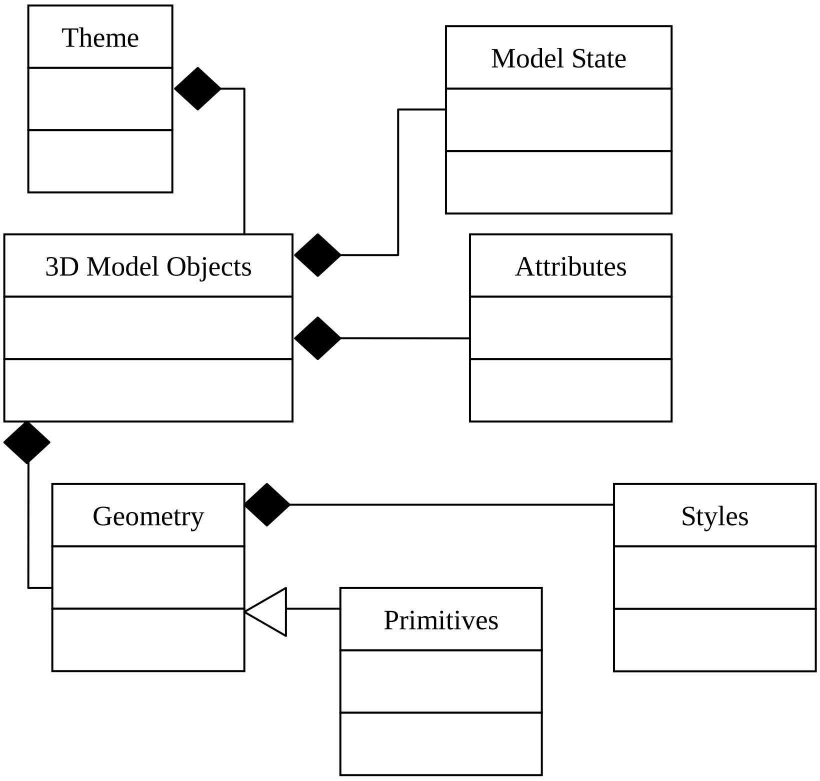


Figure 6-2 Collaborative 3D GIS data model elements and relations

6.4 Model Selection

There are two methods to design the data model. One method is to design a completely new data model like GML. Another method is to extend a commonly used model, such as GML or VRML/X3D, to meet the specified criteria. If both methods meet the requirements (the above criteria and considerations), choosing an existing 3D data model is preferred. This is because though the first method can design any model which perfectly fits into the need, defining the model is an exhausting and complicated task. Even if the model is defined and published to the users, it will require more time to be recognized by public. The worst situation is that the new model will not be accepted by other users after a long time testing and become a dead model. The second method is to find an existing 3D data model which meets the proposed criteria. According to Table 5.1, the potentials have been narrowed into four models: KML, VRML/X3D, GML and Grid/DEM. The advantages and disadvantages of these models are discussed compared to the above criteria, and the one with the best fit will be selected as the data model for SC3DGIS.

KML

KML is a XML-encoded data model used in Google Earth. Obviously, KML strongly supports Internet-based features. After it is updated to version 2.1, KML supports COLLADA format, a complex 3D model, but the solid model support cannot be found. KML itself has elements to support surface 3D geometry, styles, model state, even a “Feature” object (this “Feature” object is not the same definition as in the proposed model). Although there is not an explicit attributes definition, it is easy to extend the attribute element for KML because it is an XML-based presentation. KML with COLLADA has model status elements such as ColorStyle and AbstractView. Only the interactive and dynamic actions elements are not found in KML. Since KML is in compliance with OpenGIS standard and the XML-based format, the extensibility and interoperability are strong. Although KML is supported by an import/output option by many systems, such as NASA WorldWind, ESRI ArcGIS Explorer, Adobe PhotoShop, AutoCAD and Yahoo! Pipes, only few developing tools (loaders) are available to encode/parse KML. The 3D renders for KML are exclusively limited to Google Earth. This is the disadvantage of KML.

GML

GML supports both surface model and solid model. GML is also an XML-based Open GIS (OGC) standard especially for GIS field. Simple Feature as partial GML is a standard format for Web map service. The Web and Internet requirements are strongly supported. As well, GML supports full elements such as Theme, 3D model object, Attributes, Geometry and primitives (see Figure 5-2) and the combination relations. The support to GIS is also strong, as is extensibility and interoperability because GML is an OGC standard and has an XML-based presentation. However, there are few loaders available to encode/parse GML. The disadvantage is that GML does not present the styles and model state explicitly and has no interactive or animation elements for collaboration. Thus, the collaboration criteria are weak.

X3D

X3D has good capability to present both surface model and solid model. The 3D geometry nodes for the surface model include ElevationGrid, IndexedFaceSet and Sphere; the 3D geometry nodes for the solid model include Box, Cone and Cylinder. The ElevationGrid 3D node is a typical GRID model in the GIS area. The Box node is the fundamental geometry for Voxels presentation, a raster approach to present a 3D universe into volume elements. X3D is an XML-based data standard for presenting 3D computer graphics. X3D defines networking components to access file-based and streaming resources on the Web, which fully meets Web and Internet requirements. X3D also defines the geospatial component and the 2D and 3D geometry component. Like KML, where there is no explicit attributes definition which is useful elements in GIS, it is easy to extend the attributes elements for X3D since it is an XML-based presentation. The GIS requirement is strongly supported. X3D defines rich model style components such as rendering, texturing, lighting and interactive components. All these components will meet the requirement of collaboration. Thus, the support to collaboration is also strong. Similar to KML and GML, extensibility and interoperability are likewise strongly supported because of the XML-based representation and Web3D standard. Although there is not a dominated application using X3D, it has more loaders than GML and KML to support third-party development.

GRID

Grid format is widely used to present terrain models or DEM in a GIS field. It is used in many applications in the GIS community, for example, 3D Analyst of ArcGIS from ESRI Inc. and Image VirtualGIS from EADAS Inc. However, Grid format neither supports the 3D solid model nor has an XML-based presentation. Attribute and collaboration elements are also not presented in this model. Therefore, the Web and Internet support, GIS support, collaboration support and extendibility are all weak, even though the implementation support is strong because many GIS software applications support it. Because of this, Grid has become one of the core components of KML, GML and X3D in presenting the earth’s surface.

The above analysis leads to the conclusion that X3D needs to add/define attribute objects to meet the requirements of the collaborative 3D GIS data model, while GML requires adding/defining styles objects and model states. It is obvious that attribute objects are easier to add for XML-based data presentation. Furthermore, GML for 3D presentation is more complicated than X3D. The complication lies in the inheritance structure. For example, in GML, 3D primitives are presented by 2D primitives, and 2D primitives are presented by 1D primitives, and so on. When a 3D object is presented, all primitives (2D, 1D, 0D) have to be encoded into the system, which will cause a huge overhead for every client and slow the system down. Although KML also meets some requirements, it does not support a solid model and can only be rendered for Google Earth. X3D, as an ISO standard and XML-based file format especially for presenting 3D computer graphics, is independent from any rendering system.

6.5 Summary

This chapter identifies the main elements of a 3D model and shows how these elements meet the requirements of real-time collaboration in a 3D GIS. These criteria and analysis results, such as coordinate system consideration, primitives consideration and the comparing of the

GML, X3D, KML and GRID, also benefit the 3D GIS community

In detail, this collaborative 3D GIS environment requires not only basic 3D GIS data features such as geometry, coordinate system, and semantics, but also collaborative requirements, such as consistency and Internet accessibility. Several 3D GIS data models are reviewed and the core elements are derived. Through comparing these elements, the collaborative 3D GIS data model is believed to meet following criteria:

- Both surface model and solid model should be supported.
- Primitive geometries should be as simple as possible, which solves the over-killed encoding problem and over-killed composite problem.
- The 3D GIS data model should explicitly support both the absolute coordinate system and relative/local coordinate system.
- Topology presentation could be calculated on the fly when needed.
- Presentation for data model state and styles are necessary for collaborative 3D GIS.
- XML format support.

Although GML is well known as an international standard and has many use cases, GML 3D is still in its research stage. Referring to the above criteria, GML is not suitable for collaborative 3D GIS environment because:

- Primitive geometries have over-killed problems;
- No relative/local coordinate system is presented;
- Topology presentation is too complex; and

- No model style and status are presented.

KML can only be rendered in the Google Earth environment. Solid model support cannot be found in KML and COLLADA. X3D meets most of the criteria and has more rendering engines. Therefore, X3D is selected for the collaborative 3D GIS data model.

Chapter 7 Architecture Design of Shared 3D GIS

This chapter presents the architectural design of the shared 3D GIS view and shows how the system consistency is maintained through the event distribution model and floor control mechanism. The initial architecture includes a register server and replicated clients. The register server is used as a bridge to collect and broadcast messages. Most components, such as 3D rendering, collaboration handing, and network and multicast transportation, are in the client side.

A hybrid architecture is adopted. The architectural components of the replicated SC3DGIS clients were modeled in great detail, specifying how event and data messages are encoded, reconstructed, and coordinated. The model serves the purposes of: (1) allowing easy selection of software components or packages; and (2) providing a basis and guideline for design and implementation of this kind of system.

7.1 Hybrid Architecture

Figure 7-1 illustrates the overall system architecture of the SC3DGIS based on a hybrid (semi-replicated) architecture, i.e., replicated clients and shared central servers. The network connections between the replicated clients and the server components support multicast communication, while internal connections among server components are service calls realized through component interfaces. Two types of shared central servers are included: register server and proxy data server. The proxy data server not only requests data from other servers, such as the Oracle database server, but also transforms this data into a suitable format (see Sections 7.7 and 9.2 for detailed information). Through the proxy data server, replicated clients ("replicas") can access shared files or DBMS. If there are no restrictions on access, the replicas can access shared files and DBMS without the proxy data server. The register server is a simple component providing a Web-based access interface (e.g., IP address, port number, and session name), receiving and sending messages to replicas. According to the protocol of the messages, the data can be sent to one identified replica or broadcasted to others or to all replicas.

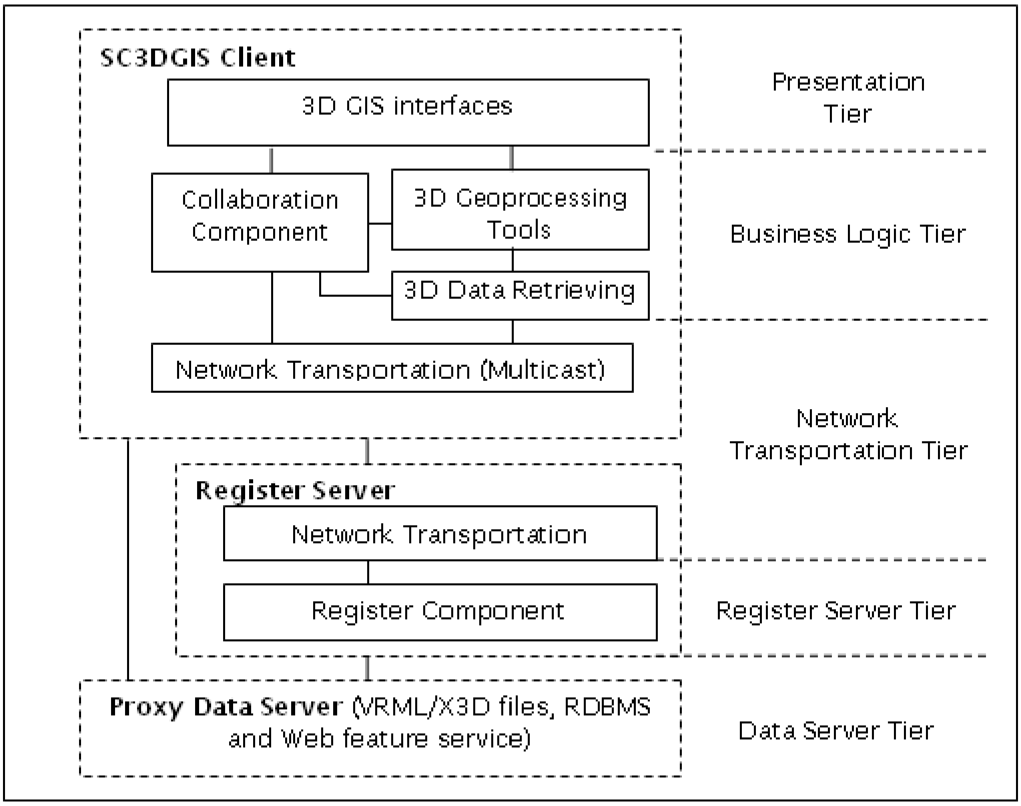


Figure 7-1 Conceptual View of SC3DGIS Architecture

Referring to traditional architecture, it is also a client/server architecture because of the following reasons:

- Clients do not communicate with each other. Rather, all messages are processed through the central server;

- The dominant design paradigm is the "Only Trust of the Server". The server is the sole focal point for message authentication and processing. As a result, the clients tend to be untrustworthy as well as under-utilized.

However, the architecture in this prototype has many special characteristics compared to traditional client/server architecture. First, the register server is not fixed in one server and can be changed to any client. The first client who organizes the shared 3D view session will be assigned as the register server. Second, multicast protocol is used to forward the events from one user to all participants, which requires special control to the data flow between the clients and the server. Traditional development tools and best practices are not applicable for this architecture. Therefore, when architecture is considered for this type of system (CSCW or groupware), the replicated architecture or centralized architecture is referred to handle the multiple users' consistency and multicast data flow.

7.2 Event Distribution Mode

An important advantage of this hybrid architecture is that the 3D rendering images do not need to be transferred every time an operation is performed on the client's side. Instead, event or data messages are transmitted according to actual operations. Figure 7-2 illustrates the procedure of event (message) distributions among components. The numbers in 8-point stars indicate the message passing sequences when Replica 1 operates on its interface. The data flow is shown as follows:

1. When a user operates GUIs in the client application (e.g., Replica 1) that handles 3D GIS contents, for example zooming in the 3D view, an event message is first triggered and sent to the collaboration component of the same client.
2. The collaboration component then serializes and sends the message to the register server on the server side.
3. The register server distributes the message to the collaborative component of every client who is in the same session including the client who created the event.
4. After reconstruction, this event is finally passed onto the 3D GIS component of each client to carry out the same operation as the initiating client did.
5. The 3D GIS component may load locally-replicated VRML files or X3D files, or access and load the same type of shared data file on the Internet through the proxy data server.

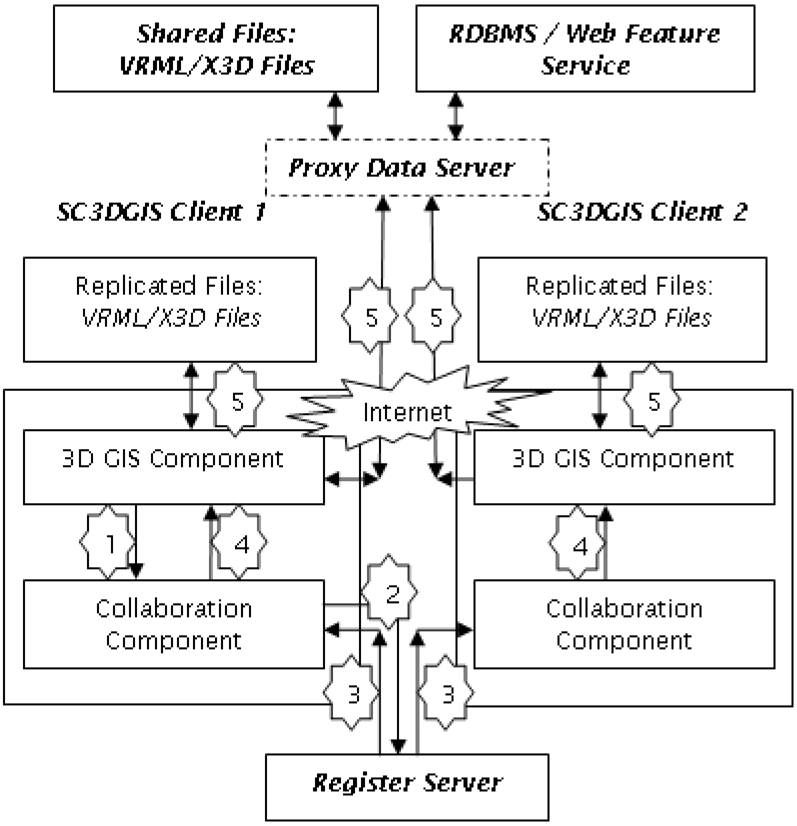


Figure 7-2 Event distribution model against system architecture

In order to maintain the consistency of the distributed environments, the state of the 3D environment, which includes the change of 3D GIS data contents, rendering styles, 3D transformations and so on, have to be sent to other participants in real time through the message passing mechanism described above. The collaboration component is responsible for listening to the changes of the 3D GIS environment (views angles, styles and so on) and transferring the changes to other clients. One of the important 3D GIS states, for example, is the 3D transformation state, which is presented by a 4 × 4 matrix (see Appendix 4 for the detailed 3D transformation and the usage example in Java3D). Figure 7-3 illustrates the call-back process of the 3D transformation message.

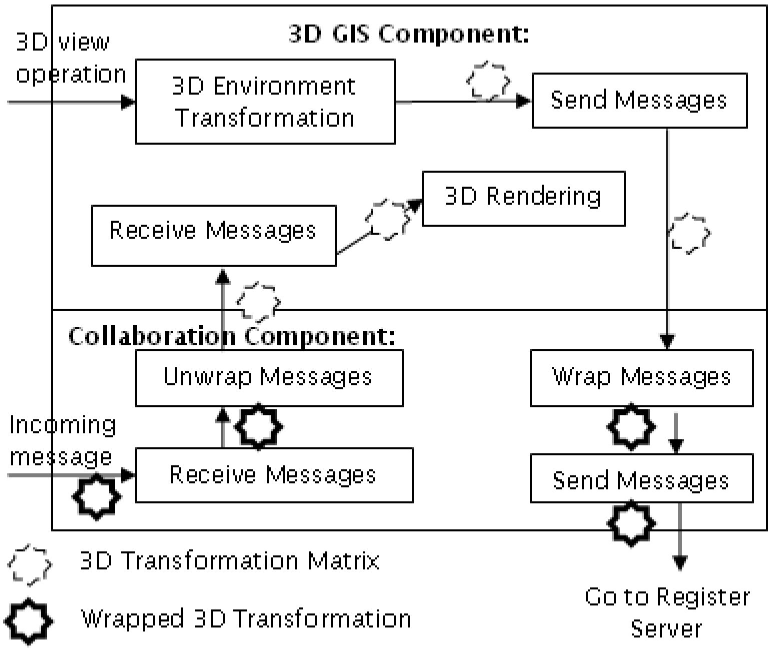


Figure 7-3 Message call-back process

When a user commits an operation on the interface and causes the change of the 3D environment, the 3D environment listener receives a state change event. The event including a 3D transformation matrix is sent to the "send message" module. After extracting the 3D transformation matrix, the "send message" module sends it to the collaboration component. The "wrap message" module then adds some information like the destination address, sender's information and type of message into the matrix message, and sends it to the "send message" module in this component. After serialization, the "send message" module sends the wrapped message to the register server. The register server then transfers this message back to this client and other clients who join the same session. As shown in Figure 7-3, the message as an incoming message is received and unwrapped in the collaboration component through the "receive message" module and "unwrap message" module. Finally, the message is reconstructed and sent to a 3D GIS component for rendering or other operations.

The collaborative component is used to handle communications between clients (replicas) and between clients and the central servers, and keep the whole system in consistency. This component is responsible for data translating and encoding, floor control, and session and user management. Among them, data translation and encoding receives data (messages) from a 3D GIS component or the register server and at the same time sends the data to the indicated destinations. According to the information from floor control and session and user management, the destination address, which is identified by IP address, port number, session name, and client name, is added and wrapped in the messages through the "warp message" module.

7.3 Data Coding and Encoding

The data coding and encoding module provides translation between messages and simple text strings or XML coded documents so that other clients can understand each other's behaviours inside the shared 3D GIS view. Figure 7-4 illustrates the structure of messages. There is a base class Message which extends to several classes, for example EventMessage class, 3DtransformationMessage class, and InstantTextMessage class. The base class Message has basic operations and attributes which help send and receive messages and store message contents. The extended messages class handles different messages separately. For example, EventMessage handles incoming events and sends them to others, while 3DtransformationMessage handles 3D transformation array. More detailed message contents will be coded and encoded for SCM and its ontology presentation.

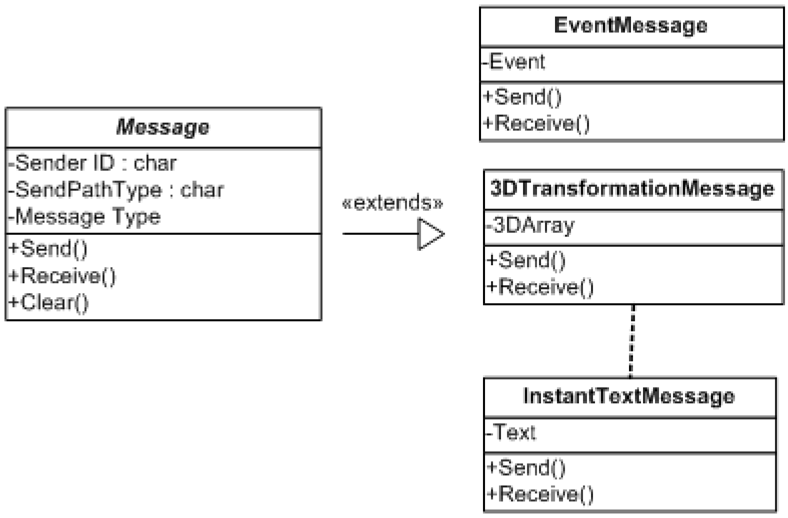


Figure 7-4 UML-based messages structures

7.4 Replicated Rich Clients

Usually the architecture of a Web-based GIS, such as Web Map Service (WMS), Web Feature Service (WFS) and Web Terrain Service (WTS), has three tiers: the Presentation (Client) Tier, Business Logic Tier, and Data Storage Tier. Each of these tiers may host several application components or services. The Web-based SC3DGIS, however, cannot adopt the above architectures. For example in WTS, though it is the only Web-based 3D GIS service specifications issued by OGC, every operation in WTS, such as zoom or pan performed by a client, will cause a new image to be created in the server and transmitted to the client over the Internet. Because a highly interactive 3D system requires quickly changing views as well as additional time consumed to keep the system in consistency, this kind of centralized architecture, like the centralized architecture of other CSCW systems discussed in Section 3.1, will cause unacceptable latency among multi-users.

The designed architectural model shifts the rendering component and other 3D GIS components from the server tier to the replicas. As far as the rendering component is concerned, this architecture is similar to WFS, where the rendering component is embedded in client and the server provides encoded geodata. This architecture holds a thick SC3DGIS client, which consists of several components, including a 3D data-retrieving component, collaboration component, 3D geo-processing tools, and 3D GIS interfaces (Figure 7-2).

A 3D data retrieving component can deal with all kinds of data sources including shared 3D data files, 3D databases and even Web map services such as WFS. 3D geo-processing tools provide functions for data rendering, 3D data buffering, overlay and 3D analysis functions. Different from its 2D GIS counterpart, 3D GIS usually uses an internal scene structure, such as scene graph, to organize the 3D data, and open rendering tools, such as OpenGL or DirectX, to render the data. One of the benefits of such a design is that the rendering process could be transparent to 3D GIS application developers. The developers only need to handle the 3D data with high-level 3D APIs, such as Java 3D and Xith3D (Xith3D 2006), which supports a scene graph. The developers also have the option of handling OpenGL or DirectX directly and building their own internal data structures (or scene graph). However, this method makes the development procedure more complex and expensive. The collaboration component handles collaboration and communication tasks: floor control, session management, latecomers and event distribution and reconstruction. This component not only communicates with other components in this client but also communicates with other clients who join the same session. The interfaces are typically graphical, menu-driven, icon-based graphical user interfaces.

7.5 Consistency Design

The system architecture adopts the avoiding inconsistency method to ensure consistency during collaboration sessions. This can be accomplished at three levels: the architecture level, system state level, and user interface level.

At the system architecture level, as mentioned in architecture design in Section 7.2, the event multicast process, in which every event always goes through a register server and then goes to the other clients, will help to keep the events passing in the same order. The ordered events passing will keep the system level cooperation consistent. At the system state level, the system states, including 3D GIS models states, 3D objects rendering styles, data loading information and other states, have to be kept in consistency. Part of the information is recorded in the 3D GIS models; others are recorded in the system variables. All the information needs to be sent to the newcomers so that the new clients can catch up with the other clients. When a newcomer arrives, the whole system will keep the system locked until all the information is sent. At the user interface level, the main action to ensure consistency is conducted by floor control. Considering decision-making process use cases, people usually stand in two situations: highly interactive discussion and monologue presentation. In an interactive discussion situation, any person who wants to talk and demonstrate his ideas would like to take the floor immediately without the permission of a moderator, while in a monologue presentation situation, people would like to present their ideas until they are finished. Hence, two mechanisms, detection floor and grant floor, are needed and adopted in this system, according to the above analysis.

For the detection floor mechanism, there is no moderator in this session. The floor is available to everyone based on a “first come, first served” policy. When the user who is holding the floor completes his/her job, the system knows if the user’s job is finished or not through testing the movement of the user’s cursor. Conversely, in the grant floor mechanism, there is a moderator who can decide which client has the right to take over the floor when he/she receives several floor requests. The moderator has the right to take away a client’s floor privileges and give it to another client.

7.6 Shared Cache

When considering keeping the system in consistency, the system states need to be stored in a shared virtual point, which is in a database system or file system, so that every client can access them instantly. However, accessing the above systems will be a time-consuming process.

A shared cache, therefore, is used to store distributed system states in this system (see Figure 7-5). Rather than storing these states in a shared virtual point, usually a storage area on hard disk, the shared cache resides in every client’s memory (RAM). There is a mechanism to keep the shared memories consistent and updated. This method is faster than shared virtual point method, since all the states are updated in RAM.

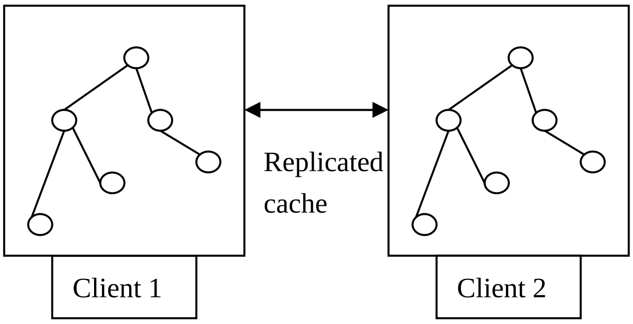


Figure 7-5 Shared cache

Shared cache, also termed Clustered cache, is a cache system where each cache instance is aware of another cache instance in a cluster and is capable of synchronizing operations with its peers. Catch content is typically mirrored. Distributed cache takes clustered caches one step further. It distributes cached data across a cluster to maximize retrieval efficiency and reduce overall memory. There is no requirement to save the data on every cache instance in the cluster. The website, Open Source Cache Solutions in Java (Java-Source.net 2008) lists many types of shared cache systems, such as JBossCache and OSCache. This system selects JGroup/JBossCatch as the shared cache system to store the replicated system state. The reason for selecting JGroup/JBossCatch will be discussed in Section 9.1.

7.7 Shared Data Sources

Replicated file repository is a replicated module existing in every replica. The design of this module is based on the rationale that there should be a shared data source or directory in which every client can exchange shared files without relying on an Internet Information Service. Any user who would like to share his/her data files (3D models) with other users can upload the file to the server. The file will be downloaded automatically to all users' replicated file repository. The benefit of the module is that the local data will greatly improve the 3D rendering performance.

Proxy Data Server is a shared module. The design of this module is based on the fact that database, Web services or a very large file as a data source is not easy to be replicated and thus has to be shared. Because of the access privileges and connection count limit, not all the clients can access the data source at the same time. The Proxy Data Server allows one user to create a connection with the data source and bring the information to the proxy data server. Other users can get the data source information through the proxy data server without directly connecting to the data source.

Furthermore, it would be more convenient for this proxy data service to provide data transformation on the fly. For example, consider the following scenario: it is known that there are some 3D data stored in an Oracle database. The plan is that people can access and load these data into the SC3DGIS through the proxy service and, at the same time, when the data goes through the proxy service, the data can be transferred to a specific data format the system can use on the fly. The procedure is as follows (Figure 7-6):

1. SC3DGIS requests data from Proxy Server.
2. Proxy Server connects to database and sends query condition.
3. Oracle Database gets and sends data back to Proxy server.
4. Proxy server transforms data format.

5. Proxy server sends data to SC3DGIS.

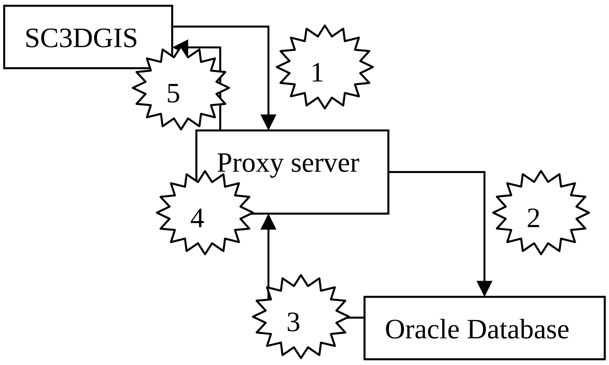


Figure 7-6 Procedure for the proxy server to get data on the fly

As an example, FME server, a spatial ETL (Extract, Transform and Load) tool, provides the above functions. SC3DGIS can interact with an FME server using a simple interface to implement the above scenario. The detailed setup and test cases can be seen in Chapter 9.

7.8 Summary

In this chapter, a shared 3D GIS view was designed and implemented based on a semi-replicated architecture. It is a first attempt in the 3D GIS field, if not ignored, to add collaborative capabilities for a 3D GIS prototype. In this prototype, every client can share not only the data source but also the 3D model, 3D viewpoints and even the operations. A floor control mechanism is designed and developed to manage the multiple users' cooperation and collaboration.

According to the semi-replicated architecture, every client has its own 3D rendering component and collaboration component. Every event as well as the system status are transferred from a certain client to the register server and then forwarded to all participating clients to keep the system in consistency. The “avoid conflict” method is adopted in the floor control mechanism so that every client can automatically detect if he/she has the privilege to operate the shared view.

Chapter 8 Multi-agent Support

Before the agent method was introduced into SC3DGIS, hybrid architecture and event multicasting processes (Chang and Li 2008) were designed (as discussed in Chapter 7). The related prototype, GeoLink3D, was also implemented to evaluate this architecture. In GeoLink3D, a collaboration component embedded in every client is designed to handle basic collaboration-related functions, such as multiple communications, floor control, and user management.

However, designing the collaboration component involves many difficulties. First, the traditional client server (or browse server) and map service (or Web service) will face difficulties in implementing peer-to-peer and peer-to-many data transportation. For example, in use case **Sharing operations**, the shared operation events will be sent to all the users who participate in this session. The traditional client-server or Web service architecture just provides the communication between the client and the Web server or database server. More complicated peer-to-peer and peer-to-many data transportation has to be exposed in the collaboration component. Considering system consistency and floor control mechanism, the collaboration component will be very complicated and error-prone.

Second, the collaborative component cannot easily deal with more complicated collaborative processes which may relate to negotiation and multi-round interaction protocols (IEEE FIPA 2008). For example, in the use case **discussing and approving issues,** the Chair and the participants would make the issues approved through a negotiation process (see Figure 5-1) such as, agree, don’t understand and refuse. This structured interaction is not easily implemented with hard-coded pattern. Moreover, collaboration among users requires not only human-machine Interface (HMI), but also interface between machine and machine. This requires a shared understanding of the interactions among machines and users.

In summary, these difficulties include: 1) peer-to-peer and peer-to-many data transportation, 2) complicated interaction protocols such as multi-user negotiation, and 3) shared understanding. However, the multi-agent system, such as JADE, provides solutions to these challenges. A multi-agent system’s capabilities are as follows:

1) Peer-to-peer and peer-to-many data transportation support. The multi-agent platform provides a method to transmit all kinds of structured data such as messages, data and operations from one client to other clients directly.
2) Complicated interaction protocols such as multi-user negotiation support. The multi-agent method usually follows FIPA interaction protocol. These interaction protocols, such as FIPA-Request, FIPA-query, FIPA-Request-When, FIPA-recruiting, FIPA-brokering, allow the initiator to verify if the expected rational effect of a single communicative act has been achieved.
3) Shared understanding. The multi-agent approach usually provides a FIPA-compliant platform and ontology to promote shared understanding in the working context and environment, topics, tasks and users.

In order to solve the problem, this research introduces a multi-agent method to integrate complex interactions for virtual group meetings into SC3DGIS. The basic collaborative functions in the collaboration component remain the same, whereas the more complicated multiple interactions are extracted into a multi-agent part. As well, a distributed virtual group meeting is simulated based on a multi-agent platform.

8.1 Multi-agent System Design Methods

Agent-based software engineering is a relatively new field and can be thought of as an evolution of object-oriented programming. Many software engineering methodologies such as Gaia, MESSAGE and Cassiopeia are used to analyze how to design multi-agent system. These methodologies are different from traditional object-oriented method mainly in the different abstractions between agents and objects (Odell 2000). Nikraz et al. (2006) gave more detailed methods and steps to design and implement multi-agent systems. These main steps include Planning, Analysis, Design and Implementation and Testing (see Figure 8-1).

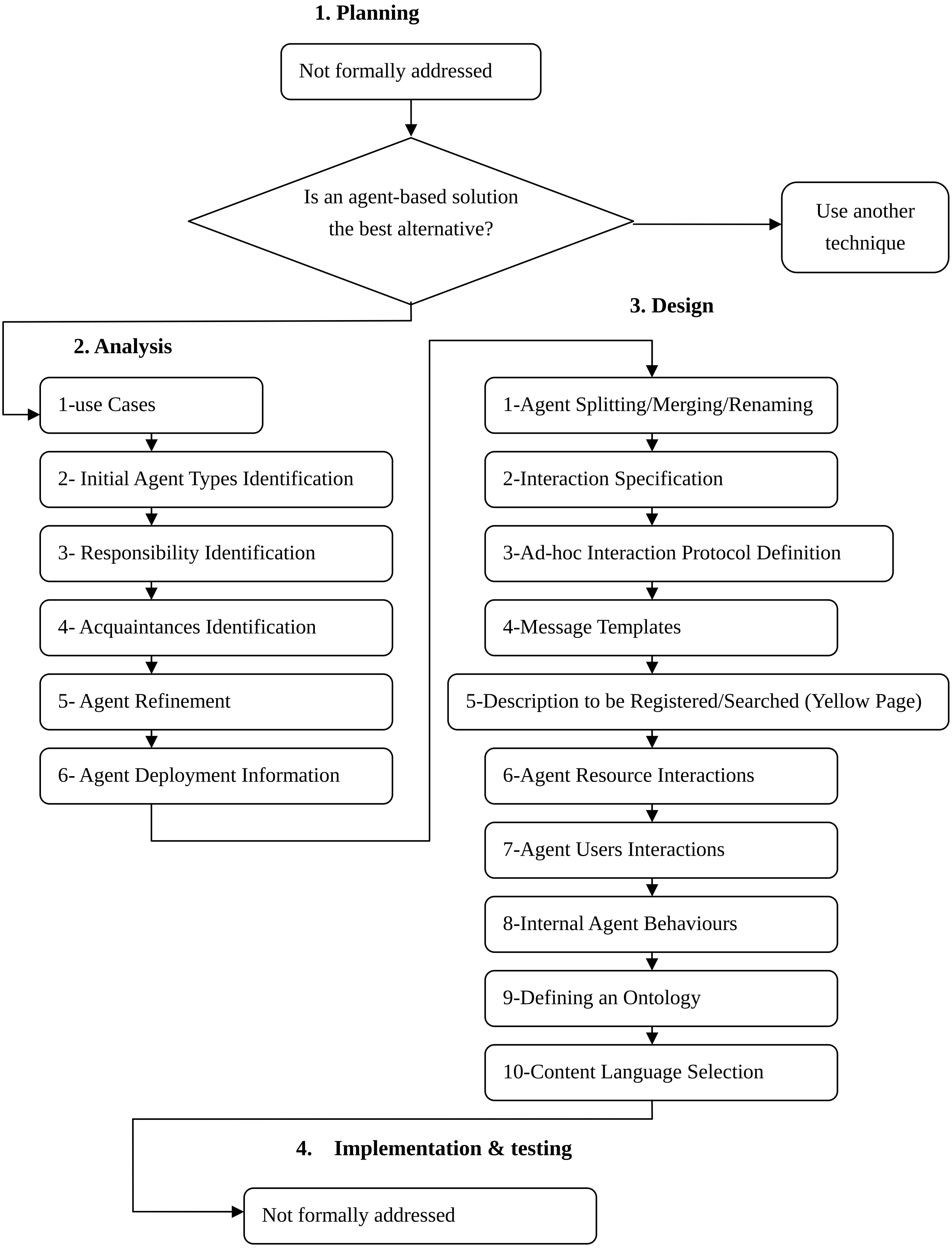

1. Planning
Not formally addressed
Is an agent-based solution the best alternative?
Use another technique
3. Design
2. Analysis
1-use Cases
2- Initial Agent Types Identification
3- Responsibility Identification
4- Acquaintances Identification
5- Agent Refinement
6- Agent Deployment Information
1-Agent Splitting/Merging/Renaming
2-Interaction Specification
3-Ad-hoc Interaction Protocol Definition
4-Message Templates
5-Description to be Registered/Searched (Yellow Page)
6-Agent Resource Interactions
7-Agent Users Interactions
8-Internal Agent Behaviours
9-Defining an Ontology
10-Content Language Selection
4. Implementation & testing
Not formally addressed

Figure 8-1 Overview of the agent design method and steps (Nikraz et al. 2006)

Since this research is intended to apply an agent method to handle shared understanding and complex communication grounded on a shared 3D environment, rather than adopting a system design life circle for a fully-functional multi-agent system, only higher-lever identification and interaction among agents are focused on here. The reconciliation of multiple users' perspectives as a case study is designed to demonstrate the complex communication.

8.2 Agent Identification

From the case study in Section 4.1.1 and mock up group meeting scenario in Section 4.1.2, different users work in a virtual meeting with a shared 3D environment. These users may play different roles, such as Chair, participant (presenter) and audience. They also have different behaviours, such as operation of the 3D environment, communication among users, and making deals. These roles and behaviours can be presented as agents.

There are three types of agents proposed in this multi-agent layer: Assistant agent, Registration agent and Transducer agent (T-agent) (See Figure 8-2). Each distributed client, which is mentioned in Section 7.4, has an assistant agent and a T-agent. The complete multi-agent layer has only one registration agent.

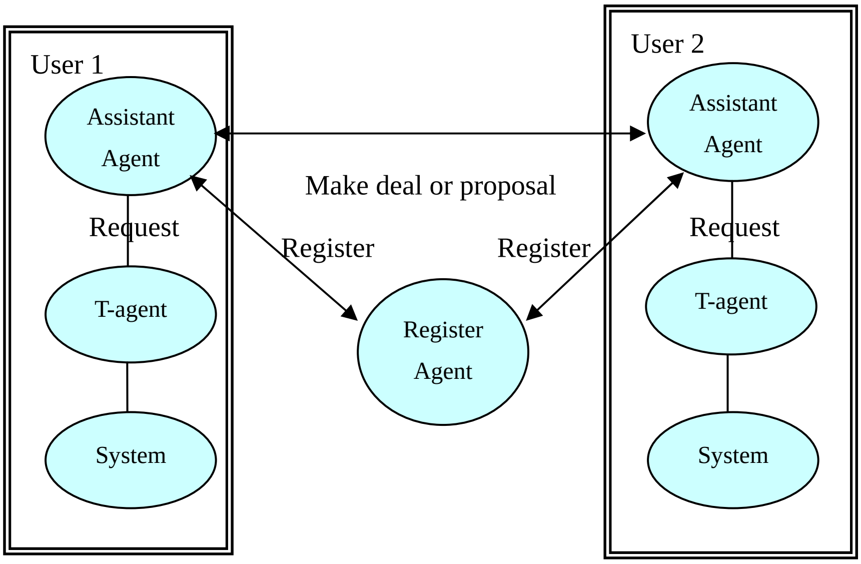


Figure 8-2 Agents and the interactions

These agents may have different responsibilities and roles, as summarized in Table 8.1. The assistant agent is launched by its client and is responsible for complex communications with other assistant agents after it registers the interested topic to the registration agent. Complex communications include making deals or proposals. The assistant agents themselves also play different roles, such as chairing the meeting, making a deal or approving a proposal, while other agents just have the right to audit and discuss in the meeting. The registration agent is responsible for setting the meeting and topics and other agents' registration and deregistration. This agent is launched by the client who is the first one to set up and chair the session. The transducer agent is responsible for communication between the shared 3D GIS environment and assistant agent.

Usually for a normal multi-agent system, there are more agents than this system. The related agents are designed according to the different use cases. In this system, because there are lots of functions such as work environment, shared storage space and system state already provided by the shared 3D GIS environments, the agents can be reduced to a minimal number.

Table 8-1 Agents identifications

Agent type	**Responsibilities**
Assistant Agents (launched by local user)	• Register to registration agent for a specific topic • Communicate with other assistant agents • Make deal or proposal with other agents
Registration Agent	• Set up meeting topics • Handle users registration/ deregistration
T-agents	• Communicate with assistant agent and shared 3D environments

8.3 Agent Interactions

From the case study in Section 4.1, some typical interactions can be identified. For example, for the **Search and Registration** use case, when a user sets up a meeting, this user will wait for other users to join in the meeting. Suppose that the meeting is set up by the registration agent. There are two types of interactions between the registration agent and assistant agent: request interaction and subscribe interaction.

In the request interaction, the assistant agent sends a request to the registration agent to join in a topic and the registration agent approves it. This seems simple, but many situations may occur. For instance, the assistant agent has sent the message, but the registration agent may not receive it, or the registration agent receives the message but may not understand what it means. As well, the registration agent may refuse the request or agree to the request and send the assistant agent related information.

The detailed workflow is as follows: 1) the assistant agent sends a request message to the registration agent to join in the topic; 2) the registration agent receives the request and replies with a rejection; or 3) the registration agent receives the request and replies with an acceptance and sends related meeting details; or 4) the registration agent does not receive the request messages.

In the subscribe interaction, the assistant agent subscribes the topic. When the meeting about the topic is set up, the assistant agent automatically joins in the meeting. The subscribe interaction also faces the same situations as request interaction when the subscription is set up.

Fortunately, FIPA has a clear specification to describe all the situations for these kinds of interactions. For example, in the request interaction, FIPA defines all possible situations which may happen when two agents communicate. Of course, not all the situations are necessary to be handled in the real work.

The agents' interactions in these multi-agent layers can be abstracted into several types, such as Request, Subscribe and Contract Net according to FIPA Interaction Protocol (see Table 8-2).

Table 8-2 Agent interaction

Interaction	IP (See FIPA)	Describes	Who
Initiate Registration	Query	Assistant agent register interested topic to registration agent	Assistant agent, registration agent
Subscribe Registration	Subscribe	Assistant agent may subscribe the interested topic to registration agent when the topic is available	Assistant agent, registration agent
Approve Proposal/negotiation	Iterate Contract Net	When a proposal is discussed for approve, the assistant agent approves the proposal.	Assistant agent, Assistant agent
Get information from shared 3D environment	Request	Assistant agent get shared information from shared 3D environment through T-agent	Assistant agent, T-agent

8.4 Shared Understanding

One challenge in designing SC3DGIS relates to shared understanding. Specifically, the distributed parties, such as the distributed clients, register server, users and multi-agents, need to understand each other. In Chapter 5, two issues are discussed: what contents need to be shared and how to present these contents. The shared contents are abstracted as a social collaboration model and are described in detail as an application-level ontology presentation. In this section, the detailed implementation method is discussed.

Given the two-layer structure, the contents, which present the outside world, are presented as a string or a sequence of bytes, while the contents inside the above two systems are presented as Java objects. Thus, when communicating between the contents and systems, there are two reversed procedures: parsing and encoding procedure. In the parsing procedure, the string- or byte-based information will be translated into information such as Java class objects which can be understood in agent systems and shared 3D GIS clients. In the encoding procedure, the internal class objects will be translated to strings or bytes, which can be understood by the outside world. The whole procedure may include the following steps (see Figure 8-3). First, the application ontology (SCM) is created with tools like Protégé (Stanford 2009), an ontology editing language, by application users and domain experts. Then the ontology is transformed into Java classes by an add-on tool such as beangenerator, which is a tool to convert ontology to Java Beans. These classes are embodied in a multi-agent system and SC3DGIS clients. Finally, a parsing and an encoding procedure are set up between contents and systems.

In this prototype system, JADE is used to implement a multi-agent system and ontology-based contents sharing. Agent Communication Language (ACL) is used to carry out the parsing and encoding processes.

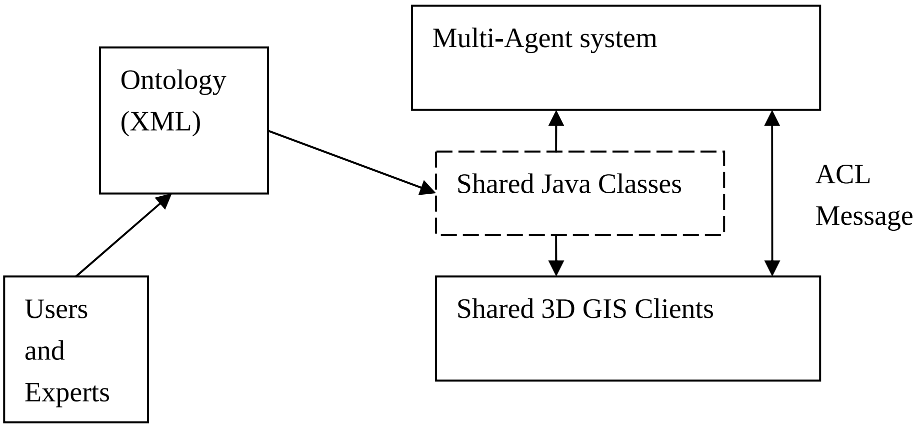


Figure 8-3 Procedure of sharing contents with ontology and ACL

8.5 Reconciliation of Conflicting Actions

This section uses the agent method to solve a collaborative problem, reconciliation of conflicted actions, as a case study. This case study demonstrates how the agent method can make a complex deal and communicate within a shared 3D environment.

Keeping a consistency state is critical for the distributed collaborative system. In Section 7.5, the Floor Control mechanism is used to keep the system in consistency. The Floor Control will decide who can use the system and at the same time prevent other users from using the system so that conflicts can be avoided. This method is also called the **avoiding method**. However, in a collaborative environment, discussion and negotiation are always important options. Another method, therefore, arises and is termed **reconciliation method**.

The reconciliation method applies the negotiation process to achieve reconciled outcomes from different perspectives. Multiple negotiations are very complex processes, as are the multiple reconciliations. Chen et al. (2008) extended Speech Act Theory (SAT) to address spatial aspects in a negotiation context. The following coordination process (see Figure 8-4) modeled by extended SAT is used to reconcile inconsistent perspectives.

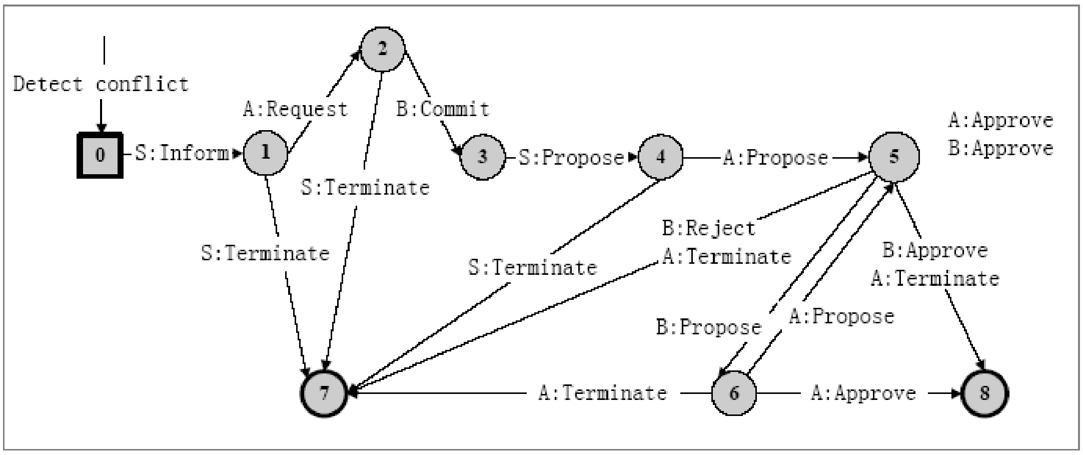


Figure 8-4 Reconciliation process of inconsistent perspectives

The SAT-based method solved the spatial objects' conflicts. However, two problems are not mentioned in this method: 1) how to solve the conflicts of system status rather than spatial objects, 2) how to simulate the generic collaborative process such as multiple negotiation which is a very complex procedure, especially in the client server architecture.

The agent-based method and social collaboration model (SCM) provide solutions to the above two problems. First, since SCM represents all the actions and system statuses which can be objects for negotiations. Second, the agent method supports standard communication protocols, such as Query, Request and Iterate Contract Net. The negotiation process can be extended from the above standard protocols. Since the process of multiple negotiations is beyond the scope of the research, a simple negotiation process is used to demonstrate how the agent method solves or reconciles multiple users' conflicts of perspectives.

Suppose that an action, navigating the 3D view to a specific location, is to be reconciled in a shared 3D view. This action and related parameters are sent from the shared 3D environment to the agent systems. Then, the iterative negotiation process is implemented in the agent system. When the process is finished and agreement is achieved, the action will be sent to the 3D environment to be carried out. If an agreement is not achieved, the null action is sent to the shared 3D environment and no action is implemented. The iterative process of negotiation among agents is as follows:

1. The initiator sends a message to all the other users for approval. This message includes an action, for example, to move the 3D view to a specific location, with related parameters.
2. Other users, called participants here, may refuse or accept this action and change part of the action within a deadline.
3. If the initiator receives a refusal from any participant, the conciliation process is terminated and no agreement is achieved.
4. If the initiator receives the changed actions from the participants, the initiator may change the actions again and send them to all participants.
5. Steps 2 is repeated until a refusal is received or all the participants reach an agreement.
6. The agreed action is sent to the shared 3D environment to implement.

The iterative process of negotiation may follow an extended Iterate Contract Net. After an agreement is reached, the final action is achieved and sent back to the shared 3D environment to implement. Otherwise, the final action is set to NULL and no action is implemented in the shared 3D environment. The advantage is that the agent system, such as

JADE, supports this negotiation process automatically. Iterate Contract Net protocol as an open standard is supported in many agent-based systems.

8.6 Integration of Multi-agents with Shared 3D View

The overall architecture (see Figure 8-5) includes two layers: a shared 3D GIS environment and a multi-agent based social structure layer.

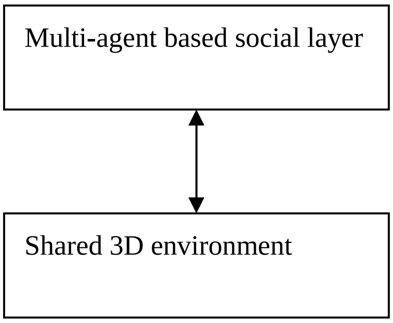


Figure 8-5 Two ties structure for SC3DGIS

The relations between the two layers are loosely coupled. The transducer approach (Nikraz1a et al. 2006) is adopted to integrate multi-agent part into the SC3DGIS in this prototype system. The transducer approach provides the interface agent, called as the transducer agent (T-agent), between the legacy system and the multi-agents system. The two layers share the same ontology-based social collaboration model. According to this method, the whole system is divided into two parts: a multi-agent part and a 3D GIS part (see Figure 8-6). The transducer agent serves as an interface between the 3D GIS and the multi-agents. The distributed 3D GIS is responsible for meeting the basic 3D GIS requirements, while the multi-agents are responsible for handling more complex interactions such as multiple communication and structured interactions.

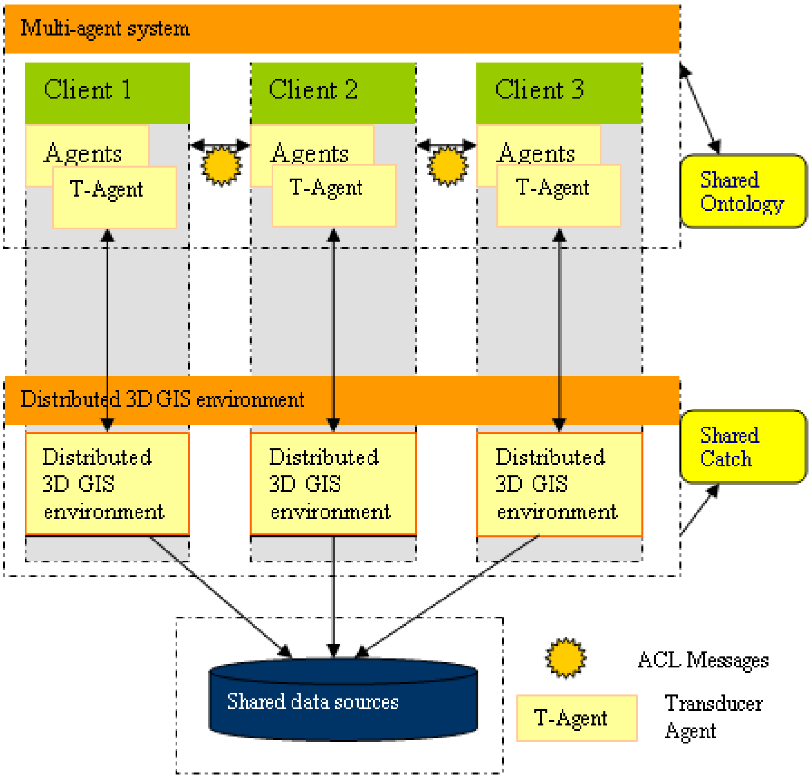


Figure 8-6 Transducer approach integrates multi-agents to SC3DGIS

The multi-agents are divided into several agents to implement specific tasks in a distributed pattern. Using Agent Communication Language (ACL), for example in Java Agent Development Framework system (Telecom Italia 2008), these agents can communicate and interact with each other while through ontology, they can share vocabulary and interaction protocols and achieve common understanding. The 3D GIS part is an independent (legacy) distributed system. The shared cache is a shared virtual memory which every client can access as well as update the shared system status and therefore keep the whole system consistent.

In order to share the same understanding, these agents need to share the same language, vocabulary and protocols. In this prototype, JADE is used to develop the multi-agent system part in which the FIPA communicative acts and coder/decoder classes for semantic languages (SL) are followed. Through defining application-specific ontology, the vocabulary and semantics for the content of the message are shared among agents and the distributed 3D GIS. Figure 8-7 shows detailed messages passing procedures between agents and the 3DGIS environment: 1) the agents initiate themselves when they get parameters from a shared cache in a 3D GIS environment; 2) the agent Chair and agent participant(s), for example, communicate with each other and make a decision; 3) the agent Chair transports the decision to the shared cache; and 4) the shared cache notifies the decision results to all the agents taking part in the session.

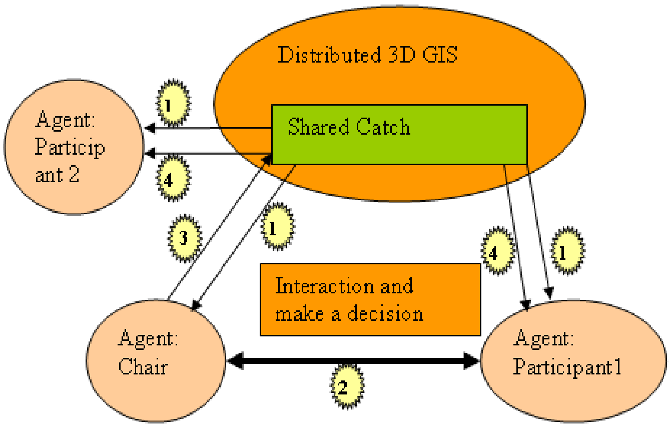


Figure 8-7 Data transportation between 3DGIS and agents

8.7 Summary

The multi-agent method was designed and implemented to support multiple users' complex communications sitting on a shared 3D GIS view. Introducing a multi-agent method in a 3D

GIS field for real-time collaboration is also a first attempt. With agent assistance, users not only can work on his/her own private 3D view but also deal with more complicated collaborative processes which may relate to negotiation and multi-round interactions.

Chapter 9 Prototyping and Evaluation

This chapter describes how the above designs, such as conceptual framework, social collaboration models, hybrid architecture, and multi-agent architecture design are implemented and evaluated. First, the suitable development tools are tested and selected. Second, the prototype is designed and developed to validate the designs, such as the architecture, consistency and data source issues discussed in the previous chapters. The walkthrough method is used to evaluate the usability of the prototype.

9.1 Development Tools Selection

In the prototyping stage, selection of the developing tools is important. Considering open source software, the following selection criteria are used: 1) is it suitable to work on the Internet? 2) Is it easy to work together with other tools like CSCW tools, 3D tools and other multimedia tools? 3) Is it still active? 4) Is it simple to implement? 5) Can the toolkit be used in a desired integrated development environment (IDE)?

At the beginning of the prototyping, JBuilder 9.0 was used as the development environment, as well as Java3D API for 3D rendering tools, and Java Shared Data Toolkit (JSDT) (Sun 2004) for data multicasting and communication. Java Web Start technology was used for system deployment. JSDT provides a common interface for general multiparty communications, beneath which a wide variety of implementation technologies can be employed. A range of different protocols can be hidden within the implementation of this interface, including standards-based, multi-party communications protocols, e.g., T.12x, custom protocols based on standard networking interfaces, TCP/IP, and arbitrary proprietary protocols.

However, JSDT became inactive when the initial prototype of GeoLink3D was close to completion. In addition, JSDT does not support JDK1.5, which is required by Java3D 1.5 (only works with JDK 1.5 and higher) to obtain better performance. An alternative was sought to replace JSDT with JGroups (JGroups 2007), which meets the requirements and has been active. JBuilder 9.0 was also replaced with Eclipse 3.1, which is an open source development environment for Java. Java Agent DEvelopment Framework (JADE) was used to develop the multi-agent layer. The reason for adopting JADE is not only its functionalities which meet the system's requirements, but also its active development community, enriched documentation and numerous use cases.

The performance issue arises through asking such questions as "why not use C++ to handle 3D rendering and multicast transportation, since Java is slower in handling 3D and multicast prototype?" First, it is believed that using Java to handle 3D and multicast transportation benefits the Web-based applications and has potential support from open source community in Java3D and JGroups. Second, the purpose of this prototype is to validate the conceptual design consideration which is independent from developing tools. The developing tools selected can rapidly develop a prototype and therefore speed the development life cycle. Third, this prototype is an attempt to develop a light-weight system instead of a "heavy" enterprise framework.

9.2 Prototype System Design and Configuration

The purpose of the prototype is to validate the conceptual design and the multiple user experience of SC3DGIS, and to explore whether or not current technology development can satisfy the design requirement.

The overall architecture of the prototype was presented and discussed in detail in Section 8.6. The prototype system configuration, which includes one HTTP server, one data server and four clients, is illustrated in Figure 9-1. The HTTP server hosts the prototype system which includes the GeoLink3D Registration server, a proxy data server and a client download link.

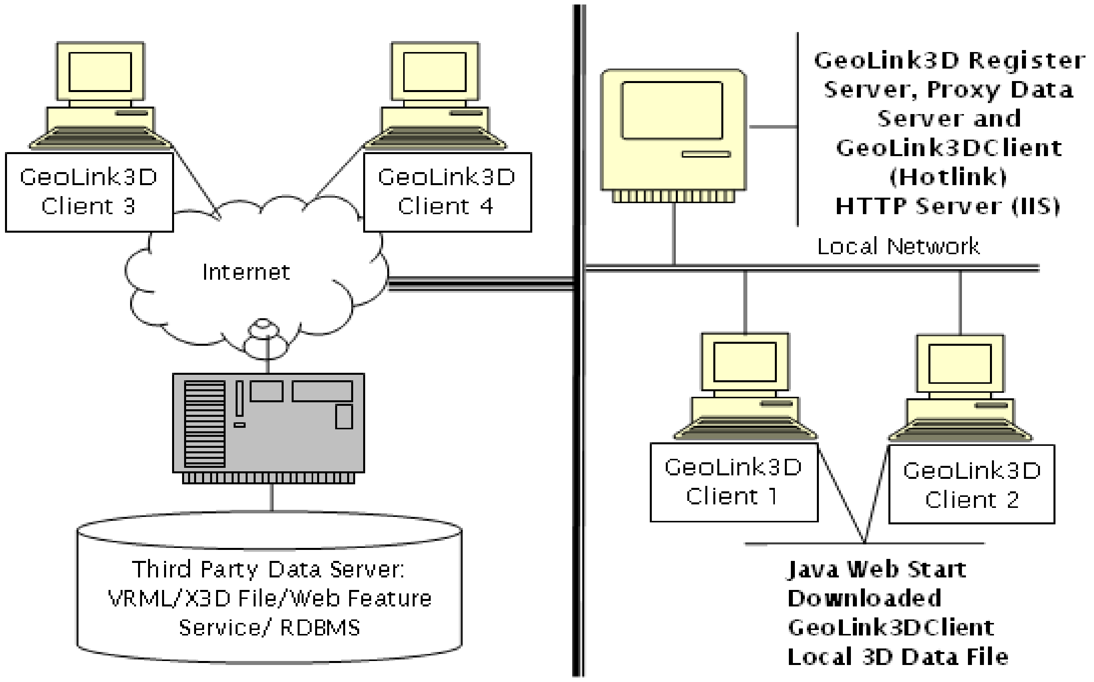

GeoLink3D Client 3
GeoLink3D Client 4
Internet
GeoLink3D Register Server, Proxy Data Server and GeoLink3DClient (Hotlink) HTTP Server (IIS)
Local Network
GeoLink3D Client 1
GeoLink3D Client 2
Third Party Data Server: VRML/X3D File/Web Feature Service/ RDBMS
Java Web Start Downloaded GeoLink3DClient Local 3D Data File

Figure 9-1 SC3DGIS system configuration

The third party data server could be any HTTP server providing X3D or VRML files for downloading, Web Feature Service providing 3D data services (which is not implemented in this prototype), or relational database, such as Oracle database. The system can also connect to a FME service and then to an Oracle Spatial database to transform the 3D data to VRML or X3D format. The FME Server is used as a shared proxy server to provide 3D data to every client (Oren 2008). Although these servers are not part of the prototype system, they play a very important role jn data loading. It is critical for the prototype to provide interfaces to all data sources.

The clients, who could be in a local network or on the Internet, need to install JDK1.5 or above and Java Web Start to run the client. Specifically, after installing the above two components, the users can go to the HTTP server website and click the hotlink for a Geolink3D client. This client will be downloaded and installed in the user’s desktop with Java Web Start. If there is no update for the server, users do not need to reinstall the client every time they access the system.

A controlled laboratory environment was used for the prototype development, which includes a Dell server, 2 high-end workstations and a few high-performance desktops connected to the Internet via an isolated local area network in the SIMAL lab. The workstations were used to code and test the prototype, and the server was used to deploy the developed prototype for further testing. One of the important benefits of using such a controlled environment is that the prototype testing may not be affected by bandwidth limitation.

9.3 Walkthrough Evaluation and Experience

9.3.1 Walkthrough Evaluation Method

Evaluation of groupware has received attention from researchers in the Computer-Supported Cooperative Work (CSCW) and groupware communities (Pinelle and Gutwin 2002, Baker et al. 2002, Baeza-Yates and Pino 2006). However, the evaluation is still considered a difficult problem, and many researchers feel that the only way to get a true picture of a groupware system is to study it in an actual application context with real users. Although these kinds of field methods are able to contextualize the evaluation, they can be both time-consuming and expensive. In addition, they can be difficult or impossible to perform if a system is not fully developed.

Gutwin and Greenberg (2000) claimed that some groupware usability problems are not strongly tied to social or organizational issues, but rather are caused by insufficient or mismatched support for the basic activities of collaboration. They therefore proposed a usability inspection technique to evaluate groupware system without utilizing a real work situation. Because of its advantages in being better able to rapidly test a groupware system compared to quantitative measures or user-based methods, this method was mainly used to test the usability of shared workspaces with the predefined criteria, Mechanics of Collaboration. The Mechanics of Collaboration, which are low-level actions and interactions such as communication, coordination, planning, monitoring, assistance, and protection, must be carried out to complete a task in a shared manner. The framework also includes gross measures of these mechanics: effectiveness, efficiency, and satisfaction. Steves et al. (2001)

gave a rating for each mechanism of collaboration (see Table 9-1).

Table 9-1 Rating and associated descriptions (Steves et al. 2001)

Rating	Description
+2	Very successful, few reports of problems
+1	Often successful, but some awkwardness
0	Adequate: no major problems or major benefits
-1	Useful in some situations but many drawbacks
-2	Rarely successful with many failures
N/A	Not enough information to rate

In order to add context to a groupware usability evaluation, Pinelle and Gutwin (2002) designed a groupware walkthrough method which adds contextual information about the work domain, enabling evaluators to test the interface in relation to real tasks and real users. Walkthrough is a form of software peer review in which a designer or programmer leads members of the development team and other interested parties through a software product, and the participants ask questions and make comments about possible errors, violation of development standards, and other problems (IEEE Standard, 1998).

The technique is a modification of cognitive walkthrough including the considerations for the complexities of teamwork. There are two components for the groupware walkthrough: a task model for identifying and analysing real-world collaborative tasks, and a walkthrough process for assessing a system's support for those tasks (see Figure 9-2).

- Scenario
  - Tasks
    - Individual subtasks
    - Collaborative subtasks

Figure 9-2 Hierarchical task model (Pinelle and Gutwin 2002)

"Scenarios are descriptive formalizations of the work users carry out in the real world, typically containing multiple tasks and providing contextual information about the users and the circumstances under which the tasks are commonly carried out. ***Tasks*** are the building blocks of scenarios, and may be explicitly stated in the scenario activity description. ***Subtasks*** are categorized as either individual (i.e. task-work) or collaborative (i.e. team-work). Collaborative subtasks are specified using the mechanics of collaboration." (Pinelle and Gutwin 2002)

9.3.2 Test and Evaluation Procedure

In this research, two walkthrough scenarios were used to test the collaboration usability of the prototype. One is about multiple users working on a shared 3D environment; another is about multiple agents working on a conflict perspective.

Three teams grouped by three students (names are hidden because of a related policy) from different backgrounds (geomatics, geography, mechanics and economy), ages and gender took part in this evaluation. The detailed names and backgrounds are listed in Table 9-2.

Table 9-2 Test team profiles

Team	Name	Education background	Age	Gender	Has 3D knowledge
Team 1	Participant 1 (leader)	Geomatics (PhD student)	35	Male	Yes
	Participant 2	Geomatics (PhD student)	30	Male	Yes
	Participant 3	Geomatics (Msc student)	25	Male	No
Team 2	Participant 4 (Leader)	Geomatics (PhD student)	37	Male	Yes
	Participant 5	Computer science (Msc student)	34	Male	No
	Participant 6	Computer science (college student)	21	Male	No
Team 3	Participant 7 (Leader)	The author of the thesis (PhD student)	38	Male	Yes
	Participant 8	Mechanical engineer	34	Male	No
	Participant 9	Economy (working in finance)	36	Female	No

Every team has a team leader who plays the Chair role for the evaluation. The leader can assign different roles for other users through floor control functions. The detailed work flow and test will follow Walkthrough 1 and Walkthrough 2 procedures. Walkthrough 1 tests how the multiple users work on a shared 3D view. Walkthrough 2 tests how the users discuss a 3D viewpoint in a shared 3D view through multi-agent support. The detailed procedures are presented in Sections 9.3.3 and 9.3.4.

During the test period, the users rate the mechanics of collaboration in Table 9-3 according to the rating system in Table 9-1. Detailed evaluation input from the testers can be seen in Appendix 4. In Table 9-3, the term “Mechanic” not only refers to Mechanics of Collaboration (Gutwin and Greenberg 2000) but also presents other SC3DGIS requirements discussed in the conceptual framework.

Table 9-3 Content of evaluation

Mechanic	Description
Explicit Communication	Intentional provision of information, either through speech, text, or gesture
Coordination of Action	Synchronizing actions and managing access to shared resources
Planning	Division of labour, reserving areas of the workspace for future use, or plotting courses of action
Monitoring	Gathering information given off by others through consequential communication or feedthrough
Assistance	Provision of help to one another, either upon request or opportunistically
Protection	Actions taken to prevent change to or deletion of a person's existing artifacts and work
Internet Access	Web based system and Internet data source
3D GIS Data Sources	GIS data source support
Structured Society	Support roles and complex interaction

9.3.3 Walkthrough Test 1: Multi-users Operate Shared 3D View

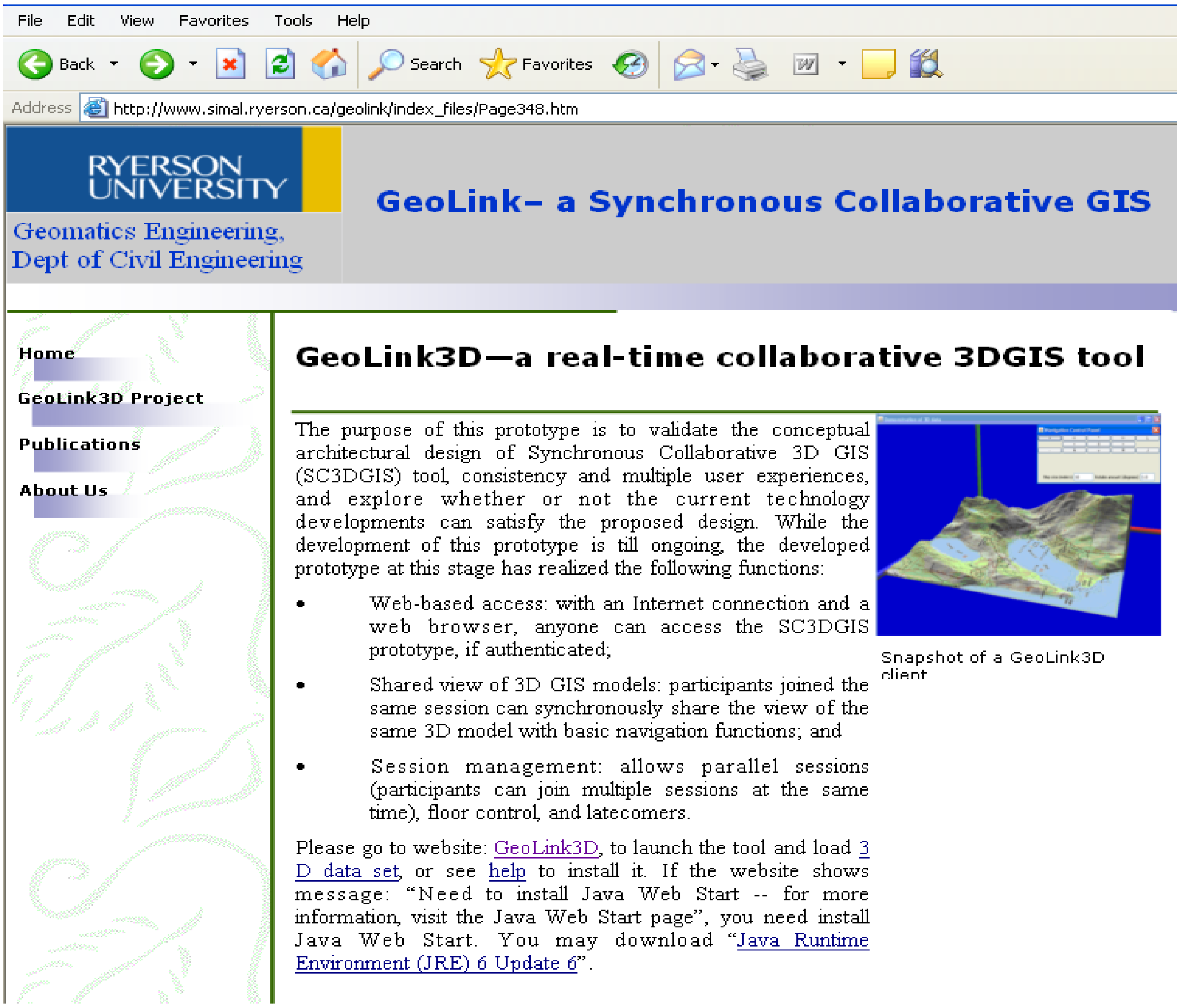
This scenario shows how the geographically-distributed participants work collaboratively on a shared 3D GIS environment according to the case study (Section 5.1). There are mainly five tasks to demonstrate the complete procedure: 1) registering to the system; 2) assigning roles to other users through floor control; 3) uploading private files to the shared folder; 4) loading different data sources to the system; and 5) operating the shared 3D view for presentation and argument. Steps 1 and 2 are based on the **Search and registration use case (Set environment)**. Steps 3 and 4 are based on the **Handling shared data source use case (Shared data source).** Step 5 is based on **the Sharing operations in 3D GIS view use case**

(Shared 3D GIS, shared operations). The **Discussion and approving issues use case (Negotiation process)** can be seen in Walkthrough 2.

1) Registration with different roles

The users are assigned different roles, such as Chair, participant and audience. These roles have different privileges in the operation of the system. The Chair role controls the shared 3D view and has the right to change other users' roles. The participant role has the right to operate the shared view, such as zoom in, zoom out and load data. The audience role just has the right to access and watch other users' operations but cannot operate the shared view. The Chair role and participant role have three operation statuses: operating, applying and idle, while the audience role always has the idle status. The operating status means the user is operating the system; the applying status means the user is applying for system operation rights; the idle status means the user just watches the others' operations. The user who is the first to accesses the system is assigned the Chair role by default. Other users are assigned the participant role by default. The floor control tool can mange the roles and operation rights (details of floor control functions can be seen in procedure 2).

There are two versions for the prototype. The previous version has a register server sit on the server side. Each client needs to register itself to join the same session. The later version puts the registered server in every client. The registration process is automated (see Figure 9-3 for the entry interface of the prototype). In order to address the registration process clearly, the following steps show the registration procedure for the old version.



Address http://www.simal.ryerson.ca/geolink/index_files/Page348.htm

Geomatics Engineering,
Dept of Civil Engineering

GeoLink– a Synchronous Collaborative GIS

GeoLink3D—a real-time collaborative 3DGIS tool

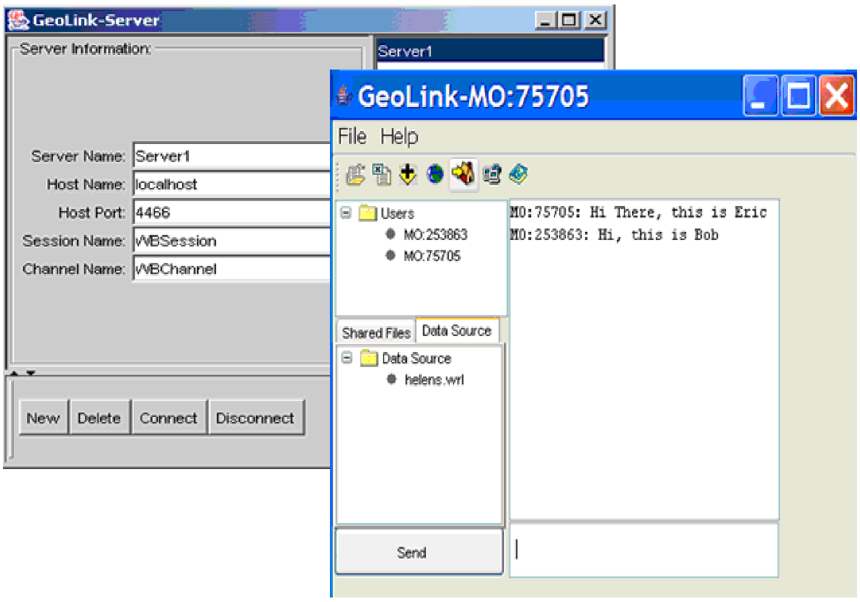
The purpose of this prototype is to validate the conceptual architectural design of Synchronous Collaborative 3D GIS (SC3DGIS) tool, consistency and multiple user experiences, and explore whether or not the current technology developments can satisfy the proposed design. While the development of this prototype is till ongoing, the developed prototype at this stage has realized the following functions:

- Web-based access: with an Internet connection and a web browser, anyone can access the SC3DGIS prototype, if authenticated;
- Shared view of 3D GIS models: participants joined the same session can synchronously share the view of the same 3D model with basic navigation functions; and
- Session management: allows parallel sessions (participants can join multiple sessions at the same time), floor control, and latecomers.

Please go to website: GeoLink3D, to launch the tool and load 3 D data set, or see help to install it. If the website shows message: "Need to install Java Web Start -- for more information, visit the Java Web Start page", you need install Java Web Start. You may download "Java Runtime Environment (JRE) 6 Update 6".

Figure 9-3 GeoLink3D prototype entry link

First, this Chair user, who is the first one to access the system and set up the session, clicks on the hotlink of the GeoLinkServer in the website to launch the register server. The register server will be downloaded in this user's machine and run as a Java Web Start application. The register server of the prototype system can manage the sessions and servers (see Figure 9-4) from one machine or several different machines. The session identification is composed of several items: Server Name, Host Name, Host Port, Session Name and Channel Name. These five items together can uniquely identify any session in a machine, which means by entering the five items, a new session can be created. The server can also manage the session with Delete, Disconnect and Connect operations. Any client can access the session by inputting the five items.



Server Name: Server1
Host Name: localhost
Host Port: 4466
Session Name: WBSession
Channel Name: WBChannel

Users
- MO:253863
- MO:75705

Data Source
- helens.wrl

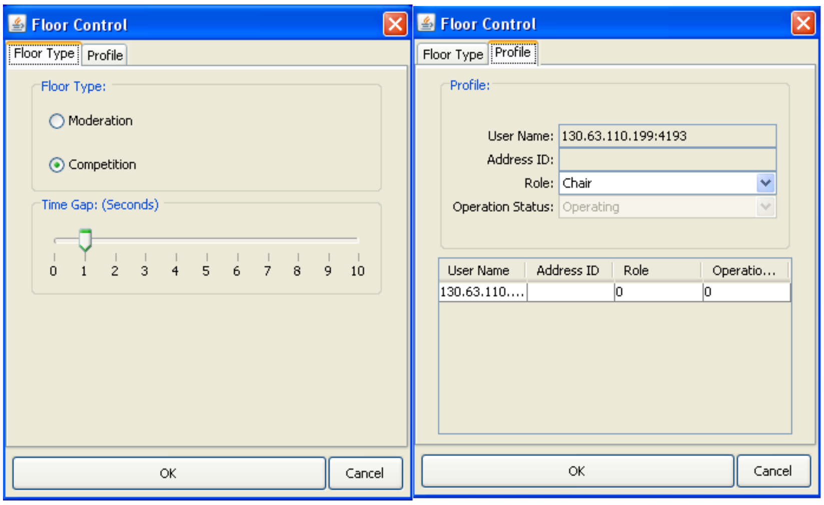
```
MO:75705: Hi There, this is Eric
MO:253863: Hi, this is Bob
```

Figure 9-4 Snapshot of a register server and main interface of the collaboration workspace

The chair then clicks on the corresponding hotlink of the website to launch the GeoLink3DClient. This client will be downloaded to the user's local machine and also run as a Java Web Start application. The Chair inputs his IP address and session name to set up the session. Other session information such as host port and channel name can be created automatically. The session information can be seen in the register server. The other participants then click on the same hotlink of GeoLink3DClient to launch the client and input this Chair's IP address and session name to register itself.

2) Floor Control and roles assignment

The floor control function manages roles and operation statuses (see Figure 9-5). Two mechanisms for floor control are implemented: detection floor mechanism and grant floor mechanism corresponding to the interfaces of Competition and Moderation respectively.



○ Moderation

● Competition

User Name: 130.63.110.199:4193

Address ID:

Role: Chair

Operation Status: Operating

User Name	Address ID	Role	Operatio...
130.63.110....		0	0

Figure 9-5 Snapshot of floor control

For the Competition option, there is a 10-second limit timer to control the period of competition. For example, if the timer is set to 3 seconds, the participant, who is operating the shared view and controlling the floor, will lose the control if he/she does not have any operation on the floor for 3 seconds. Once the control is lost, this user has to re-catch it as other users do. When no user operates this view, the user who performs the first operation on the shared view will be assigned the operation privilege.

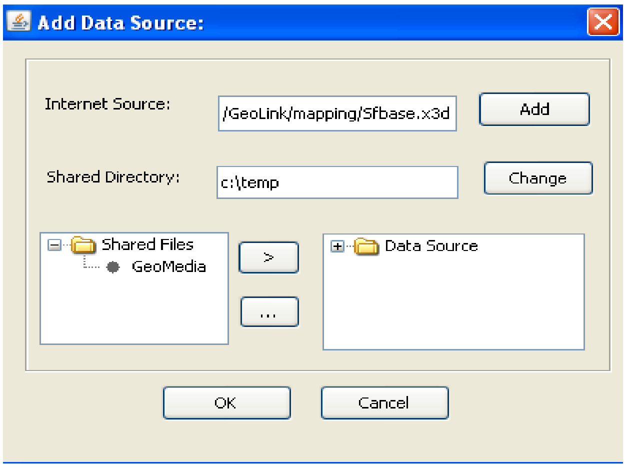
For the Moderation option, the operation right can be assigned by a Chair. The Chair can see who has applied for the operation and then make a decision to grant the operation right. The Chair can also assign his chairing right to others so that if he leaves the session, the shared view still can be active and under control.

3) Upload private files to the shared folder

The users, who are geographically dispersed and may come from different agencies, may hold different data regarding to their disciplines and themes. It is necessary for the data to be shared when multidiscipline issues are discussed. This prototype allows users to upload their own data set into the shared folder so that everyone in the session can access them. When one user uploads his/her file to the shared folder, the file is sent to all the participants automatically.

4) Load different data sources to the system

There are two kinds of data sources for this prototype system: files and Web services. File data source can be in the shared folder or in a HTTP server. As to the files in the shared folder, the user uses a pop-up "Add data source" dialog to select the data file and load it in the prototype. A similar situation happens for files in an HTTP server. The user simply inputs the address of the file in the HTTP server, and the file is then loaded in all the clients (see Figure 9-6).



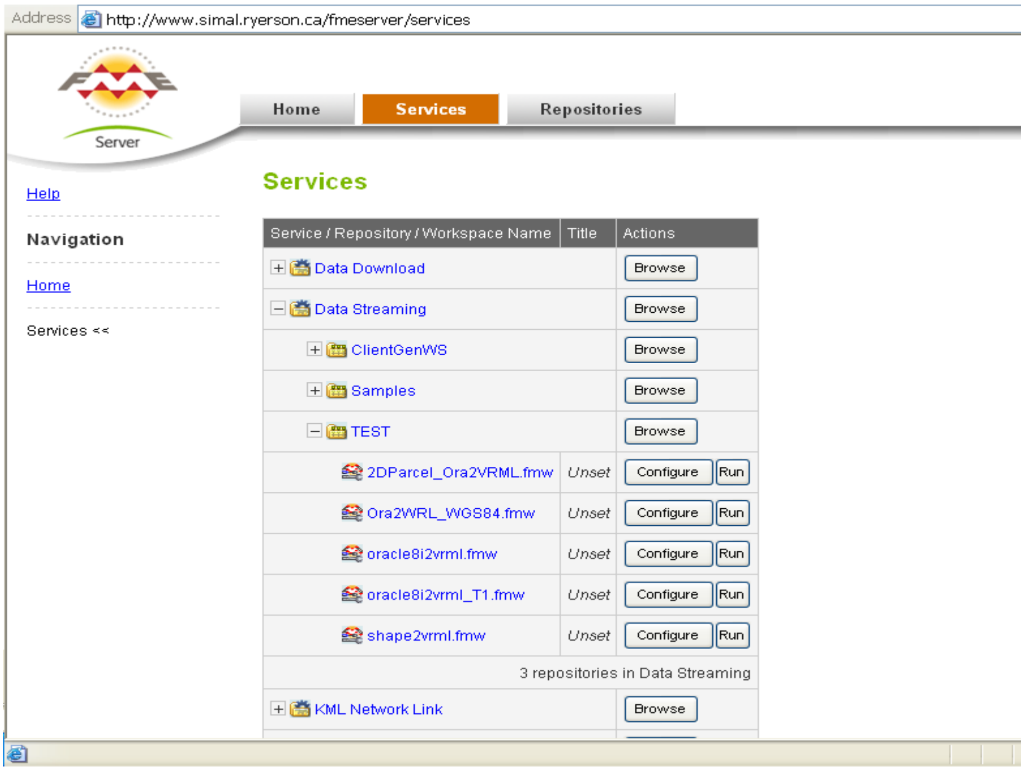
Add Data Source:

Internet Source: /GeoLink/mapping/Sfbase.x3d

Shared Directory: c:\temp

Figure 9-6 Snapshot of Add Data Source interface

The Web service data source is mainly from a FME server. FME connects to the Database server and transforms and delivers the streaming data to the prototype system (see Figure 9-7) (Oren 2008). When the FME server is configured and data sources are also set up, such as loading a file from the HTTP server, the users can input the data address configured in FME and load the data source into the prototype.



Services

Service / Repository / Workspace Name	Title	Actions
+ Data Download		Browse
− Data Streaming		Browse
+ ClientGenWS		Browse
+ Samples		Browse
− TEST		Browse
2DParcel_Ora2VRML.fmw	Unset	Configure Run
Ora2WRL_WGS84.fmw	Unset	Configure Run
oracle8i2vrml.fmw	Unset	Configure Run
oracle8i2vrml_T1.fmw	Unset	Configure Run
shape2vrml.fmw	Unset	Configure Run
3 repositories in Data Streaming		
+ KML Network Link		Browse

Figure 9-7 Snapshot of FME server providing data services (Oren 2008)

5) Operate the shared 3D view

When the users are registered, users' roles are configured, and all the data are loaded, the users can experience collaborative work based on the shared 3D GIS view.

Users can be aware of other users' operations in the shared view and communicate with each other. Clients participating in the same session have the same 3D point of view, which allows geographically distributed users to have What-You-See-Is-What-I-See capability. While the current version of the prototype system provides synchronized control of a 3D GIS model view and basic GIS functions including zoom, pan, rotation, and identification of a GIS data feature (object), the further efforts will be made on collaborative query, analysis, and model construction so that the collaborators in the real work place are able to share these processes and/or outputs from these processes simultaneously.

The shared 3D view, as illustrated in Figure 9-8[4], is a basic 3D model view with collaborative features, such as collaborative viewing. All users in the same session have the same point of view. Any new navigation operation performed on the shared 3D view by any user, e.g., zoom in, zoom out, pan and rotation, are transmitted to all other users (participants) in the session synchronously.

There are three operation states: the operating state, applying operation state and idle state. In the operating state, a green/yellow/gray icon before the user's name shows which user is operating the 3D view. For example, when a user is operating the 3D view, the operation state, shown as green, is an operating state. When the user is applying the operation to the 3D view, the operation state, shown as yellow, is an applying operation state. When the user

[4] Figure 9-8 shows a dataset containing several geological structures, including four lenses, a body of ore or rock, or a deposit thick in the middle and thin at the edges. The data was obtained from MIRARCO-Mining Innovation's Web site (MIRARCO 2008).

has nothing to do, the operation state, shown as gray, is an idle state.

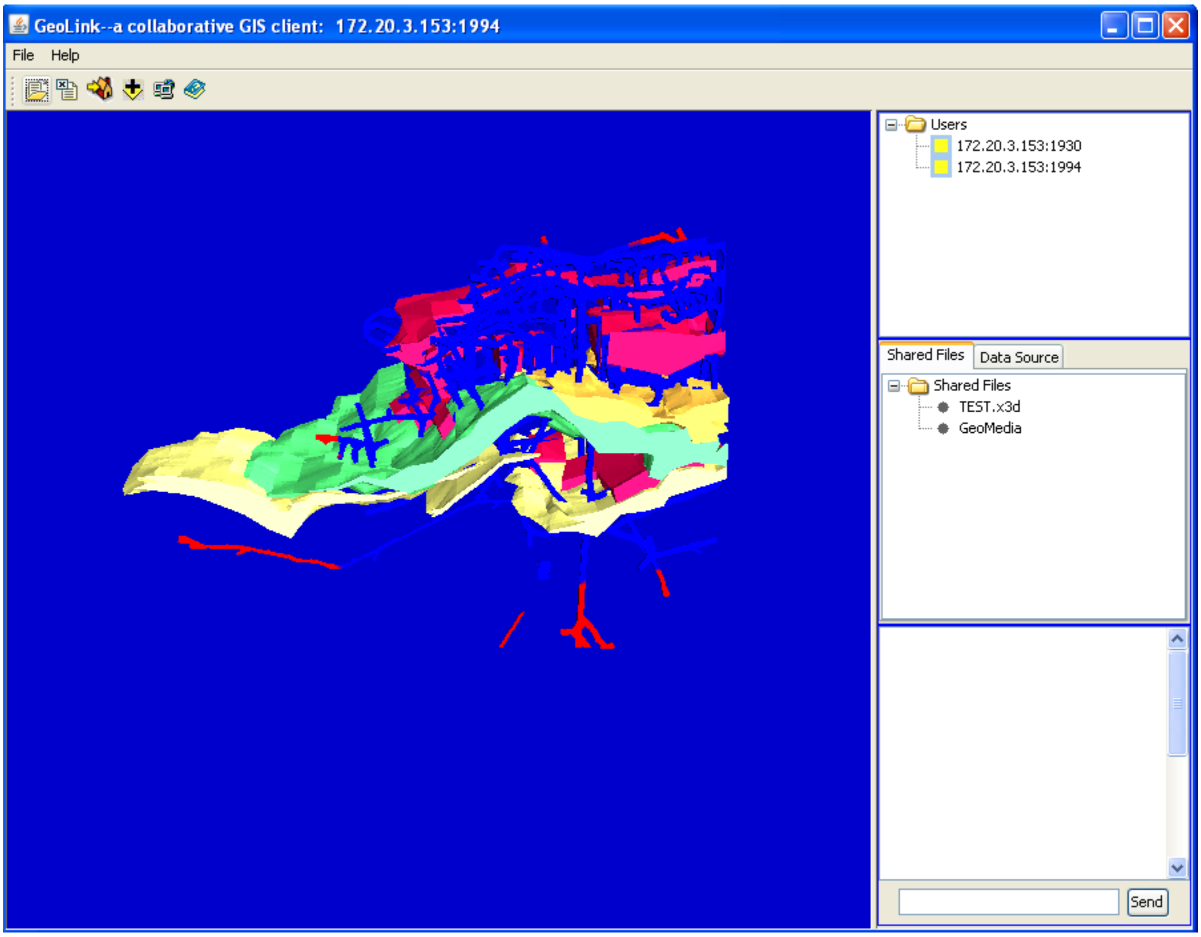


Figure 9-8 Shared 3D view

9.3.4 Walkthrough Test 2: Conflicts Reconciliation

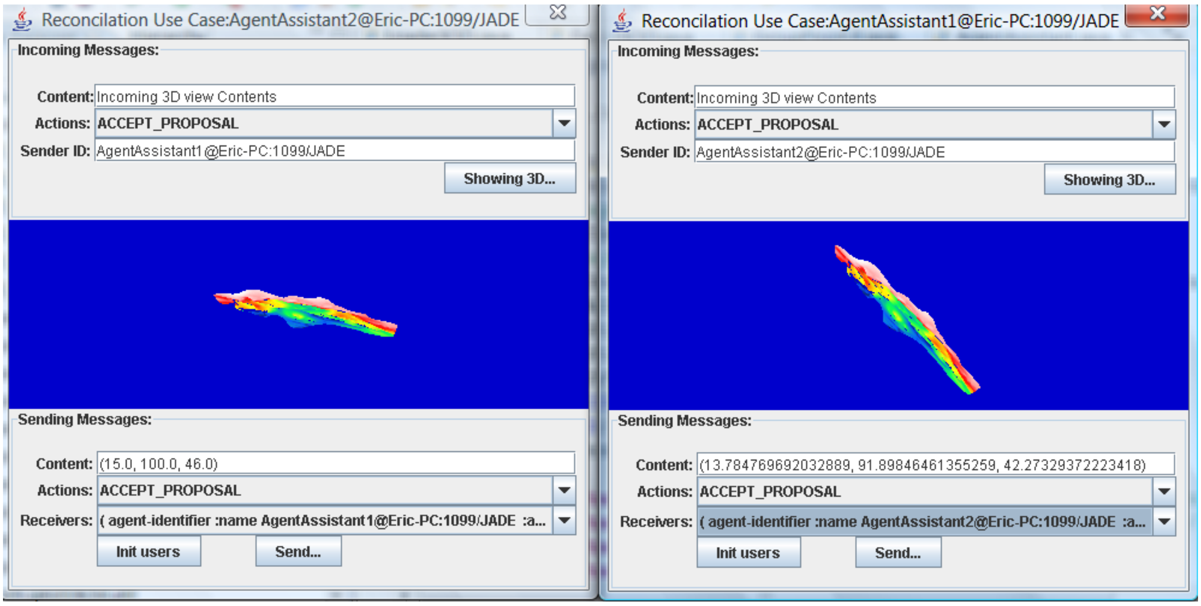
In this walkthrough, how the conflict actions can be solved through reconciliation instead of the avoiding method is demonstrated. As discussed in Section 6.2.5, system consistency can be maintained through avoiding conflict operations/actions. For example, only one user has the capability to operate the shared 3D view, while with the agent method, users can solve conflicting actions through user negotiation.

The negotiation procedure may follow some standard processes in FIPA Protocol. The detailed workflow can be seen in Section 6.3.4. In this walkthrough, “zoom to a specific point of view” is used as a simple action to demonstrate how the conflicted action can be solved through the reconciliation process. More actions can be achieved based on the social collaboration model. The negotiation procedure can be described as follows:

1. The agent system is launched from every client. The Chair role client may launch its registered agent and assistant agent, while other clients just launch their assistant agents.
2. The assistant agents register to the register agent. The registration is an automatic proceeding if an assistant agent is subscribed to service.
3. One of the assistant agents, we suppose this agent is a Chair role, tries to navigate its point of the view to a specific location in the shared 3D view. The transformation parameters are sent to all the other assistant agents.
4. Other assistant agents receive the parameters, make changes, and send these back to the Chair agent.
5. The Chair agent makes a change according to the feedback and resends the parameters to every agent again.
6. After several iterative revision procedures, the Chair agent accepts the changes and sends it to every assistant agent, or the Chair agent rejects the changes and terminates this action.

Figure 9.9 shows the interface of the assistant agent and Figure 9.10 shows the Agent

Management GUI. The assistant agent has a private 3D view in which the different points of view can be checked. After the assistant agent makes a decision regarding the 3D point of view, it sends the view parameters to other agents for negotiation until agreement is achieved or terminated. If agreement is reached, the action to make a point of view for the shared view is sent to the shared 3D environment. The agent management GUI monitors all the agents' life cycles.



Reconcilation Use Case:AgentAssistant2@Eric-PC:1099/JADE

Incoming Messages:

Content: Incoming 3D view Contents
Actions: ACCEPT_PROPOSAL
Sender ID: AgentAssistant1@Eric-PC:1099/JADE

Sending Messages:

Content: (15.0, 100.0, 46.0)
Actions: ACCEPT_PROPOSAL
Receivers: (agent-identifier :name AgentAssistant1@Eric-PC:1099/JADE :a...

Reconcilation Use Case:AgentAssistant1@Eric-PC:1099/JADE

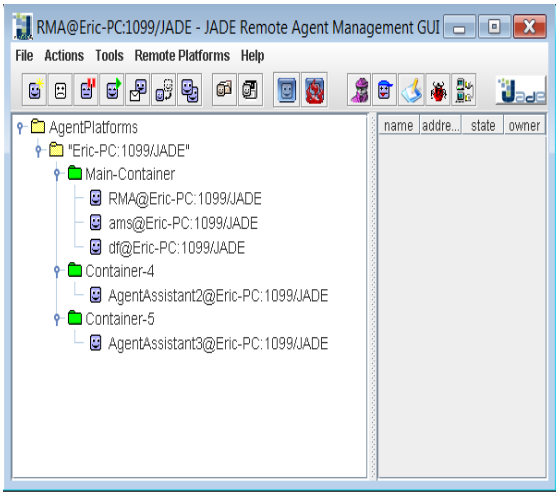
Incoming Messages:

Content: Incoming 3D view Contents
Actions: ACCEPT_PROPOSAL
Sender ID: AgentAssistant2@Eric-PC:1099/JADE

Sending Messages:

Content: (13.784769692032889, 91.89846461355259, 42.27329372223418)
Actions: ACCEPT_PROPOSAL
Receivers: (agent-identifier :name AgentAssistant2@Eric-PC:1099/JADE :a...

Figure 9-9 Snapshot of assistant agent interface



RMA@Eric-PC:1099/JADE - JADE Remote Agent Management GUI

- AgentPlatforms
  - "Eric-PC:1099/JADE"
    - Main-Container
      - RMA@Eric-PC:1099/JADE
      - ams@Eric-PC:1099/JADE
      - df@Eric-PC:1099/JADE
    - Container-4
      - AgentAssistant2@Eric-PC:1099/JADE
    - Container-5
      - AgentAssistant3@Eric-PC:1099/JADE

Figure 9-10 Snapshot of JADE Remote Agent Management GUI

9.4 Results Analysis

The results analysis presents two types of analysis: 1) a rating of the mechanics of collaboration for SC3DGIS, 2) the added value of a third dimension of GIS for the group decision-making process, especially in Geology.

9.4.1 Evaluation of Mechanics of Collaboration

The evaluators were grouped by three participants to follow the walkthrough steps described in Section 9.3. After the walkthroughs were completed, the evaluator gave scores to the mechanics based on Table 9-1 and Table 9-2. The observations of these evaluations are summarized as follows:

1) Web-based access

The prototype is a Web-based system. The data sources also support files on an http server and FME Web services. It is configured within a Local Area Network. The average score is 1.

2) Explicit communication

In Walkthrough 1, the explicit communication is implemented through text chat in this prototype. The participants can talk to all other participants at the same time or talk to just one of the participants. Video and radio will be more helpful to support communication if it is possible. In Walkthrough 2, the complex interactions and communications follow FIPA Protocol, which is a reliable communication protocol for complex agents. The average score is 1.8.

3) Coordination of actions

In Walkthrough Test 1, Task 3 (upload private files to the shared folder) and Task 4 (load different data sources to the system) managed shared files and resources. Task 5 (Operate shared 3D view) demonstrated the capability to synchronize actions. These tasks presented the coordination of actions. The average score is 2.

4) Planning

This planning capability is not shown in this walkthrough. Scores: N/A

5) Monitoring

In Walkthrough Test 1, there are several functions to monitor other users' work. For example, the prototype not only lists the participants' names, which are in the shared workplace, but also shows who is operating the 3D model through different colors (green, yellow and gray). The shared 3D view also follows What-You-See-Is-What-I-See strategy, so that everyone shares the same 3D view. The shared files and data sources also advise everyone which kind of files and data sources can be used. The average score is 2.

6) Assistance

In the Walkthrough Test 1, the shared 3D view and text chat provide assistance support. While because of the communication is based on text chat, the question and answer (Q&A)

can not be presented efficiently. The average score is 1.

7) Protection

In Walkthrough Test 2, every participant has a private view which provides a private workplace and can be seen by this participant only. This view can become a shared view through submitting the view parameters to the shared workplace. A shared file folder and shared data sources provide a place every participant can access. However, there is a problem in that the shared file can be deleted by any participants. The average score is 0.

8) GIS-oriented data modeling

VRML/X3D format has potential to support a 3D surface model and a solid model. Through the FME Web service, any GIS-based 3D data model can be used in this system. There is a problem that the prototype cannot get the attributes for X3D because of the X3D loader's problem. The average score is 1.

9) Well-structured

In Walkthrough 1, Task 2 (Floor Control and roles assignment), the participant with the Chair role can assign different participants with different roles (Chair, operator and audience) and privileges (control, operation and view), which helps the shared 3D environment be well organized and structured. The private view and multi-agent method in Walkthrough 2 can help participants make deal and negotiations. The average score is 1.

According to the observation and results, most of the synchronous collaboration functions are satisfied and successful. When Team 3 was testing the system, even though two of the testers did not have the Geology background, they could work on the walkthrough smoothly with the guide of the team leader. The synchronous collaborative functions, such as the What-You-See-Is-What-I-See feature, text messages and floor control mechanism, play important roles to assist participants in understanding the context and working environment. The protection mechanism may cause problems because the shared file can be deleted by any participant. The planning mechanism is not available for testing because no planning tasks are used in the scenario. The 3D GIS data model is supported through extending X3D and using FME Services, but the attributes cannot be visualized in shared 3D view because of the X3D driver's limitation. A new drive is needed to handle all X3D attributes. This prototype is also well-structured through supporting roles and negotiations.

9.4.2 Added Value of a 3rd Dimension for Group Decision-making in Geology

From the walkthrough experience, the added value of a third dimension is witnessed in several aspects, such as advanced 3D view visualization, enriched 3D data presentation and navigation. For example, in Figure 9.8, the four lenses are presented by true 3D visualization. These lenses are so complicated and overlapped that it is impossible to clearly present them with a 2D method. The advanced 3D navigation not only provides ordinary 2D navigation functions such as zoom-in, zoom-out and pan capabilities, but also has unique 3D navigation functions such as the rotation of 3D objects with different viewpoints. In the same example, users can take a closer look at a 3D model from different viewpoints, which helps them to better understand the model. With the help of 3D data modeling, the presentation and visualization of the sub-surface can be more realistic and accurate than with the 2D method. True 3D visualization is very important for geologists who use it to solve 3D problems by observing a relatively small portion of the rock units that are exposed and using judgement, experience, and imagination to extrapolate the areas that are hidden.

Moreover, the synchronous collaboration functions of the 3D view and the multiple communications significantly help to transfer the judgement and experience from a geologist to people who are not trained in this area during this decision-making process. For example, in Walkthrough 1, the last group includes group members who are from the economics and mechanics fields. When the team leader, who has geological knowledge, navigates the shared 3D view and explains the lenses' meaning, the other two members can easily understand the same 3D model. Thus, the overall decision-making process can benefit from collaborative 3D GIS.

Chapter 10 Conclusions and Recommendations

This thesis presents methods and key technologies, along with a prototype system, for adding synchronous collaboration capabilities into a 3D GIS environment. These methods and technologies not only addressed the social and organizational factors but also put these factors into system design and realization of a prototype system, GeoLink3D. The prototype system along with these factors is also evaluated by a walkthrough method. In this chapter, the realization of the research objectives is presented, followed by some conclusions and outcomes. Future work is presented at the end.

10.1 Realization of Research Objectives

The goal of this research was to add synchronous collaboration capabilities into a 3D GIS environment. More specifically, the objectives of this research were to investigate methods and technologies to design a synchronously collaborative 3D GIS environment for small group decision-making and collaboration. To realize these goals, this research was carried out in the following steps:

1) Requirements Collection and Conceptual Framework Design

One important objective of this research was to investigate the requirements analysis and conceptual framework, which relates to the basic research question: *what can 3DGIS do if it has collaborative capability?* The requirements were collected through a literature review, related groupware systems and a mock-up scenarios study. The conceptual framework delineated important social, GIS 3D and technological parameters and problems that must be considered to extend or reinvent geo-information technologies to support work by groups in a 3D environment. This framework not only provided the guidelines for the research contents but also limited the research scope and boundaries for SC3DGIS.

2) Social Collaboration Model

This part attempted to give a logical design about how the collaborated parties work together. The Social Collaboration Model (SCM) describes the behaviours, relations and collaboration activities among parties, such as users, systems and data sources. The related ontology presentation makes these behaviours and activities shareable and understandable in a distributed environment. The SCM was analyzed and designed through a case study in which a group meeting was simulated. The users' behaviours and interactions were also obtained in this simulation. The SCM was further presented as application-level ontology for shared understanding.

3) Collaborative 3D GIS Data Model

A 3D GIS data model for collaborative purposes requires explicit data state presentation, such as color and style, so that the distributed system can be kept in consistency. The criteria of SC3DGIS data model were presented. Compared with GML, KML and GRID, with respect to elements such as geometry, coordinate system, semantics, model state and styles, X3D was considered the most suitable for collaborative 3D GIS.

4) Prototype Design and Development: Shared View System and Multi-agent Support

This part tried to answer the basic research question: *What are the special problems in the design and development of SC3DGIS when considering it as a specific application of synchronous collaborative system?* Semi-replicated architecture was used in this shared 3D view prototype. A 3D rendering component and collaboration control component are embedded in every client. System states and events are transported in a real-time and multicast manner to every client. This architecture has better performance than centralized architecture, in which images are transported among clients. More intelligent collaboration such as negotiation and discussion can be achieved through introducing a multi-agent method. Agents are used to present users' behaviours and FIPA protocols are used to describe complex interactions. The multi-agent method also provides a solution for the shared understanding problem by introducing ontology.

5) Evaluation: Walkthrough Method

A walkthrough method was used to evaluate the prototype system, GeoLink3D. The evaluation results showed that the prototype system satisfied the goal of the research. The results also showed that adding synchronous and collaborative capabilities to a 3D GIS environment can significantly improve the efficiency and satisfaction of decision-making for geographically-distributed people.

10.2 Discussion of Thesis Outcomes and Conclusions

The work presented in this thesis represents a comprehensive research to the next generation 3D GIS, SC3DGIS, a real-time collaborative 3D GIS, from conceptual framework to architecture and prototype. The advantages of the SC3DGIS compared to traditional 3DGIS lie in the collaboration of multiple users and the integration of distributed system and data resources. Geographically dispersed people can work together and share not only data sources but also system operations and even knowledge from other users, which will improve the decision-making performance and save time and costs. From the walkthrough evaluation results, the conclusion can be drawn that the goal to add collaboration capability to 3D GIS is realized.

The research results indicate that the synchronous collaboration for 3D GIS has special features related to social, GIS and technological aspects. These aspects define the overall scope of the research as not only a technical issue but also a social one. The reason that the social collaboration model and the multi-agent method are involved in this research is to handle social issues. Because of the complexity of the collaboration, the research scope is narrowed to a small group synchronous collaboration in 3D GIS. The case study is also based on a virtual group meeting through simulating a shared 3D GIS environment. The social collaboration model, derived from the above situation, is also limited within a certain scenario. However, the method to create and use this model for a collaboration system is still valuable.

Nevertheless, it is well known that human-human collaboration is highly case- and context-specific. More detailed investigations into specific use cases in the 3D GIS field are needed as well as a refined understanding and representation of the context. For example, modeling the behaviours of users, sub-systems and data in a 3D GIS context is necessary to design SC3DGIS. Sharing the understanding of this behaviour model is related to the domain ontology, reasoning and rule-based technology. Furthermore, the well-defined description of behaviours among human-human, human to machine (system), and machine to machine will benefit broader research in spatial Web sensor network and public participation GIS (PPGIS).

The research results also indicate that the SC3DGIS is mainly classified as a different location and same time collaboration according to the time-space matrix (Johansen 998, Dix 1996). However, the workflow with different time patterns is also necessary for people to negotiate and make deals during the decision-making process. The unique feature of the workflow in SC3DGIS is the time-critical request-response process, which requires that every request need an immediately response. This feature also requires a semi-replicated architecture for SC3DGIS. In this architecture, a highly interactive 3D rendering component is located in every client. Instead of transferring view images in a general groupware system, the system consistency is maintained through event messages and a floor control mechanism. The kind of rich client is also used in Google Earth 3D, Skyline, MS Virtual Earth and other interactive 3D games. It can be concluded that real-time collaboration for 3D systems requires a rendering component for every client.

The developing tools for this prototype are based on pure Java and open sources such as Eclipse, Java3D, Xj3D, JGroup and JADE. The advantages of using open sources include that the tools are free of charge and that there is free access to the source codes. Related feedback to these tools also benefits an open source community. Problems with an open source mainly lie in a shortage of documentation and in unreliable quality of control and support.

As far as this prototype is concerned, there are also many limitations in terms of performance and functionalities because of the limited work force and time. For example, 3D rendering with Java3D is slower than C++ for rendering and requires much more time to load a large dataset, especially for the Internet data source. VRML loaders such as SUN VRML or Xj3D, are limited and may not be fully compatible with a Java3D system. The multi-agent methods using JADE also have performance problems. When many agents are working together, huge amounts of memory is used up and will cause reduced performance. There are as well inevitable limitations in the functions of this prototype for real work places. However, because the conceptual design is able to be independent from the implementation, GeoLink3D, as a proof-of-concept prototype, is mainly used to validate the design issues. The performance and functionality could be improved if implementation tools could be changed without changing the logical design.

The agent methods have advantages of solving complex communication and shared understanding problems for distributed clients. The agent method provides not only shared presentation such as ontology presentation but also tools such as agent communication language and FIPA standards at an implementation level. However, these methods are also not yet mature not only in agent system design methods but also in agent developing kits. As well, the performance problem is still not resolved in the Multi-Agent System area. The multi-agent method is another hot topic in semantic services and intelligent systems from boarder areas such as computer science and Artificial Intelligence. Related achievements in these areas will benefit and improve SC3DGIS.

The 3DGIS data model is a complex topic. Many researchers are working on this area. This research, rather than design and develop a brand new 3D GIS data model for collaborative purposes (which is essentially also not necessary because of the interoperability requirement), attempted to provide the criteria for the 3D data model by meeting: 1) the synchronous collaborative requirement, 2) the GIS requirement, and 3) the basic 3D geometry representation requirement. Although X3D was selected here as the data model, other 3D models such as KML, GML could be considered as import data sources. In fact, these models can be loaded through an FME service.

The added value of a third dimension for a synchronous collaborative GIS platform is obviously important especially for a geology-related group decision-making process. Since 3D visualization provides many benefits, including the ability to provide better models and views of a real-world situation, faster data exploration and understanding, and the ability to convey additional information that is not possible with a 2D presentation, geologists can use this visualization to solve geological problem through observation, judgment, experience and imagination, which are all critical methods for geologists to understand the sub-surface world. Because of the collaborative features of SC3DGIS, such as shared understanding and WYSIWIS interface, this knowledge or the solution to the geological problems can be easily transferred to other people who are not trained in geology but need to make a decision.

10.3 Future Work

Refine the requirements and the prototype

The next step of the research will refine the requirements and prototype to fit the real work place in mineral, petroleum and groundwater resources. When the prototype is used, more detailed collaborative behaviours and domain-oriented requirements in these work places can be obtained. The feedback from them will refine the conceptual framework and enrich the functionality of the prototype.

Investigate the multi-agent approach

It is worthwhile to further investigate the multi-agent method for the design and prototype of SC3DGIS. The multi-agent method was used to solve the complexity of multiple users' interactions and build a shared understanding in this research. Although multi-agent technology is still in the research stage, its capabilities in adaptability, autonomy, collaborative behaviour and mobility have huge potential to solve problems in collaborative computing, shared knowledge and rules, and complex communication such as negotiation. Since a multi-agent framework, like JADE, provides multicast data transportation, it is also possible to mitigate the whole collaborative component, which is using shared cache in this prototype, to a multi-agent framework.

Broaden the social collaboration model use case

In a SC3DGIS environment, multiple users have basic collaborative capabilities to work together. However, it is still a challenge for the users to simulate a real face-to-face work place to carry out one common goal. In a real work environment, the work structure or organization helps people work together through assigned roles, workflow and face-to-face communication. There exists a model like SCM to bridge the gap between the real work environment and collaborative virtual environment like SC3DGIS. The SCM could sit above the SC3DGIS framework or any virtual 3D systems such as Google Earth 3D, Second Life, and Microsoft Virtual Earth to help the multiple users take over roles, assign tasks and rebuild social structures. The participants can not only observe other's activities but also collaboratively work with other participants in a virtual organization. The SCM will specify the participant's behaviour, provide shared data source, define and implement work tasks and create common understanding for specific issues.

Sensor Web network application

The sensor Web network is a distributed sensing system in which information is globally shared by all networked platforms. However, a single sensor is a very simple functioning unit, for example, to measure temperature. The sensor Web network needs to describe and define the interactions among sensors so that a more complex work (goal) can be carried out by many sensors. There are still some challenges not only in collecting and presenting continuously changing diverse types of data (Nath et al. 2006) but also in the cooperative working together of the sensors.

The social collaboration model, like SCM in this research, will help organise many kinds of sensors to work together. Through this model, participants (users) not only find and collect, detect and monitor a sensor’s data and work stats, but also assign and coordinate sensor tasks. The Web sensor network, as a case study, can also be visualized in a shared 3D GIS environment, like SC3DGIS, so that researchers can dynamically monitor the sensors activities.

Intelligent GIS

The GIS community is believed to be more intelligent and automated, and is also called Intelligent GIS (IGIS). IGIS is not only used for powerful visualization and spatial queries, but also has intelligent capabilities to collaboratively work among users and machines. More specifically, the collaborative GIS, spatial semantic Web services, intelligent sensors together with all kinds of data source are connected and collaboratively working together through ontology and semantics, multi-agent, and GRID technology. The work flow, data flow and tasks will be smoothly carried out with the help of all kinds of virtual social collaboration and networking models and automatically run on the Internet. In a word, GIS technology together with its third and fourth dimensions is shifting towards ways that are mutually connected, distributed, ubiquitous, collaborated and therefore more intelligent.

References

Adams, M., Tenney, Y., and Pew, R. (1995): Situation Awareness and the Cognitive Management of Complex Systems. Human Factors, vol. 37, no. 1, pp. 85–104.

Ahuja, M.K., and Carley, K.M. (1998): Network Structure in Virtual Organizations, JCMC 3(4).

Altmaier, A., and Kolbe, T., H. (2004): Applications and Solutions for Interoperable 3d Geo-Visualization, the Photogrammetric Week 2003, pp. 1-5. September in Stuttgart, Germany.

Armstrong, M.P., and Densham, P.J. (1995): Cartographic support for collaborative spatial decision-making. Auto Carto 12, ACSM/ASPRS Technical Papers, 4, pp. 49-58.

Attitude Group. (2008): website explaining what is RSS, http://www.whatisrss.com/ , (accessed on January 16, 2009).

AutoDesk. (2009): AutoDesk 3ds Max website, http://usa.autodesk.com/adsk/servlet/index?siteID=123112&id=5659302, (Accessed on June 22, 2009).

Baeza-Yates, R., Pino, J. (2006): Towards formal evaluation of collaborative work and its application to information retrieval. Information Research 11(4), 271.

Baker, K., Greenberg, Gutwin, S. (2002): Empirical development of a heuristic evaluation methodology for shared workspace groupware. In: CSCW ’02, pp. 96–105.

Baldwin, R., G. (2007): Understanding Transforms in Java 3D. Java Program Notes # 1552, http://www.developer.com/java/other/article.php/3717101/Understanding-Transforms-in-Java-3D.htm, (last accessed on June 22, 2010)

Balram, S., and Dragicevic, S. (2006): Extending objects to model agents: a collaborative design framework using Agent UML extension. In Collaborative Geographic Information Systems, edited by Balram S., and Dragicevic, S. (Hershey: Idea Group Inc) pp. 121-133

Begole, J., Rosson, M., and Shaffer, C. (1999): Flexible collaboration transparency: supporting worker independence in replicated application-sharing systems, ACM Transactions on Computer Human Interaction, 6(2), 1999, pp. 95 – 132.

Bischofberger, W., and Pomberger, G. (1992): Prototyping-Oriented Software Development: Concepts and Tools. Texts and Monographs in Computer Science. Springer-Verlag.

Boyd, d. m., and Ellison, N. B. (2007): Social network sites: Definition, history, and scholarship. Journal of Computer-Mediated Communication, 13(1), article 11.

Breslin, J.G., and Decker, S. (2007): The Future of Social Networks on the Internet: The Need for Semantics, IEEE Internet Computing, vol. 11, pp. 86-90.

Brickley, D. (2008): The Friend of a Friend (FOAF) project website, http://www.foaf-project.org/, (Accessed on January 12, 2009).

Boroushaki, S. and J. Malczewski, 2009, ParcitipatoryGIS: A WebGIS-based Collaborative Multicriteria Decision Analysis. http://www.urisa.org/files/Boroushaki%20-%20ParcitipatoryGIS%20com%20(2).pdf (last accessed: 4/10/2010).

Bryson, J., Martin, D., McIlraith, S., and Stein, L.,A. (2003): Agent-Based Composite Services in DAML-S: The Behavior-Oriented Design of an Intelligent Semantic Web. Web Intelligence, N. Zhong, J. Liu, and Y. Yao, Springer Verlag.

Cai, G. (2005): Extending distributed GIS to support geocollaborative crisis management. Geographical Information Science. Special issue on distributed GIS 11(1): 4-14.

Cameron, B., DePalma, D.A., O'Herron, R. and Smith, N. (1995): Where does groupware fit? , The Forrester Report: Software Strategies, 6(3), June 1995.

Chabert, A., Grossman, E., Jackson, L., Pietrowicz, S., and Seguin, C. (1998): Java Object-Sharing in Habanero. Communications of the ACM, 41(6), pp 69-76.

Chang, E. and Li, S. (2007): Collaboration enabled GIS Tools for Emergency Operation Centre, Geomatics Solutions for Disaster Management, Lecture Notes in Geoinformation and Cartography, Springer-Verlag Berlin, Heidelberg, Pages. 149-163.

Chang, E. and Li, S. (2008): Architectural Design and Prototyping of a Web-based Synchronous Collaborative 3D GIS, Cartography and Geographic Information Science, Vol. 35, No. 2, pp. 117-132.

Chen, J., He, C., Jiang, J., and Han, G. (2008): Reconciliation of Inconsistent Perspectives in Collaborative GIS , Cartography and Geographic Information Science, Vol. 35, No. 2, pp. 77-89.

Chen, J., Zhao, R., Li, Z. (2004): Voronoi-based k-order neighbour relations for spatial analysis, PandRS(59), No. 1-2, August 2004, pp. 60-72.

Churcher, N., and Churcher, C. (1996): GroupARC – A Collaborative Approach to GIS. Proceedings of the 8th Colloquium of the spatial Information Research Center, University of Otago.

Churcher, N. and Churcher, C. (1999): RealTime Conferencing in GIS, Transactions in GIS, 1999, 3(1): 23-30.

CISCO. (2008): WebEx website, http://www.Webex.com/ (last accessed on January 16, 2008).

Coors, V. (2003): 3D GIS in Networking environments, Computer, Environment and Urbaqn Systems, Vol. 27 (4), pp. 345–357.

Dewan, P. (1999) :Architectures for Collaborative Applications, Computer Supported Cooperative Work. M. Beaudouin-Lafon. Chichester, UK, John Wiley & Sons: 169-194.

Dix, A. (1996): Challenges and Perspectives for Cooperative Work on the Web, Proceedings of the ERCIM workshop on CSCW and the Web, Sankt Augustin, Germany, February 7-9, 1996.

Dragicevic, S., and Balram, S. (2004): A Web GIS collaborative framework to structure and manage distributed planning processes. Journal of Geographical Systems 6(2): 133-153

Egenhofer, E., Franzosa, R. (1991): Point-Set Topological Spatial Relations, International Journal of Geographical Information Systems, 5(2), pp. 161-174.

Ellis, C., Gibbs, S., and Rein, G. (1991): Groupware: Some issues and experiences, Communications of ACM, 34(1), 38-58.

Endsley, M. (1995): Toward a Theory of Situation Awareness in Dynamic Systems. Human Factors, vol. 37, no. 1, pp. 32–64.

Friedman Memorial Airport. (2006): Wood River Region Airport Site Selection and Feasibility Study website, http://www.flyfma.com/index.asp?Type=B_BASIC&SEC=%7B2BDCB453-0891-4A8F-96D7-503AF71D1BE8%7D&DE=%7BA4F6DE87-077C-4944-9E26-DF20E56D3B2F%7D, (Accessed on January 16, 2009).

Fuh, J.Y.H., and Li, W.D. (2005): Advances in collaborative CAD: the-state-of-the art. Computer-Aided Design 37(5): pp. 571-581.

GeoVRML Working Group. (2008): GeoVRML website, http://www.ai.sri.com/geovrml/ , (accessed on November 20, 2008).

Gong, J., and Lin, H. (2006): A Collaborative VGE: Design and Development. In Collaborative Geographic Information Systems, edited by Balram S., and Dragicevic, S. (Hershey: Idea Group Inc) pp. 186-206.

Google. (2008): Google Docs Software website http://docs.google.com/ (Access December 2008).

Google. (2009): Google SketchUp Software website http://sketchup.google.com/ (Access December 2009).

Google. (2006): KML Documentation Introduction website, http://code.google.com/apis/kml/documentation/ , (accessed on January 16, 2006).

GROVE. (2008): Microsoft Office Grove website http://office.microsoft.com/en-ca/groove/HA101656331033.aspx (accessed December, 2006).

Gruber, T. R. (1993): A translation approach to portable ontologies. Knowledge Acquisition, 5(2):199-220.

GSA USA. (2007): U.S. General Services Administration Site Selection Guide, http://www.gsa.gov/Portal/gsa/ep/channelView.do?pageTypeId=8195&channelPage=%252Fep%252Fchannel%252FgsaOverview.jsp&channelId=-15016 (accessed on November 28, 2007).

Guarino, N. (1997): Semantic Matching: Formal Ontological Distinctions for Information Organization, Extraction, and Integration. SCIE 1997, pp. 139-170.

Gutwin, C. and Greenberg. S. (1996): Workspace Awareness for Groupware. Proceedings of the Conference on Human Factors in Computing Systems. Vancouver, pp. 208–209.

Gutwin, C. and Greenberg, S. (2000): The mechanics of collaboration: Developing low cost usability evaluation methods for shared workspaces. In: WETICE ’00, pp. 98–103. IEEE Comp. Soc, Los Alamitos.

Gutwin, C. and Greenberg, S. (2002): A Descriptive Framework of Workspace Awareness for Real-Time Groupware. Computer Supported Cooperative Work: The Journal of Collaborative Computing, 11(3-4):411-446, Special Issue on Awareness in CSCW, Kluwer Academic Press.

Hazzan, O., and Kramer, J. (2006): Abstraction in Computer Science and Software Engineering: A Pedagogical Perspective, System Design Frontier Journal, vol. 3, no. 12, 1-9.

Hill, R.D., Brinck, T., Rohall, S.L., Patterson, J.F., and Wilner, W. (1994): the rendezvous architecture and language for constructing multiuser applications. ACM Transactions on Computer-Human Interaction, 1 (June 1994), 2, p. 81-125.

Hinssen, P.J.H. (1998): What difference does it make?, The use of groupware in small groups. Elematica Instituut Fundamental Research Series, vol. 002. Telematica Instituut, Enschede, the Netherlands, in press, http://www.telin.nl/publicaties/1998/scout/scout.htm.

Hofte, G. H. (1998): Working Apart Together: Foundations for component groupware. Telematica Institute Fundamental Research Series.

Huang, B., Jiang, B., and Lin, H. (2001): An integration of GIS, virtual reality and the Internet for spatial data exploration. International Journal of Geographic Information Science, 15(5): 439-456.

IEEE FIPA. (2008): IEEE Foundation for Intelligent Physical Agents, Interaction Protocol Specifications website, ttp://www.fipa.org/repository/ips.php3. (accessed on April 27, 2008).

IEEE Standard, (1998): IEEE Std. 1028-1997, IEEE Standard for Software Reviews, pp. 1-37.

InfoPatterns. (2007): Toucan Navigate website, http://www.infopatterns.com/ (last accessed on July 10, 2007).

Internet Mail Consortium. (2008): VCard Overview website, http://www.imc.org/pdi/vcardoverview.html (last accessed on January 16, 2009).

Jankowski, P., and Nyerges, T. (2001): Geographic Information Systems for Group Decision Making: Towards a participatory, geographic information science (New York: Taylor & Francis).

Jankowski, P. and Nyerges, T. (2003): Towards a Framework for Research on Geographic Information-Supported Participatory Decision Making, Urban and Regional Information Systems Journal, vol. 15, pp. 9-17.

Jasper, R. and Uschold, M. (1999): A framework for understanding and classifying ontology applications. In Proceedings of the IJCAI99 Workshop on Ontologies and Problem-Solving Methods (KRR5).

Java-Source.net. (2008): Open source Catch solutions website http://java-source.net/open-source/cache-solutions, ((last accessed on July 10, 2008).

Jeffery, C. L. (2008): Collavorative Virtual Environment Category website, http://www.cs.nmsu.edu/~jeffery/courses/579-cve/cves.html (accessed December, 2008).

JGroup. (2007): JGroup website, http://www.jgroups.org/javagroupsnew/docs/index.html, (accessed on Febrary 12, 2007).

Johansen, R. (1998): Current user approaches to groupware. In R. Johansen (ed.), Groupware : Computersupport for business teams. Free Press, New York, 1988, p. 12-44.

Jones, R.M., Copas, C., and Edmonds, E.A. (1997): GIS support for distributed group-work in regional planning. International Journal of Geographical Information Systems. 11(1): 53 – 71.

Khronos Group. (2006): COLLADA website, http://www.collada.org/mediawiki/index.php/Main_Page, (Accessed on May 1, 2006).

Kolbe, H. T. (2006): CityGML website, http://www.citygml.org/, (Accessed on April 2, 2006).

Larrivee, S., Bedard, Y., and Pouliot, J. (2005): How to enrich the semantics of geospatial Databases by properly expressing 3D objects in a conceptual model. OTM Workshops 2005: 999-1008.

Leica. (2007): Leica Virtual Explorer, http://gi.leica-geosystems.com/ (last accessed on July 10, 2007).

Li, S., and Coleman, D. (2003): A Web-based collaboration system for managing distributed GIS data production. Geomatica 57 (1): 351–361.

Li, S., Guo, X., Ma, X. and Chang, Z. (2007): Towards GIS-enabled Virtual Public Meeting Space for Public Participation, Photogrammetric Engineering & Remote Sensing 73(6): 641-650.

MacEachren, A.M. (2001): Cartography and GIS: extending collaborative tools to support virtual teams. Progress in Human Geography 25(3): 431-444.

MacEachren, A.M., and Brewer, I. (2004): Developing a conceptual framework for visually-enabled geocollaboration. International Journal of Geographical Information Science 18(1): 1-34.

MacEachren, A. M., Brewer, I., and Steiner, E. (2001): GEOVISUALIZATION TO MEDIATE COLLABORATIVE WORK: Tools to Support Different-Place Knowledge Construction and Decision-making, Conference Proceedings of the 20th International Cartographic Conference, Beijing, China, August 6-10.

MacEachren, A. M., Brewer, I., Cai, G., and Chen, J. (2003): Visually Enabled Geocollaboration to Support Data Exploration and Decision-Making. Proceedings of the 21st International Cartographic Conference, Durban, South Africa, 10-16 August 2003.

MIRARCO, (2008): MIRARCO Mining Innovation’s website, http://www.mirarco.org/ (accessed on July, 2008)

Microsoft, (2000): NetMeeting 2.1 OverView http://www.microsoft.com/technet/prodtechnol/netmting/reskit/netmtg2/chpt1.mspx (accessed on July, 2006).

Mika, P. (2005): Ontologies Are Us: A Unified Model of Social Networks and Semantics. In Proceedings of the 4th International Semantic Web Conference, Springer-Verlag, LNCS 3729, pp. 522-536.

Molenaar, M. (1990): A formal data structure for 3D vector maps. In: Proceedings of EGIS'90, Vol. 2. Amsterdam, The Netherlands, pp. 770–781.

Myers, B. A., Shan, Y., Chuang, A., Tj, M., Chen, M., and Lee, C. (2004): Floor Control in a Highly Collaborative Co-located Task. http://www.cs.cmu.edu/~pebbles/papers/pebblesfloorcontrol.pdf (last accessed on July 2004).

Nath, S., Liu, J., Miller, J., Zhao, F., Santanche, A. (2006): SensorMap: a Web site for sensors world-wide. SenSys 2006: 373-374.

NC4. (2006): ETeam website, http://www.nc4.us (accessed December, 2006).

Nikrazla, M., Caireb, G., and Bahria, P. (2006): A methodology for the development of multi-agent systems using the JADE platform. International Journal of Computer Systems Science & Engineering 21(2).

Norman, D. (1993): Things That Make Us Smart. Reading, MA: Addison-Wesley.

Noy, N. F. and McGuinness, D. L. (2001): Ontology Development 101: A Guide to Creating Your First Ontology. Knowledge Systems Laboratory, March, 2001. (http://www.ksl.stanford.edu/KSL_Abstracts/KSL-01-05.html)

Nyerges, T. (1999): Progress in spatial decision making using geographic information systems. Geographic Information Research: Trans-Atlantic Perspectives, edited by M. Craglia and H. Onsrud (London: Taylor & Francis), pp. 129-142.

Odell, J. (2000): Objects and agents: how do they differ?. Journal of Object-Oriented Programming, October 2000.

OGC. (2003): Open Geospatial Consortium Web Terrain Service, http://portal.opengeospatial.org/files/index.php?artifact_id=11499, (accessed on July, 2006).

OGC. (2006): the Geography Markup Language website, http://www.opengis.net/gml/, (last accessed on July 25, 2006).

Oren, N. (2008): Managing 3D Geological Data Using Oracle 3D: a Study on Data Modeling and Management. Master of Science These, Department of Civil Engineering, Ryerson University.

Ousterhout, J. (1994): Tcl and the Tk Toolkit. Addison Wesley.

Pilouk, M. (1996): Integrated modelling for 3D GIS, Ph.D. Dissertation, ITC, Netherlands. Conference on Geo-spatial theory, Processing and Applications.

Pinelle, D. and Gutwin, C. (2002): Groupware walkthrough: adding context to groupware usability evaluation. ACM Press, New York, In: CHI ’02, pp. 455–462.

Rhyne, T. (1999): A commentary on GeoVRML: a tool for 3D representation of georeferenced data on the Web. International Journal of Geographic Information Systems, (13)4: 439-443.

Rikkers, R., Molenaar, M., Stuiver, J. (1993): A query oriented implementation of a 3D topologic data structure. In: Proceedings of EGIS’93, Vol.2. Genoa, Italy, pp. 1411–1420.

Rinner, C., 2001. Argumentation maps: GIS-based discussion support for online planning. Environment and Planning B: Planning and Design 28(6), 847-863.

Rodham, K. J., and Olsen D. R. (1994): Smart telepointers: maintaining telepointer consistency in the presence of user interface customization. ACM Transactions on Graphics, 13(3), July 1994, pp. 300 – 307.

Roseman, M. and Greenberg, S. (1992): Groupkit: A Groupware Toolkit for Building Real-Time Conferencing Applications. In Proceedings of the ACM 1992 Conference on Computer Supported Cooperative Work (CSCW '92), 43-50, Toronto, Canada, November 1992.

Schafer, W.A., Doug A. B. (2003): A Comparison of Traditional and Fisheye Radar View Techniques for Spatial Collaboration. Graphics Interface 2003: 39-46.

Shi, W.Z., Yang, B.S., Li, Q.Q. (2003): An object-oriented data model for complex objects in three-dimensional geographic information systems. International Journal of Geographic Information Science, Vol. 17 (5), pp. 411–430.

Skyline. (2006): Skyline Software System, Inc. website, http://www.skylinesoft.com/corporate/corporate_home.asp (last accessed on July 10, 2006).

Skype. (2006): Skype website. http://www.skype.com/ (accessed on July 10, 2006)

Stanford. (2009): Stanford Center for Biomedical Informatics Research, website, http://protege.stanford.edu/ (accessed on October 10, 2009).

Steves, M., Morse, E., Gutwin, C., Greenberg, S. (2001): A Comparison of Usage Evaluation and Inspection Methods for Assessing Groupware Usability. In Proceedings of the ACM Conference on Supporting Group Work, 125-134.

Stock, C. and Bishop, I.D., (2006): Linking GIS with Real-Time Visualisation for Exploration of Landscape Changes in Rural Community Workshops, Virtual Reality 9:260-270.

Sun. (2004): JSDT website. http://java.sun.com/products/java-media/jsdt/. (last accessed on July 2004).

Suthers, D. (2001): Architectures for Computer Supported Collaborative Learning. IEEE International Conference on Advanced Learning Technologies (ICALT 2001).

Tang, T., and Zhao., J., and Coleman, D.J. (2005): Design of a GIS-enabled Online Discussion Forum for Participatory Planning. Proceedings of the 4th Annual Public Participation GIS Conference, August 2005.

Telecom Italia. (2008): Java Agent DEvelopment Framework website, http://jade.tilab.com/, (assessed in May, 2008).

Tencent. (2008): TencentQQ's website http://www.qq.com/ (accessed on July 10, 2008).

Theoktisto, V., Fairen, M. (2005): Enhancing collaboration in virtual reality application. Computers & Graphics, 29, (2005) 708-722.

Tjortjis, C., Dafoulas, G., Layzell, P., and Macaulay, L. (2002): A Model for Selecting CSCW Technologies for Distributed Software Maintenance Teams in Virtual Organisations, Proc. of 26th Annual International Computer Software and Applications Conference.

Unype. (2007): Unype website, http://www.unype.com/ (last accessed on July 25, 2007)

Web3D. (2008): X3D specification, Web3D Consortium, inc. ISO/IEC 19775-1:2008. http://www.Web3d.org/x3d/specifications/ (last accessed on June 02, 2010)

Web3D-BOX. (2008): X3D box node definition. Web3D Consortium, inc. ISO/IEC 19775-1 http://www.Web3d.org/x3d/specifications/ISO-IEC-19775-1.2-X3D-AbstractSpecification/index.html

Wong, S. and Chua, Y.L. (2001): Data Intermediation and Beyond: Issues for Web-based PPGIS. Cartographica, 38(3&4).

Xia, S., Sun, D., Sun, C., Chen, D., and Shen, H.F. (2004): Leveraging single-user applications for multi-user collaboration: the CoWord approach, Proceedings of ACM 2004 Conference on Computer Supported Cooperative Work, Nov 6-10, Chicago, IL USA, pp.162-171.

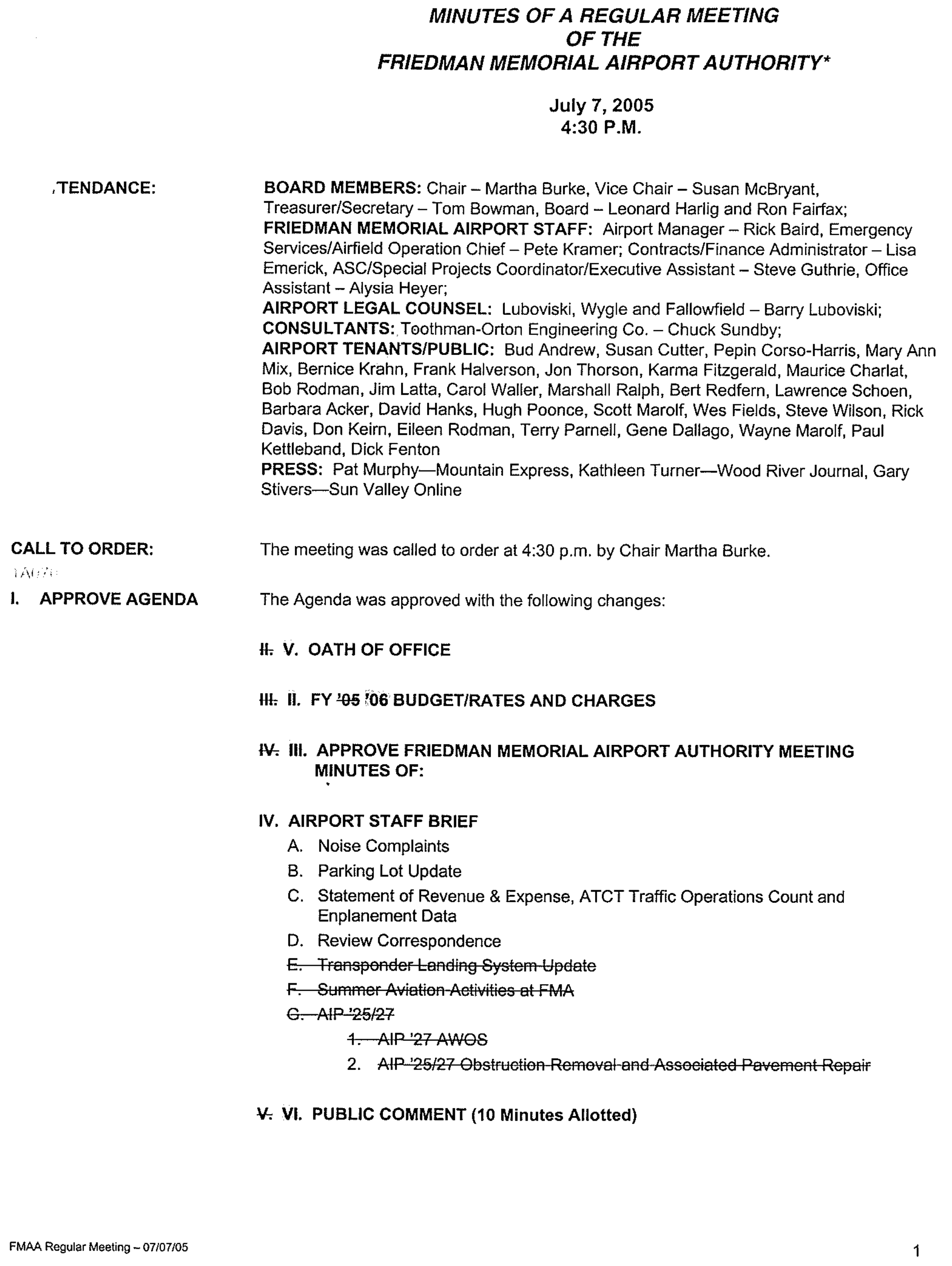
Xith3D. (2006): Xith3D website, http://xith.org/, (Accessed on May 12, 2006).

Zhdanova, A.V., Predoiu, L., Pellegrini, T., Fensel, D. (2007): A Social Networking Model of a Web Community. In Proceedings of the 10th International Symposium on Social Communication, 22-26 January 2007, Santiago de Cuba, Cuba, ISBN: 959-7174-08-1, pp. 537-541.

Zlatanova, S. (2000): 3D GIS for urban development, Ph.D. dissertation, ITC, The Netherlands.

Zlatanovaa, S., Abdul, R., and Piloukc, M. (2002): 3D GIS: current status and perspectives, Proceedings of the Joint Conference on Geo-spatial theory, Processing and Applications, Ottawa, Canada.

APPENDIX 1 Minutes of Regular Meeting of the Friedman Memorial Airport Authority



MINUTES OF A REGULAR MEETING
OF THE
FRIEDMAN MEMORIAL AIRPORT AUTHORITY*

July 7, 2005
4:30 P.M.

,TENDANCE:

BOARD MEMBERS: Chair – Martha Burke, Vice Chair – Susan McBryant, Treasurer/Secretary – Tom Bowman, Board – Leonard Harlig and Ron Fairfax;
FRIEDMAN MEMORIAL AIRPORT STAFF: Airport Manager – Rick Baird, Emergency Services/Airfield Operation Chief – Pete Kramer; Contracts/Finance Administrator – Lisa Emerick, ASC/Special Projects Coordinator/Executive Assistant – Steve Guthrie, Office Assistant – Alysia Heyer;
AIRPORT LEGAL COUNSEL: Luboviski, Wygle and Fallowfield – Barry Luboviski;
CONSULTANTS: Toothman-Orton Engineering Co. – Chuck Sundby;
AIRPORT TENANTS/PUBLIC: Bud Andrew, Susan Cutter, Pepin Corso-Harris, Mary Ann Mix, Bernice Krahn, Frank Halverson, Jon Thorson, Karma Fitzgerald, Maurice Charlat, Bob Rodman, Jim Latta, Carol Waller, Marshall Ralph, Bert Redfern, Lawrence Schoen, Barbara Acker, David Hanks, Hugh Poonce, Scott Marolf, Wes Fields, Steve Wilson, Rick Davis, Don Keirn, Eileen Rodman, Terry Parnell, Gene Dallago, Wayne Marolf, Paul Kettleband, Dick Fenton
PRESS: Pat Murphy—Mountain Express, Kathleen Turner—Wood River Journal, Gary Stivers—Sun Valley Online

CALL TO ORDER: The meeting was called to order at 4:30 p.m. by Chair Martha Burke.

I. APPROVE AGENDA The Agenda was approved with the following changes:

~~II.~~ V. OATH OF OFFICE

~~III.~~ II. FY ~~'05~~ '06 BUDGET/RATES AND CHARGES

~~IV.~~ III. APPROVE FRIEDMAN MEMORIAL AIRPORT AUTHORITY MEETING MINUTES OF:

IV. AIRPORT STAFF BRIEF

A. Noise Complaints
B. Parking Lot Update
C. Statement of Revenue & Expense, ATCT Traffic Operations Count and Enplanement Data
D. Review Correspondence
~~E. Transponder Landing System Update~~
~~F. Summer Aviation Activities at FMA~~
~~G. AIP '25/27~~
    ~~1. AIP '27 AWOS~~
    2. ~~AIP '25/27 Obstruction Removal and Associated Pavement Repair~~

~~V.~~ VI. PUBLIC COMMENT (10 Minutes Allotted)

VII. AIRPORT STAFF BRIEF

E. Transponder Landing System Update
F. Summer Aviation Activities at FMA
G. AIP '25/27
   1. AIP '27 AWOS
   2. AIP '25/27 Obstruction Removal and Associated Pavement Repair

~~VII.~~ VIII. OLD BUSINESS

~~VIII.~~ IX. NEW BUSINESS

~~IX.~~ X. EXECUTIVE SESSION – I. C. §67-2345(1)(f)

~~X.~~ XI. ADJOURNMENT

II. FY '06 BUDGET/RATES AND CHARGES

(See Brief and PowerPoint Presentation)

The Airport Manager and Board discussed concerns that the Airport is significantly undercharging for some services and has an opportunity to revise Rates and Charges more accurately. A rate increase was proposed for FY '06 and during the next 12 months Staff would be categorizing aircraft by sizes and rating charges accordingly. If Staff completes work on this subject early, then the Rates and Charges may be on the Board's Agenda sooner than Budget time next year.

1A[illegible]

MOTION: ***Made by Board Member Harlig to adopt the FY '06 Rates and Charges/Budget as proposed and publish them for the August 2, 2005 FMAA Board Meeting. Seconded by Board Member McBryant.***

PASSED UNANIMOUSLY

III. APPROVE MEETING MINUTES OF:

The Junè 7, 2005 Friedman Memorial Airport Authority Regular Meeting with the following changes:

Page 4, IV. OLD BUSINESS, A. AIP '29 Wood River Region New Airport Site Selection & Feasibility Studies (See Brief and PowerPoint Presentation)
Consultant Sundby stated the goal of the Consultants is to have graphics developed for comparison of ~~the preferred alternative site to~~ C-III compliant alternatives at the existing Airport.

1A270

MOTION: ***Made by Board Member Harlig to approve the June 7, 2005 Regular Meeting Minutes as amended. Seconded by Board Member Fairfax. Board Member Bowman abstained.***

MOTION PASSED

IV. AIRPORT STAFF BRIEF

A. Noise Complaints (See Brief)

B. Parking Lot Update (See Brief)

C. **Statement of Revenue & Expense and ATCT Traffic Operations Count and Enplanement Data (See Brief)**

D. **Review Correspondence (See Brief and PowerPoint Presentation)**

Board Member Fairfax and Airport Manager discussed VFR (Visual Flight Restriction) activities during the first week of July 2005.

(Airport Staff Brief items E., F., and G. continued below.)

V. OATH OF OFFICE

City/County Board Member Appointee

Board Member Bowman was given the Oath of Office by Hailey Mayor McBryant.

Board Discussion:

Board Member Harlig suggested Public Comment be conducted for ten minutes for subjects that are not on the Agenda, and allow more Public Comment within the AIP '29 Wood River Region New Airport Site Selection & Feasibility Studies section.

Board Member McBryant noted the Hailey City Council was in full attendance for the noticed meeting.

VI. PUBLIC COMMENT

Wes Fields of Camas County (80-year resident) stated he is a retired rancher and would like to see the airport location approved in Camas County.

VII. AIRPORT STAFF BRIEF

E. **Transponder Landing System Update (See Brief and PowerPoint Presentation)**

F. **Summer Aviation Activities at FMA (See Brief and PowerPoint Presentation)**

Chair Burke, Board Member Harlig and Airport Manager discussed September 2005 activities and how operations and security would be handled.

The Board, Airport Manager and Operations Manager addressed noise-complaint expectancies.

Dick Fenton inquired about the Naples Airport decision.

The Airport Manager indicated, in the case City of Naples Airport Authority v. Federal Aviation Administration, the City of Naples Airport Authority believed they had won the court decision. The Airport Manager also noted the decision does not automatically allow other communities to implement a Stage 2 aircraft ban, however, that route could be pursued. He added that FAR 161 Noise Studies were costly and did not guarantee FAA acceptance.

G. **AIP '25/27 (See Brief and PowerPoint Presentation)**

1. **AIP '27 AWOS**

2. **AIP '25/27 Obstruction Removal and Associated Pavement Repair**

VIII. OLD BUSINESS

A. AIP '29 Wood River Region New Airport Site Selection & Feasibility Studies (See Brief and PowerPoint Presentation)

Board Member Fairfax and Chair Burke addressed the June 30, 2004 [sic] letter that the Board had received from the Sun Valley/Ketchum Chamber of Commerce. Chair Burke read sections of the letter forwarded to the SSAC dated June 13, 2005 to clarify Board action that had occurred during the June 7, 2005 FMAA Board Meeting.

The Board further discussed the decisions regarding the SSAC at the June 7, 2005 FMAA Regular Meeting.

Public Comment with regard to AIP '29 Wood River Region New Airport Site Selection & Feasibility Studies noted:

David Hanks, a resident of Camas County, Mayor of Fairfield, and business owner in Camas County, stated positive discussions regarding Site 13 have occurred at City of Fairfield Council Meetings and that it has been looked upon as one of the most viable economic generators presented to Fairfield (other than the commuter population boom). Mr. Hanks further reported that the Fairfield Chamber of Commerce members have endorsed the continued pursuit of locating an airport at Site 13; that there have been complaints by Camas County residents for the lack of community/educational meetings to address Site 13; and about the localized group with the "big shot lawyers" to sway the FMAA Board and SSAC's vote away from Site 13. He then requested the Board delay the Special Meeting to the end of September in order for the Camas County community to work towards a greater sense of consensus regarding Site 13.

Jim Latta, from the Sun Valley/Ketchum Chamber, stated he was not on the SSAC and had not seen the June 13, 2005 letter, however, his perception was the Board had no more use for the SSAC. He also claimed the Sun Valley/Ketchum Chamber believed the SSAC helps maintain the integrity of the Site Selection Process.

Rob Cronin, a Hailey business owner and Fairfield property owner, stated the airfield in Hailey is dangerous and requested consideration of Site 13.

Hugh Koonce, a lifetime Fairfield resident and adjacent landowner to Site 13, believes an airport located there would be the best thing that could happen to Camas County.

Frank Halverson, an SSAC member, stated his perception was that the SSAC was "fired." He requested the SSAC be involved to look at the remaining items for discussion on the SSAC agenda dated October 2004.

Dick Fenton reported at the (1) March 22, 2005 SSAC Meeting next steps for the April 25, 2005 meeting were to look at funding and financial issues, however, the April meeting was postponed; (2) the May 24, 2005 SSAC next steps indicated for the June meeting to discuss financial feasibility analysis, but the June meeting was cancelled; and (3) the study steps for the SSAC included a comparative analysis for the existing Airport to the preferred alternative site and a subsequent discussion with the SSAC for implementation strategies schedule and finance. He also suggested the SSAC be included in discussions about what would become of the current Airport site and concluded with a wish that the process would remain open.

Bernice Krahn, of Camas County, stated she does not want Site 13 chosen for an airport.

Maurice Charlat stated that Steven Hart of SkyWest had sent out an email revealing disappointment that the SSAC was disbanded.

Bert Redfern, a Hailey resident, inquired if the overlay graphics for the existing Airport site to meet C-III Compliance were still in the works.

Consultant Sundby stated that the Consultants intended to make those graphics available at the Special Meeting.

Stephanie Marvel, a Hailey resident, announced her opposition to expanding the existing Airport site and thought the issue had been settled years ago.

Steve Wilson stated he is the owner of the largest private business in Camas County and he supports an airport at Site 13.

Paul Kettleband, Sr., lives within a mile of Site 13, and he wanted to be on record for his support of an airport located there.

Bud Bolan, a property owner in Hailey and Camas County, agreed with the stance of not expanding the existing Airport site. He has faith in the Board, and the SSAC, and that the process would be appropriately pursued. He also requested if the SSAC was to continue, then there should be equal representation from Lincoln County, Camas County and Blaine County (Sun Valley/Ketchum, Hailey, and Bellevue).

Marshall Ralph, worked closely with the Preserve the Camas Prairie's submittals to the Board, and because of the involvement with the "big shot lawyers" he had acquired significant knowledge from them. Mr. Ralph submitted a handwritten comment (Attachment #1) to the Board.

Wayne Marolf, of Camas County, announced his support of an airport at Site 13 and believes it would be the best thing to happen to Camas County.

Larry Schoen, who resides adjacent to Highway 20, requested an outline or direction on how the Site Selection Process would proceed in addition to an Agenda at the Special Meeting.

Carol Waller, Sun Valley/Ketchum Chamber, agreed with Mr. Schoen and inquired about the scope of work left.

Scott Marolf, a resident of Fairfield and on the Fairfield City Council, agreed with Mayor Hank's statements. He also asked that a decision on the airport location be postponed to allow time to get feedback from Camas County residents.

Board Member Bowman asked if there was a plan for how public input would be collected for Camas County residents.

David Hanks stated the Fairfield City Council would approach the County Commissioners to conduct public hearings to present information that has been gathered through the SSAC process in order to gauge Camas County resident interest.

Board Member Harlig asked if there could be more than one site proposed to the FAA for the EIS/NEPA process.

Consultant Sundby stated he believed there was a high degree of flexibility and that more than one site can be considered in an EIS process, this is a principle part of the EIS scoping process, however, it would make the process more complicated.

Barbara Acker asked if the FAA could dismiss all researched sites.

Consultant Sundby stated it was possible that all the sites could fail as a result of information developed in the EIS/NEPA process.

Pepin Corso-Harris, a resident south of Bellevue, stated that safety, reliability, functionality, having the least impact on quality of life, and room for expansion are needed at an airport site, and the existing Airport does not meet these standards.

Chair Burke addressed the concern of communities not having the time to present the Airport Site Selection Process results and possibilities.

Board Member McBryant suggested Lincoln County and Camas County conduct public meetings to report community consensus to the Board. She also suggested that the Special Meeting date be rescheduled for mid-September (instead of August) and for Staff to draw up a format to identify the next steps at the end of each meeting, and meetings to remain facilitated to allow public commentary.

The Airport Manager, Consultant Sundby and the Board discussed rescheduling the Special Meeting. After taking into account the time-sensitive safety-related projects that fully occupy the current time of Staff and Consultants, the 9/11 Commemorative Event scheduled for the second week in September, and schedule availability for the FMAA, Consultants, and Staff the earliest available date was September 28.

The Special Meeting was rescheduled for Wednesday, September 28, 2005 at 6:00 p.m. and the location will be determined based on venue availability.

Board Member McBryant reiterated the suggestion to have a "posted agenda" and asked for other suggestions to be submitted to the Staff before the August 2, 2005 FMAA Board Meeting.

Board Member Bowman and Chair Burke discussed a "road map" be created to help convey the next steps of the process to the public, and to allow the "road map" to have the flexibility to change month to month.

Legal Counsel Luboviski agreed with the decision to allow time for Lincoln and Camas Counties' Commissioners to poll their residents.

B. AIP '31 Improve Terminal Building (Phase 2) (See Brief and PowerPoint)

C. AIP '30 Runway 13-31 Safety Area Improvements (See Brief and PowerPoint)

Board Member McBryant requested Staff publicize the official 17 consecutive dates of Airport closure (that are to occur next summer) due to community activities already in the works.

Board Member Bowman asked what the benefits of the new storm drain would be.

Consultant Sundby stated the current style of drainage, grassy swales and dry wells, needed to be graded due to the safety area widening; by grading the drainages the water storage capacity is lost and the water would spill over onto adjacent taxiways.

Consultant Sundby stated the Airport would need to acquire a Runway Protection Zone (RPZ), which requires an easement of 21.75 acres when the runway extension is completed.

Board Member Harlig and Consultant Sundby discussed the timeline of acquiring the easement. Board Member Harlig asked why the easement was not negotiated first. Consultant Sundby responded the FAA expects the extension and acquisition of the easement to occur concurrently and that the RPZ easement should be acquired by August 1, 2006.

D. Tower Hours of Operation

The Airport Manager stated that information was insufficient to address the subject at this time and would be placed on the August Agenda.

IX. NEW BUSINESS

A. Landing System Possibilities (See Brief and PowerPoint)

The Airport Manager introduced Terry Parnell, FAA Manager Northwest Mountain Region of the Seattle Flight Procedures Office, who was to present to help explain the different landing system possibilities at Friedman Memorial Airport (FMA). (See Acronym Guide Attachment # 2)

The Airport Manager and Mr. Parnell stated the FAA was currently developing a landing procedure for the Airport (expected to be in use Fall 2005).

Board Member Harlig, Mr. Parnell and Consultant Sundby discussed the timeline for an LDA (Localizer Type Directional Aid) approach system for next year.

Dick Fenton inquired if vertical guidance was also characteristic of the WAAS (Wide Area Augmentation System), as the TLS (Transponder Landing System), and what was the vertical guidance benefit.

Mr. Parnell stated vertical guidance presents the pilot a glide slope to follow and lower minimums. He also noted that trees near the Airport would interfere with the approach clear airspace requirements.

Discussion continued between Public, Staff and Board regarding the different approaches and meeting clear zone and virtual area requirements at FMA.

Mr. Parnell stated he would like to be notified as soon as possible of any decisions regarding landing systems in order for the FAA office to review planning.

Legal Counsel Luboviski mentioned land use complications could not be determined until the system is modeled and that the FAA would not model a landing system until the Board made a decision.

Board Member McBryant asked if the FAA was equipped to help locate funding for the new landing system and acquiring necessary land to meet requirements.

Mr. Parnell stated he believed the Administrator implied reasonable monies would be available.

The Board and Mr. Parnell discussed the technical difficulties of the LDA system and the TLS.

Chair Burke inquired if the TLS system were implemented, then what would happen if the approval applicant using the TLS system were to no longer operate at FMA.

Mr. Parnell stated the technology would still be there, however, a new applicant would have to come forward. He also stated that the FAA was neutral to all landing systems.

Dick Fenton requested the Board listen to ANPC's (Advanced Navigation and Positioning Corporation) concerns on what type of landing system was implemented at the Airport.

Board Member Harlig asked that the PowerPoint slides on Landing System Possibilities be supplied to the Board before a decision was to be made.

X. EXECUTIVE SESSION – I.C. §67-2345(1)(f)

MOTION: ***Board Member Harlig moved to enter into Executive Session - §67-2345(1)(f). Seconded by Board Member McBryant.***

ROLL CALL VOTE:

Board Member McBryant	***Yes***
Board Member Bowman	***Yes***
Board Member Harlig	***Yes***
Board Member Fairfax	***Yes***
Board Chair Burke	***Yes***

PASSED UNANIMOUSLY

MOTION: ***Board Member McBryant moved to end Executive Session - §67-2345(1)(f). Seconded by Board Member Harlig.***

PASSED UNANIMOUSLY

XI. ADJOURNMENT

MOTION: ***Board Member Harlig moved to adjourn the Regular Board Meeting. Seconded by Board Member Bowman.***

PASSED UNANIMOUSLY

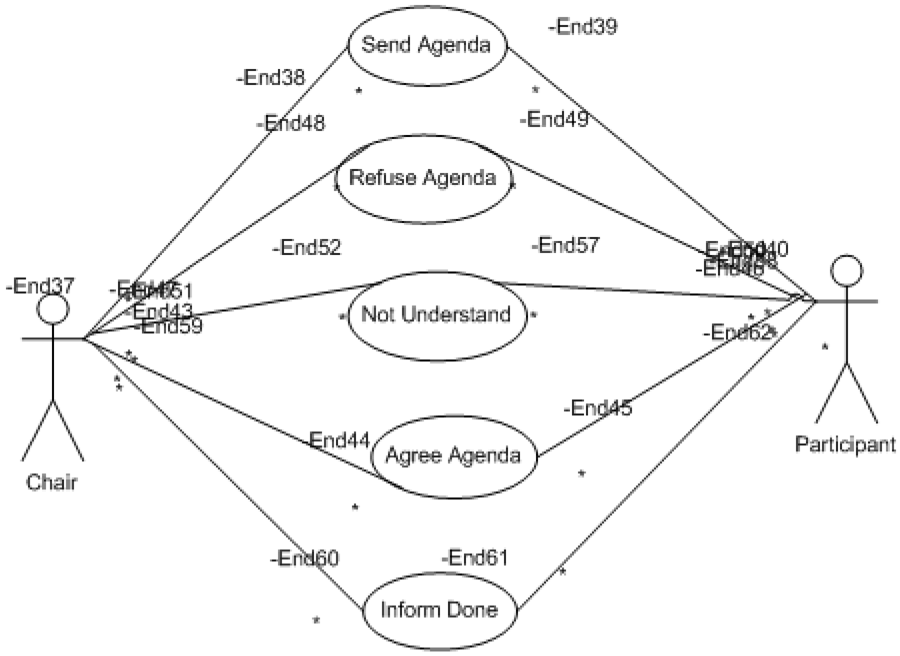
The July 7, 2005 Regular Meeting of the Friedman Memorial Airport Authority was adjourned at approximately 9:45 p.m.

Tom Bowman, Secretary

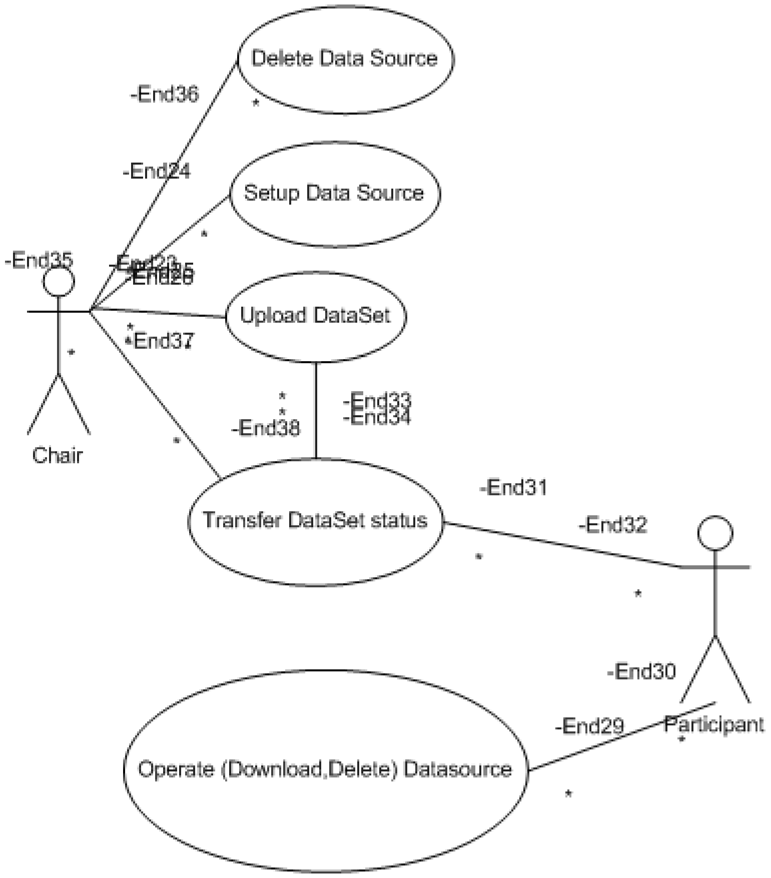
* *Additional resources/materials that should be reviewed with these meeting minutes include but are not limited to the Friedman Memorial Airport Authority Board Packet briefing, the PowerPoint presentation prepared for this meeting and any referenced attachments.*

APPENDIX 2 Use Cases for Virtual Group Meeting

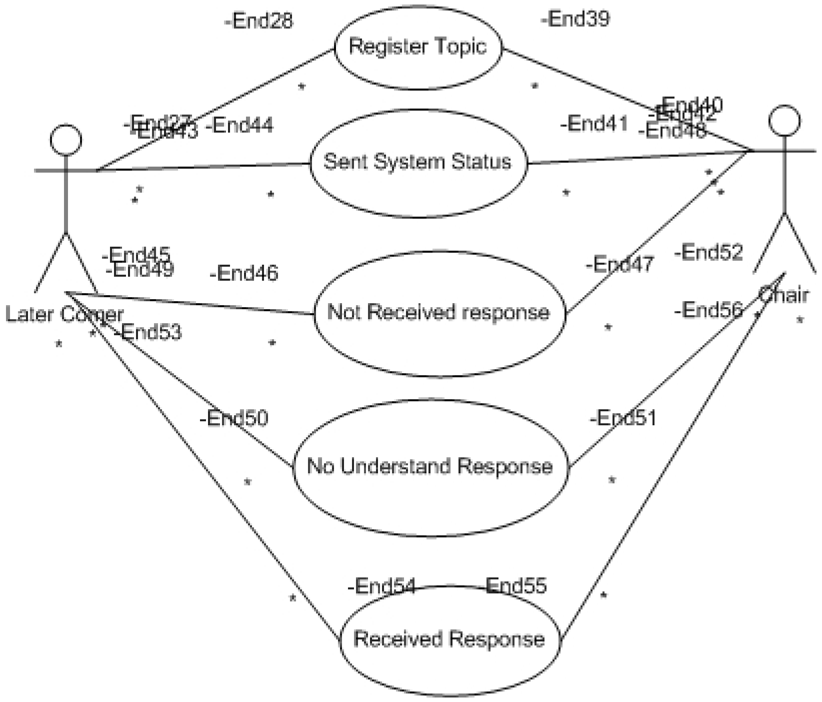
Approving issues (Agenda) use case



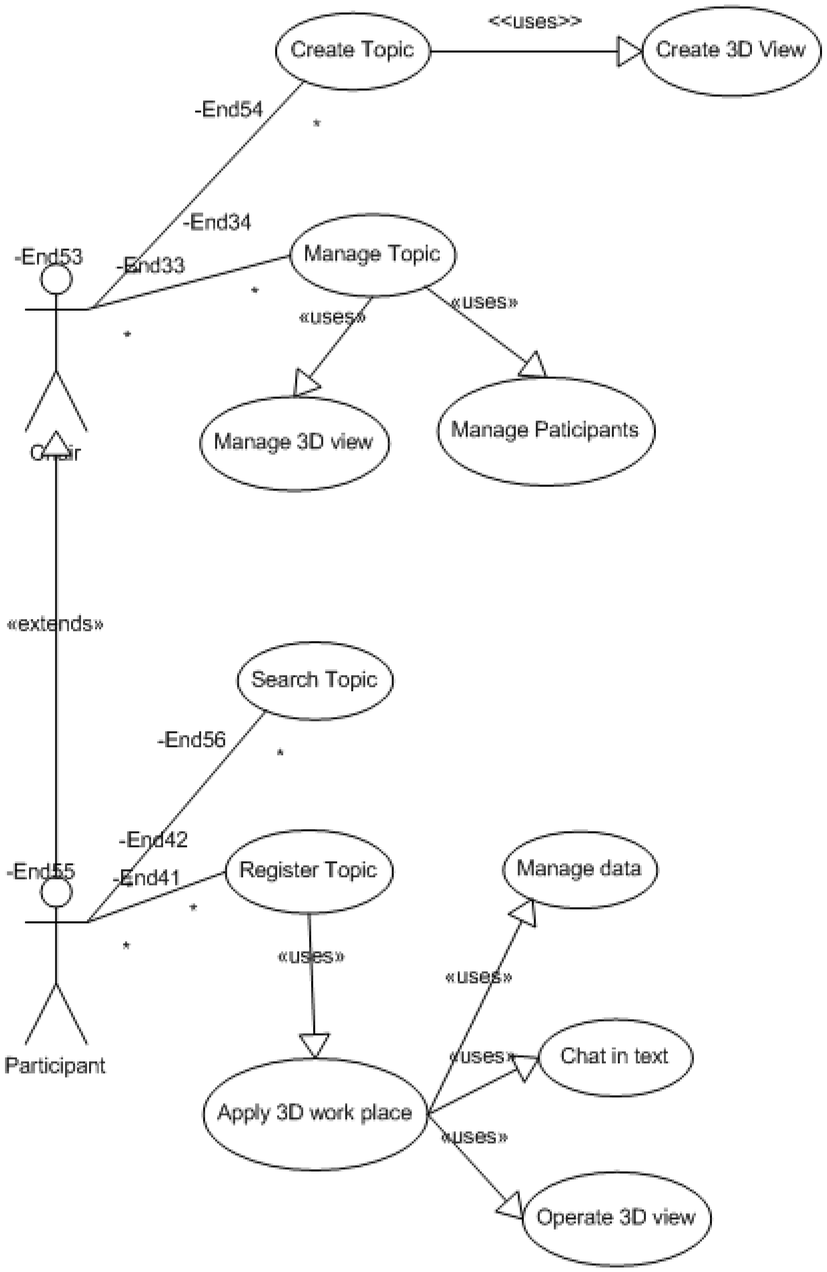
Handling shared data source User Case



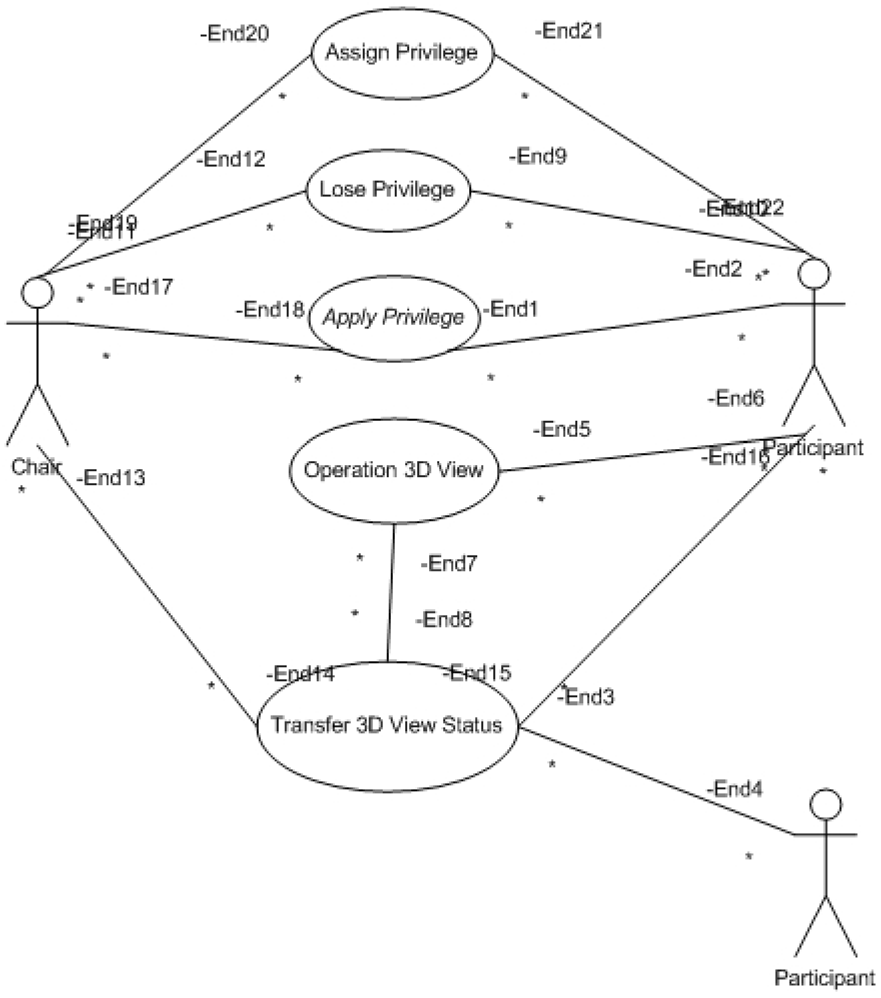
Later Comer User Case



Search and registration use case



Sharing operations in 3D GIS view use case



APPENDIX 3 OWL/XML Presentations for Social Collaboration Model

```
<?xml version="1.0"?>


<!DOCTYPE Ontology [
    <!ENTITY owl "http://www.w3.org/2002/07/owl#" >
    <!ENTITY owl11 "http://www.w3.org/2006/12/owl11#" >
    <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#" >
    <!ENTITY owl11xml "http://www.w3.org/2006/12/owl11-xml#" >
    <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" >
    <!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#" >
    <!ENTITY                                                OntologySC3DGIS
"http://www.semanticWeb.org/ontologies/2008/OntologySC3DGIS.owl#" >
    <!ENTITY                                               OntologySC3DGIS2
"http://www.semanticWeb.org/ontologies/2008/OntologySC3DGIS.owl#3" >
]>


<!-- Axioms: 68 -->

<Ontology xmlns="http://www.w3.org/2006/12/owl11-xml#"
     xml:base="http://www.w3.org/2006/12/owl11-xml#"

xmlns:OntologySC3DGIS="http://www.semanticWeb.org/ontologies/2008/OntologySC3DGIS.owl#"
     xmlns:owl11="http://www.w3.org/2006/12/owl11#"
     xmlns:owl11xml="http://www.w3.org/2006/12/owl11-xml#"
     xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
     xmlns:OntologySC3DGIS2="&OntologySC3DGIS;3"
     xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
     xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
     xmlns:owl="http://www.w3.org/2002/07/owl#"
     URI="http://www.semanticWeb.org/ontologies/2008/OntologySC3DGIS.owl">
    <DataPropertyRange>
        <DataProperty URI="&OntologySC3DGIS;hasFloorTypeTimeGap"/>
        <Datatype URI="&xsd;time"/>
    </DataPropertyRange>
    <ObjectPropertyRange>
        <ObjectProperty URI="&OntologySC3DGIS;hasLoadedData"/>
        <OWLClass URI="&OntologySC3DGIS;DataSource"/>
    </ObjectPropertyRange>
    <DisjointClasses>
    <OWLClass URI="&OntologySC3DGIS;DataSource"/>
    <OWLClass URI="&OntologySC3DGIS;Participant"/>
</DisjointClasses>
<DataPropertyDomain>
    <DataProperty URI="&OntologySC3DGIS;hasFloorTypeTimeGap"/>
    <OWLClass URI="&OntologySC3DGIS;FloorType"/>
</DataPropertyDomain>
<DisjointClasses>
    <OWLClass URI="&OntologySC3DGIS;Container"/>
    <OWLClass URI="&OntologySC3DGIS;Topic"/>
</DisjointClasses>
<DisjointClasses>
    <OWLClass URI="&OntologySC3DGIS;Container"/>
    <OWLClass URI="&OntologySC3DGIS;System"/>
</DisjointClasses>
<Declaration>
    <DataProperty URI="&OntologySC3DGIS;hasTopicNote"/>
</Declaration>
<SubClassOf>
    <OWLClass URI="&OntologySC3DGIS;FloorType"/>
    <OWLClass URI="&owl;Thing"/>
</SubClassOf>
<ObjectPropertyRange>
    <ObjectProperty URI="&OntologySC3DGIS;hasSystem"/>
    <OWLClass URI="&OntologySC3DGIS;System"/>
</ObjectPropertyRange>
<SubClassOf>
    <OWLClass URI="&OntologySC3DGIS;DataSource"/>
    <OWLClass URI="&owl;Thing"/>
</SubClassOf>
<DisjointClasses>
    <OWLClass URI="&OntologySC3DGIS;System"/>
    <OWLClass URI="&OntologySC3DGIS;Participant"/>
</DisjointClasses>
<DataPropertyRange>
    <DataProperty URI="&OntologySC3DGIS;hasTopicNote"/>
    <Datatype URI="&xsd;string"/>
</DataPropertyRange>
<SubClassOf>
    <OWLClass URI="&OntologySC3DGIS;3DGISTask"/>
    <OWLClass URI="&OntologySC3DGIS;Task"/>
</SubClassOf>
<SubClassOf>
    <OWLClass URI="&OntologySC3DGIS;Identity"/>
    <OWLClass URI="&OntologySC3DGIS;3DGISTask"/>
</SubClassOf>
<DisjointClasses>
    <OWLClass URI="&OntologySC3DGIS;System"/>
    <OWLClass URI="&OntologySC3DGIS;Task"/>
</DisjointClasses>
<SubClassOf>
    <OWLClass URI="&OntologySC3DGIS;Pan"/>
    <OWLClass URI="&OntologySC3DGIS;3DGISTask"/>
</SubClassOf>
<SubClassOf>
    <OWLClass URI="&OntologySC3DGIS;RomoveData"/>
    <OWLClass URI="&OntologySC3DGIS;3DGISTask"/>
</SubClassOf>
<DisjointClasses>
    <OWLClass URI="&OntologySC3DGIS;Container"/>
    <OWLClass URI="&OntologySC3DGIS;Participant"/>
</DisjointClasses>
<SubClassOf>
    <OWLClass URI="&OntologySC3DGIS;3DGIS"/>
    <OWLClass URI="&OntologySC3DGIS;System"/>
</SubClassOf>
<ObjectPropertyDomain>
    <ObjectProperty URI="&OntologySC3DGIS;hasTask"/>
    <OWLClass URI="&OntologySC3DGIS;System"/>
</ObjectPropertyDomain>
<ObjectPropertyDomain>
    <ObjectProperty URI="&OntologySC3DGIS;hasFloorType"/>
    <OWLClass URI="&OntologySC3DGIS;Container"/>
</ObjectPropertyDomain>
<DisjointClasses>
    <OWLClass URI="&OntologySC3DGIS;Container"/>
    <OWLClass URI="&OntologySC3DGIS;FloorType"/>
</DisjointClasses>
<DisjointClasses>
    <OWLClass URI="&OntologySC3DGIS;Task"/>
    <OWLClass URI="&OntologySC3DGIS;FloorType"/>
</DisjointClasses>
<DisjointClasses>
    <OWLClass URI="&OntologySC3DGIS;System"/>
    <OWLClass URI="&OntologySC3DGIS;FloorType"/>
</DisjointClasses>
<SubClassOf>
    <OWLClass URI="&OntologySC3DGIS;Container"/>
```

```
        <OWLClass URI="&owl;Thing"/>
</SubClassOf>
<Declaration>
        <DataProperty URI="&OntologySC3DGIS;hasFloorTypeTimeGap"/>
</Declaration>
<SubClassOf>
        <OWLClass URI="&OntologySC3DGIS;Audiance"/>
        <OWLClass URI="&OntologySC3DGIS;Participant"/>
</SubClassOf>
<ObjectPropertyRange>
        <ObjectProperty URI="&OntologySC3DGIS;hasRegisteredTopic"/>
        <OWLClass URI="&OntologySC3DGIS;Topic"/>
</ObjectPropertyRange>
<DisjointClasses>
        <OWLClass URI="&OntologySC3DGIS;Webservice"/>
        <OWLClass URI="&OntologySC3DGIS;Database"/>
</DisjointClasses>
<SubClassOf>
        <OWLClass URI="&OntologySC3DGIS;Webservice"/>
        <OWLClass URI="&OntologySC3DGIS;DataSource"/>
</SubClassOf>
<DataPropertyRange>
        <DataProperty URI="&OntologySC3DGIS;hasTopicID"/>
        <Datatype URI="&xsd;string"/>
</DataPropertyRange>
<DataPropertyDomain>
        <DataProperty URI="&OntologySC3DGIS;hasFloorTypeID"/>
        <OWLClass URI="&OntologySC3DGIS;FloorType"/>
</DataPropertyDomain>
<SubClassOf>
        <OWLClass URI="&OntologySC3DGIS;Chair"/>
        <OWLClass URI="&OntologySC3DGIS;Participant"/>
</SubClassOf>
<ObjectPropertyDomain>
        <ObjectProperty URI="&OntologySC3DGIS;hasLoadedData"/>
        <OWLClass URI="&OntologySC3DGIS;System"/>
</ObjectPropertyDomain>
<DisjointClasses>
        <OWLClass URI="&OntologySC3DGIS;Container"/>
        <OWLClass URI="&OntologySC3DGIS;DataSource"/>
</DisjointClasses>
<DisjointClasses>
        <OWLClass URI="&OntologySC3DGIS;Container"/>
        <OWLClass URI="&OntologySC3DGIS;Task"/>
```

```
</DisjointClasses>
<DataPropertyRange>
    <DataProperty URI="&OntologySC3DGIS;hasFloorTypeID"/>
    <Datatype URI="&xsd;integer"/>
</DataPropertyRange>
<DisjointClasses>
    <OWLClass URI="&OntologySC3DGIS;FloorType"/>
    <OWLClass URI="&OntologySC3DGIS;Topic"/>
</DisjointClasses>
<DisjointClasses>
    <OWLClass URI="&OntologySC3DGIS;System"/>
    <OWLClass URI="&OntologySC3DGIS;Topic"/>
</DisjointClasses>
<DisjointClasses>
    <OWLClass URI="&OntologySC3DGIS;DataSource"/>
    <OWLClass URI="&OntologySC3DGIS;Topic"/>
</DisjointClasses>
<SubClassOf>
    <OWLClass URI="&OntologySC3DGIS;Database"/>
    <OWLClass URI="&OntologySC3DGIS;DataSource"/>
</SubClassOf>
<SubClassOf>
    <OWLClass URI="&OntologySC3DGIS;Transform"/>
    <OWLClass URI="&OntologySC3DGIS;3DGIS"/>
</SubClassOf>
<ObjectPropertyRange>
    <ObjectProperty URI="&OntologySC3DGIS;hasFloorType"/>
    <OWLClass URI="&OntologySC3DGIS;FloorType"/>
</ObjectPropertyRange>
<Declaration>
    <DataProperty URI="&OntologySC3DGIS;hasFloorTypeID"/>
</Declaration>
<Declaration>
    <DataProperty URI="&OntologySC3DGIS;hasTopicID"/>
</Declaration>
<ObjectPropertyRange>
    <ObjectProperty URI="&OntologySC3DGIS;hasTask"/>
    <OWLClass URI="&OntologySC3DGIS;Task"/>
</ObjectPropertyRange>
<Declaration>
    <OWLClass URI="&OntologySC3DGIS;Transform"/>
</Declaration>
<SubClassOf>
    <OWLClass URI="&OntologySC3DGIS;ZoomOut"/>
```

```
        <OWLClass URI="&OntologySC3DGIS;3DGISTask"/>
</SubClassOf>
<SubClassOf>
        <OWLClass URI="&OntologySC3DGIS;Participant"/>
        <OWLClass URI="&owl;Thing"/>
</SubClassOf>
<DisjointClasses>
        <OWLClass URI="&OntologySC3DGIS;Topic"/>
        <OWLClass URI="&OntologySC3DGIS;Participant"/>
</DisjointClasses>
<SubClassOf>
        <OWLClass URI="&OntologySC3DGIS;File"/>
        <OWLClass URI="&OntologySC3DGIS;DataSource"/>
</SubClassOf>
<DisjointClasses>
        <OWLClass URI="&OntologySC3DGIS;DataSource"/>
        <OWLClass URI="&OntologySC3DGIS;FloorType"/>
</DisjointClasses>
<DataPropertyDomain>
        <DataProperty URI="&OntologySC3DGIS;hasTopicNote"/>
        <OWLClass URI="&OntologySC3DGIS;Topic"/>
</DataPropertyDomain>
<SubClassOf>
        <OWLClass URI="&OntologySC3DGIS;ZoomIn"/>
        <OWLClass URI="&OntologySC3DGIS;3DGISTask"/>
</SubClassOf>
<DisjointClasses>
        <OWLClass URI="&OntologySC3DGIS;DataSource"/>
        <OWLClass URI="&OntologySC3DGIS;System"/>
</DisjointClasses>
<ObjectPropertyDomain>
        <ObjectProperty URI="&OntologySC3DGIS;hasSystem"/>
        <OWLClass URI="&OntologySC3DGIS;Container"/>
</ObjectPropertyDomain>
<SubClassOf>
        <OWLClass URI="&OntologySC3DGIS;LoadData"/>
        <OWLClass URI="&OntologySC3DGIS;3DGISTask"/>
</SubClassOf>
<DisjointClasses>
        <OWLClass URI="&OntologySC3DGIS;DataSource"/>
        <OWLClass URI="&OntologySC3DGIS;Task"/>
</DisjointClasses>
<DataPropertyDomain>
        <DataProperty URI="&OntologySC3DGIS;hasTopicID"/>
        <OWLClass URI="&OntologySC3DGIS;Topic"/>
    </DataPropertyDomain>
    <SubClassOf>
        <OWLClass URI="&OntologySC3DGIS;Topic"/>
        <OWLClass URI="&owl;Thing"/>
    </SubClassOf>
    <SubClassOf>
        <OWLClass URI="&OntologySC3DGIS;System"/>
        <OWLClass URI="&owl;Thing"/>
    </SubClassOf>
    <SubClassOf>
        <OWLClass URI="&OntologySC3DGIS;Presenter"/>
        <OWLClass URI="&OntologySC3DGIS;Participant"/>
    </SubClassOf>
    <SubClassOf>
        <OWLClass URI="&OntologySC3DGIS;Task"/>
        <OWLClass URI="&owl;Thing"/>
    </SubClassOf>
    <DisjointClasses>
        <OWLClass URI="&OntologySC3DGIS;Task"/>
        <OWLClass URI="&OntologySC3DGIS;Topic"/>
    </DisjointClasses>
    <DisjointClasses>
        <OWLClass URI="&OntologySC3DGIS;Webservice"/>
        <OWLClass URI="&OntologySC3DGIS;File"/>
    </DisjointClasses>
    <DisjointClasses>
        <OWLClass URI="&OntologySC3DGIS;FloorType"/>
        <OWLClass URI="&OntologySC3DGIS;Participant"/>
    </DisjointClasses>
    <DisjointClasses>
        <OWLClass URI="&OntologySC3DGIS;File"/>
        <OWLClass URI="&OntologySC3DGIS;Database"/>
    </DisjointClasses>
    <DisjointClasses>
        <OWLClass URI="&OntologySC3DGIS;Task"/>
        <OWLClass URI="&OntologySC3DGIS;Participant"/>
    </DisjointClasses>
</Ontology>
```

APPENDIX 4 3D Coordinate Transformations and Usage Example in Java3D

An affine transformation is any transformation that preserves collinearity (i.e., all points lying on a line initially still lie on a line after transformation) and ratios of distances (e.g., the midpoint of a line segment remains the midpoint after transformation). In homogeneous coordinates, 3D affine transformations can be represented by 4×4 matrices (see Equation 1):

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} a & b & c & t_x \\ d & e & f & t_y \\ g & h & i & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \qquad (1)$$

According to the Java 3D documentation, "An affine matrix can translate, rotate, reflect, scale, and shear. Lines remain straight, and parallel lines remain parallel, but the angle between intersecting lines can change. In order for a transform to be classified as affine, the 4th row must be: [0, 0, 0, 1]." More references can be found in Baldwin (2007).

For example, a zooming in/out operation on a 3D model is a 3D scaling transformation. The detailed equations and functions can be found in Equation 2, which presents the scaling operation in X, Y and Z axis

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} a & 0 & 0 & 0 \\ 0 & b & 0 & 0 \\ 0 & 0 & c & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \qquad (2)$$

A rotation operation in a 3D environment is a rotation transformation. Equation 3 presents a rotation among X axis.

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad (3)$$

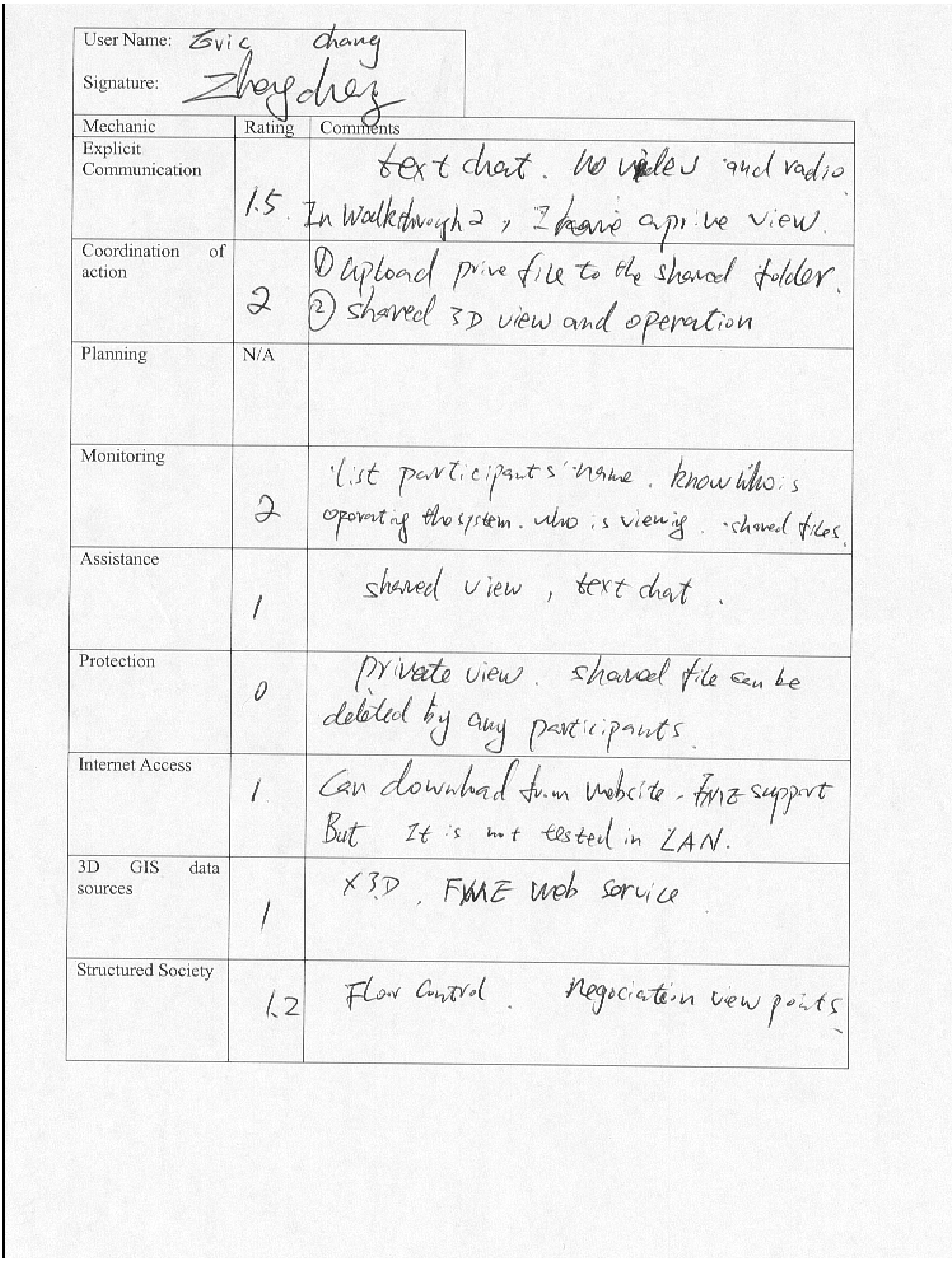
A movement of the 3D object is a translation transformation. Equation 4 presents the translation in a, b and c distance alone X, Y and Z axis.

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad (4)$$

When the 3D view is navigated and explored, the viewpoints to the 3D model are changed through the operations of a mouse or a key board. In this prototype, any viewpoint can be transformed to another viewpoint through the combined transformation such as translation, scaling and rotation transformations. Following codes present how to rotate a 3D viewpoint among y, x and z Axis with angles xAngle, yAngle and zAngle in Java3D. The detailed codes can be found in Appendix 6, code example 1.

```
// set 3D viewpoint to tVect
tVect.set(platformVect);
// set x axis rotation matrices
tXRot.set(new AxisAngle4d(1.0,0.0,0.0,Math.toRadians(xAngle)));
// set y axis rotation matrices
tYRot.set(new AxisAngle4d(0.0,1.0,0.0,Math.toRadians(yAngle)));
// set z axis rotation matrices
tZRot.set(new AxisAngle4d(0.0,0.0,1.0,Math.toRadians(zAngle)));
// rotate 3D viewpoint yAngle among YAxis
tVect.mul(tYRot);
// rotate 3D viewpoint xAngle among xAxis
tVect.mul(tXRot);
// rotate 3D viewpoint zAngle among zAxis
tVect.mul(tZRot);
```

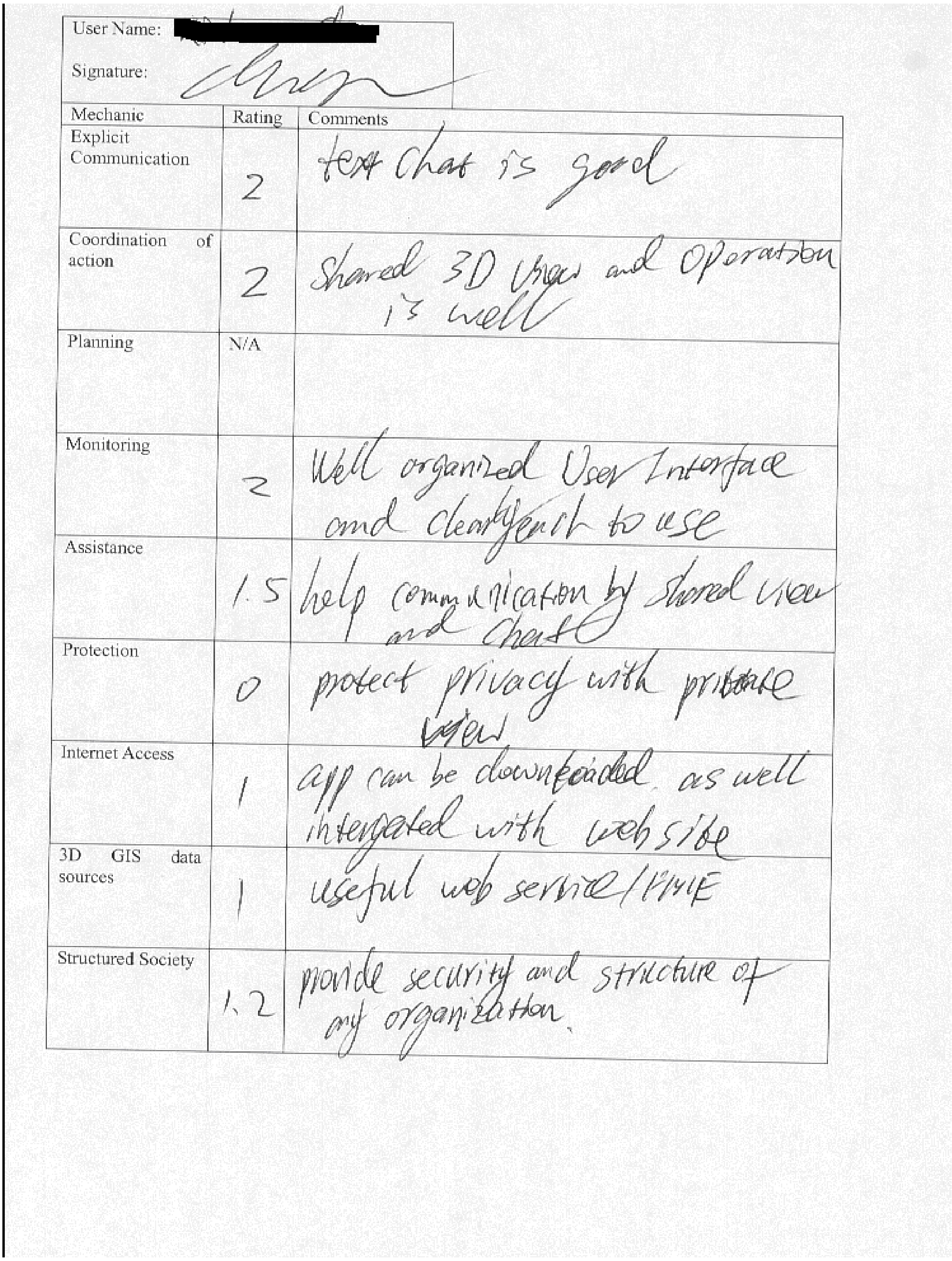
APPENDIX 5 Examples of Walkthrough Evaluation Forms



User Name: Evic chang

Signature:

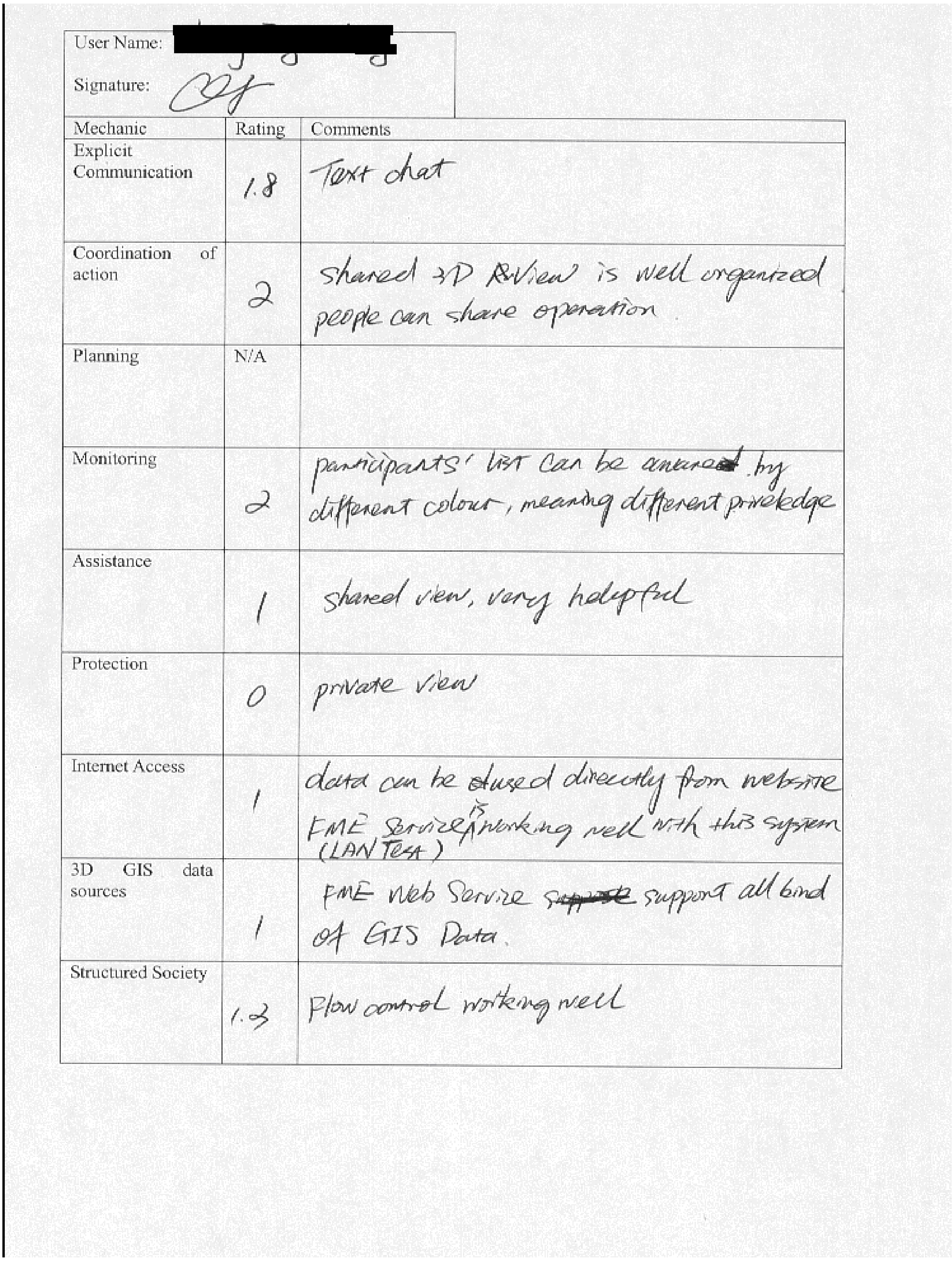
Mechanic	Rating	Comments
Explicit Communication	1.5	text chat. no video and radio. In Walkthrough 2, I have a prive view.
Coordination of action	2	① upload prive file to the shared folder. ② shared 3D view and operation
Planning	N/A	
Monitoring	2	list participants' name. know who is operating the system. who is viewing. shared files.
Assistance	1	shared view, text chat.
Protection	0	private view. shared file can be deleted by any participants.
Internet Access	1.	Can download from website. FME support But It is not tested in LAN.
3D GIS data sources	1	X3D, FME web service.
Structured Society	1.2	Flow Control. Negociation view points.



User Name: [redacted]

Signature:

Mechanic	Rating	Comments
Explicit Communication	2	text chat is good
Coordination of action	2	Shared 3D View and Operation is well
Planning	N/A	
Monitoring	2	Well organized User Interface and clear/each to use
Assistance	1.5	help communication by shared view and chat
Protection	0	protect privacy with private view
Internet Access	1	app can be downloaded as well integrated with website
3D GIS data sources	1	useful web service/PMIE
Structured Society	1.2	provide security and structure of my organization.



User Name: [redacted]

Signature:

Mechanic	Rating	Comments
Explicit Communication	1.8	Text chat
Coordination of action	2	shared 3D RView is well organized people can share operation.
Planning	N/A	
Monitoring	2	participants' list can be arranged by different colour, meaning different privekdge
Assistance	1	shared view, very helpful
Protection	0	private view
Internet Access	1	data can be used directly from website FME Service is working well with this system (LAN Test)
3D GIS data sources	1	FME web Service ~~suppose~~ support all kind of GIS Data.
Structured Society	1.3	Flow control working well

APPENDIX 6 Code Examples

Code Example 1 : 3D Environmental Platform

```
package com.geolink3d.dimension3;

/**
 * <p>Title: </p>
 * <p>Description: </p>
 * <p>Copyright: Copyright (c) 2006</p>
 * <p> </p>
 * @author Eric Chang
 * @version 1.0
 */


import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import javax.swing.border.*;
import java.text.*;

import javax.vecmath.*;
import javax.media.j3d.*;
import com.sun.j3d.utils.behaviors.vp.*;
import java.util.Vector;
import java.util.Enumeration;
import java.awt.event.MouseWheelEvent;
import java.awt.event.MouseWheelListener;

import com.geolink3d.collaboration.FloorOperation;
import com.geolink3d.util.GraphPaperLayout;

public class FlyingPlatform extends ViewPlatformAWTBehavior implements
ItemListener,ActionListener, MouseWheelListener
{

/** base XAxis attitude determines whether you are climbing or diving */
private static final int HOME_XANGLE = 0;
/** base YAxis attitude determines whether you are flying east/west/north/south
*/
private static final int HOME_YANGLE = 0;
/** base ZAxis attitude determines whether you are banking left or right */
private static final int HOME_ZANGLE = 0;
```

```
/** home X location */
private   double HOME_X = 15; //300;//-1999;
/** home Y location */
private   double HOME_Y = 0;
/** home Z location */
private   double HOME_Z = 46; //963;//9639;

 /** holds view platform location*/
 private Vector3d platformVect;
 // world center
 private Vector3d worldCenter;
 //private Vector3f eyeCenter;
 private Vector3d eyeCenter;

 private double xAngleP = 1;
/** holds current X axis attitude */
 private double xAngle = HOME_XANGLE; // degrees
 private double xAngleHome = HOME_XANGLE; //
/** holds current Y axis attitude */
 private double yAngle = HOME_YANGLE; // degrees
  private double yAngleHome = HOME_YANGLE; //
/** holds current Z axis attitude */
 private double zAngle = HOME_ZANGLE; // degrees
  private double zAngleHome = HOME_ZANGLE; //

 private int oldx = -1, oldy = -1;
 private static int sensitivity = 3;

 private Transform3D  Trans3D;
 private double[] transArray;
 private Vector m_listeners = new Vector();

 /** amount to move (in meters) on each operation */
private double moveAmt = 1;
 /** amount to turn(in degrees) on each operation */
 private double turnAmt = 5.0d; // degrees to turn

 private static  float  INITIAL_TERRAIN_FOLLOW_ALTITUDE = 100;
 private static  float MINIMUM_ALTITUDE = 0;
 private boolean followTerrain = false; // terrain following enabled/disabled
 private float terrainFollowAltitude; // how high to maintain platform
//
//  popup menu controls
//
```

```
 private PopupMenu popupMenu = new PopupMenu();
 private CheckboxMenuItem terrainFollowMenu = new CheckboxMenuItem("Terrain
following");
 private MenuItem settingsMenu = new MenuItem("Navigation panel classic");
 private MenuItem settingsMenu2 = new MenuItem("Navigation panel advance");

 private MenuItem homeBaseMenu = new MenuItem("Return to home Base");
 private MenuItem levelOffMenu = new MenuItem("Level off");
 private MenuItem aerialViewMenu = new MenuItem("Aerial view");


 private ElevationModelInterface model; // call back to get terrain information
 private Canvas3D canvas;  // canvas object

 private SettingsDialog settingsDialog; // pop settings dialog
 private SettingNaviDialog settingnaviDialog;

 private boolean isOperating=true;
 private FloorOperation floorOp=null;
 //private boolean isMouseMoving = false;
/**
 *  Create the flying platform
 *  @param aCanvas Canvas3D object that is used to display the world
 *
 */
public FlyingPlatform(Canvas3D aCanvas, ElevationModelInterface aModel)
{
  super(aCanvas,MOUSE_MOTION_LISTENER|MOUSE_LISTENER|KEY_LISTENER);
  aCanvas.requestFocus();  // get the focus to the Canvas, allows keyboard
inputs
  model = aModel;
  canvas = aCanvas;
  canvas.addMouseWheelListener(this);
  //HOME_Y = model.getElevationAt(0,0)+INITIAL_TERRAIN_FOLLOW_ALTITUDE;
  HOME_Y = INITIAL_TERRAIN_FOLLOW_ALTITUDE;
  //moveAmt = Math.round(model.getModelLength()/100); //temp
  //moveAmt = Math.max(moveAmt%10, moveAmt-(moveAmt%10)); //temp
  platformVect = new Vector3d(HOME_X,HOME_Y,HOME_Z);
  // try ....
  /* changing...............................
 // worldCenter = aModel.getModelConter(); /// temp
//pull the eye back far enough to see the whole object

  //float radius = aModel.getRadus();
  //float eyeDist = (float)(1.4 * radius / Math.tan( Math.toRadians( 40 ) /
2.0 ));

  //Vector3f temp = new Vector3f();
 // temp.x = 0;
  //temp.y = 0;
  //temp.z = eyeDist;

  //eyeCenter = temp;
  */
  worldCenter = new Vector3d(0,0,0);

  eyeCenter = worldCenter;
  setXAngleP();

  Container c = canvas.getParent();

  while(c.getParent() != null)
     c= c.getParent();
  settingsDialog = new SettingsDialog((JPanel)c);
  settingnaviDialog = new SettingNaviDialog((JPanel)c);

  popupMenu.add(settingsMenu);
  popupMenu.add(settingsMenu2);
  popupMenu.add(levelOffMenu);
  popupMenu.add(terrainFollowMenu);
  popupMenu.addSeparator();
  popupMenu.add(aerialViewMenu);
  popupMenu.add(homeBaseMenu);
  canvas.add(popupMenu);

  terrainFollowMenu.addItemListener(this);
  settingsMenu.addActionListener(this);
  settingsMenu2.addActionListener(this);

  homeBaseMenu.addActionListener(this);
  levelOffMenu.addActionListener(this);
  aerialViewMenu.addActionListener(this);
}

// get the view transform array
//public TransformGroup[] getViewTransformGroupArray(BranchGroup
sceneGroup )
public Transform3D getViewTransformGroupArray(BranchGroup sceneGroup )
```

```
{

    TransformGroup[] tgArray = new TransformGroup[1];
    tgArray[0] = new TransformGroup( );

    Transform3D viewTrans = new Transform3D( );
    Transform3D eyeTrans = new Transform3D( );

    BoundingSphere sceneBounds = (BoundingSphere) sceneGroup.getBounds( );

    // point the view at the center of the object
    Point3d center = new Point3d( );
    sceneBounds.getCenter( center );
    // set home

    //eyeCenter = (Point3f)center;
    double radius = sceneBounds.getRadius( );
    Vector3d temp = new Vector3d( center );

    //eyeCenter = temp;

    viewTrans.set( temp );

    // pull the eye back far enough to see the whole object
    double eyeDist = 1.1 * radius / Math.tan( Math.toRadians( 40 ) / 2.0 );
    //double eyeDist = 1 * radius / Math.tan( Math.toRadians( 40 ) / 2.0 );
    //double eyeDist = radius;

    System.err.println("dis:" +  eyeDist + "X:"+ temp.x + "Y:"+ temp.y + "Z:"+
temp.z + "r:" +radius);
    temp.x = 0.0;
    temp.y = 0.0;
    temp.z = eyeDist;
    //temp.z =
    eyeTrans.set( temp );
    viewTrans.mul( eyeTrans );

    // set ini variables
    HOME_X = center.x;
    HOME_Y = center.y;
    HOME_Z = center.z + eyeDist;
    eyeCenter.x = center.x;
    eyeCenter.y = center.y;
    eyeCenter.z = center.z;
    platformVect.x= HOME_X;
    platformVect.y= HOME_Y;
    platformVect.z= HOME_Z;

    return viewTrans;
    // set the view transform
    //tgArray[0].setTransform( viewTrans );

    //return tgArray;
}

/**
 *  reset the viewplatform transformation based on
 * the x,y,z rotation and location information.
 *
 */

protected void integrateTransforms()
{
   Transform3D tVect = new Transform3D();
   Transform3D tXRot = new Transform3D();
   Transform3D tYRot = new Transform3D();
   Transform3D tZRot = new Transform3D();

   // shows ini view
  System.err.println("PlatFormV; " + platformVect.toString());
  System.err.println("xAngle: " + xAngle +"yAngle: " + yAngle +"zAngle: " +
zAngle );

   tVect.set(platformVect);
   tXRot.set(new AxisAngle4d(1.0,0.0,0.0,Math.toRadians(xAngle)));
   tYRot.set(new AxisAngle4d(0.0,1.0,0.0,Math.toRadians(yAngle)));
   tZRot.set(new AxisAngle4d(0.0,0.0,1.0,Math.toRadians(zAngle)));
   tVect.mul(tYRot);
   tVect.mul(tXRot);
   tVect.mul(tZRot);


   targetTransform = tVect;
   //vp.getViewPlatformTransform().setTransform(tVect);

   transArray = new double [ 16 ];
   tVect.get(transArray);
   notifyListeners(transArray);
}
public double[] Send3DTransformAction(){
   return transArray;
  }

  public void addPositionListener(PositionListener listener)
  {
    if(!m_listeners.contains(listener)) {
      m_listeners.addElement(listener);
    }
  }

  public void removePositionListener(PositionListener listener)
  {
    m_listeners.removeElement(listener);
  }

  private void notifyListeners(double[] trans3D)
  {
    Vector copyOfListeners = (Vector)(m_listeners.clone());
    PositionEvent posEvent = new PositionEvent(this, trans3D);
    Enumeration enum1 = copyOfListeners.elements();
    while(enum1.hasMoreElements()) {
      PositionListener listener = (PositionListener)enum1.nextElement();
      listener.positionChanged(posEvent);
    }
  }


public void actionPerformed(ActionEvent e)
{
    if (!isOperating){
        return;
    }
    if(e.getSource() == homeBaseMenu)
         //goHomeNew();
        goHome();

    if(e.getSource() == levelOffMenu)
     levelOff();

   if(e.getSource() == settingsMenu)
     settingsDialog.setVisible(true);
   if(e.getSource() == settingsMenu2)
     settingnaviDialog.setVisible(true);

   if(e.getSource() == aerialViewMenu)
     aerialView();
}

public void levelOff()
{
           xAngle = 0;
           zAngle = 0;
    integrateTransforms();
}
// moves the viewplatform to the default home position with a default angle

/**
 *  Moves the viewplatform to the default home position and
 *  turns off terrain following.
 */
public void goHome()
{
           xAngle = HOME_XANGLE;
           yAngle = HOME_YANGLE;
           zAngle = HOME_ZANGLE;

           platformVect.x = HOME_X;
           platformVect.y = HOME_Y;
           platformVect.z = HOME_Z;
           terrainFollowMenu.setState(false);
           followTerrain = false;
           integrateTransforms();


}

public void itemStateChanged(ItemEvent e)
{
       if(terrainFollowMenu.getState())
       {
        xAngle = 0;
        zAngle = 0;
        //platformVect.y = model.getElevationAt(platformVect.x,
platformVect.z) + INITIAL_TERRAIN_FOLLOW_ALTITUDE;
```

```
         platformVect.y = INITIAL_TERRAIN_FOLLOW_ALTITUDE;
         terrainFollowAltitude = INITIAL_TERRAIN_FOLLOW_ALTITUDE;
         integrateTransforms();
         followTerrain = true;
       }
       else
       {
        followTerrain = false;
       }

}
/**
 *  updates the amount of space (in meters) that the platform is advanced
 *  with each mouse move/arrow key event
 *  @param amt number of meters to move with each operation
 *
 */
public void setStepSize(double amt)
{
 moveAmt = amt;
}
/**
 * returns the setsize
 * @return moveAmt in meters
 */
public double getStepSize()
{
 return moveAmt;
}
/**
 *  updates the amount of rotation (in degrees) that the platform is rotated
 *  with each mouse move/arrow key event
 *  @param amt number of degrees to rotate with each operation
 *
 */

public void setRotateAmt(double amt)
{
 turnAmt = amt;
}
/**
 * returns the rotation amount
 * @return turnAmt
 */
```

```
public double getRotateAmount()
{
 return turnAmt;
}

/*
 * (non-Javadoc)
 * @see
com.sun.j3d.utils.behaviors.vp.ViewPlatformAWTBehavior#mouseWheelMoved(jav
a.awt.event.MouseWheelEvent)
 * move forward and move backward
 */
public void mouseWheelMoved(MouseWheelEvent e) {
    if (!isOperating){
        return;
    }
    int notches = e.getWheelRotation();
    moveForward(-notches*2);
}

/**
 * process keyboard input
 *
 *   up arrow - move forward
 *   down arrow - move backward
 *   left arrow - turn left
 *   right arrow - turn right
 *   U, u - increase altitude
 *   D, d - decrease altitude
 * @param e  keyboard event
 */
public void keyPressed(KeyEvent e)
{
 if (!isOperating){
        return;
  }
  switch(e.getKeyCode())
  {
   case KeyEvent.VK_UP: // move forward
        moveForward(moveAmt);
   break;
   case KeyEvent.VK_DOWN: // move backward
        moveForward(-moveAmt);
   break;
   case KeyEvent.VK_LEFT: // turn left
  increaseYRotate(turnAmt);
   break;
   case KeyEvent.VK_RIGHT: // turn right
  increaseYRotate(-turnAmt);
   break;
   case KeyEvent.VK_U:       // increase altitude
       increaseY(moveAmt);
   break;
   case KeyEvent.VK_D:       // decrease altitude
       increaseY(-moveAmt);
   break;

  }
}
/**
 * process mouse clicked event, check if it is the right button, if so, bring
 * up the popup menu.
 * @param e mouse event
 */
public void mouseClicked(MouseEvent e)
 {
 if (!isOperating){
       return;
  }
  int mods = e.getModifiersEx();
  boolean alt = (mods & MouseEvent.ALT_DOWN_MASK) != 0;
  boolean ctrl = (mods & MouseEvent.CTRL_DOWN_MASK) != 0;
  //
  // on a right click, pop a control menu
  //
 if(e.getButton() == MouseEvent.BUTTON3)
  popupMenu.show(canvas,e.getX(),e.getY());

 }
/**
 * process mouse moved event, just reset old mouse locations
 *
 * @param e mouse event
 */
public void mouseMoved(MouseEvent e)
 {
   if (floorOp!=null){
       floorOp.updateMouseEvent();
```

```
    }
    if (!isOperating){
       return;
    }
  oldx = -1;
  oldy = -1;

 }

public void setFloorOperation(FloorOperation fo){
    floorOp = fo;

}

/**
 * process mouseDragged event.  determine which buttons are down and
 * move the view platform accordingly.
 *  left button down, mouse moved up - go forward
 * left button down, mouse moved down - go backward
 * left button down, mouse moved right - turn right
 * left button down, mouse moved left - turn left
 * right button down, mouse moved up - increase altitude
 * right button down, mouse moved down - decrease altitude
 * right button down, mouse moved left - bank left
 * right button down, mouse moved right - bank right
 * both buttons down, move up - climb
 * both buttons down, move down - dive
 */
public void mouseDragged(MouseEvent e)
{
  if (!isOperating){
       return;
  }
  int mods = e.getModifiersEx();

  int x = e.getX();
  int y = e.getY();

  if(oldx < 0 || oldy < 0)
  {
   oldx = x;
   oldy = y;
   return;
  }
```

```
//
// skip the event if it moved just a little
//
  if(Math.abs(y-oldy) < sensitivity &&
    Math.abs(x-oldx) < sensitivity)
    return;
//
// first check to see if both buttons are down.
//
  if((mods & MouseEvent.BUTTON1_DOWN_MASK) != 0
     && (mods & MouseEvent.BUTTON3_DOWN_MASK) != 0)
  {
         if(y > oldy+sensitivity)
            increaseXRotate(turnAmt);

         if(y < oldy-sensitivity)
        increaseXRotate(-turnAmt);

    return;
  }
//
// process left only down
//
  if((mods & MouseEvent.BUTTON1_DOWN_MASK) != 0)
  {
       if(y > oldy+sensitivity) //mouse moves down screen
      moveForward(-moveAmt);

    if(y < oldy-sensitivity) // mouse moves up the screen
      moveForward(moveAmt);

    if(x > oldx+sensitivity)
      increaseYRotate(-turnAmt);

    if(x < oldx-sensitivity)
      increaseYRotate(turnAmt);
  }

//
// process right button down
//
  if((mods & MouseEvent.BUTTON3_DOWN_MASK) != 0)
   {
        if(y > oldy+sensitivity) // mouse moves down the screen
       increaseY(-moveAmt);

     if(y < oldy-sensitivity) // mouse moves up the screen
       increaseY(moveAmt);

     if(x > oldx+sensitivity)
       increaseZRotate(turnAmt);

     if(x < oldx-sensitivity)
       increaseZRotate(-turnAmt);
   }

  oldx = x;  // save for comparison on next mouse move
  oldy = y;
 }

/**
 * move the viewplatform forward by desired number of meters
 * forward implies in the direction that it is currently pointed.
 * if terrain following is enabled, then keep the altitude a steady
 * amount above the ground.
 * @param amt number of meters to move forward
 */
  public void moveForward(double amt)
  {
  //
  //  Calculate x,y,z movement
  //
  // Setup Transforms
         Transform3D tTemp = new Transform3D();
         Transform3D tXRot = new Transform3D();
         Transform3D tYRot = new Transform3D();
         Transform3D tZRot = new Transform3D();

         tXRot.set(new AxisAngle4d(1.0,0.0,0.0,Math.toRadians(xAngle)));
         tYRot.set(new AxisAngle4d(0.0,1.0,0.0,Math.toRadians(yAngle)));
         tZRot.set(new AxisAngle4d(0.0,0.0,1.0,Math.toRadians(zAngle)));
         tTemp.mul(tYRot);
         tTemp.mul(tXRot);
         tTemp.mul(tZRot);
     // move forward in z direction
     // this implies decreasing z since we are looking at the origin from the
pos z
```

```
      Vector3d tv = new Vector3d(0,0,-amt);
         tTemp.transform(tv);  // translates z movement into x,y,z movement
based on heading

   //
       //  set new values for the platform location vector
       // if terrain following is on, then find the terrain elevation at the
new x,z
       // coordinate and base the new altitude on that. Else use the computed
altitude.
       //
        if(followTerrain)
        {
          //platformVect.y = model.getElevationAt(platformVect.x+tv.x,
platformVect.z+tv.z)+terrainFollowAltitude;
         platformVect.y = terrainFollowAltitude;
        }
         else
           platformVect.y += tv.y;

         platformVect.x += tv.x;
         platformVect.z += tv.z;

         integrateTransforms();  // apply transformations
  }
/**
 *  Increase the Y axis rotation. This effects the heading of the platform
 *  value is clamped to 0-359.
 *  @param amt number of degrees to change the heading
 *
 */
  public void increaseYRotate(double amt)
  {
   yAngle += amt;
   if(yAngle >= 360)
      yAngle -= 360;
   if(yAngle < 0)
      yAngle += 360;

   integrateTransforms();
  }

/**
 *  Increase the X axis rotation. This effects the pitch (nose up/down) of the
```

```
platform
 *  value is clamped to -360 to 360.
 *  @param amt number of degrees to change the pitch
 *
 */
  public void increaseXRotate(double amt)
  {
   xAngle += amt;
   if(xAngle >= 360)
     xAngle -= 360;
   if(xAngle <= -360)
     xAngle += 360;

   integrateTransforms();
  }
/**
 *  Increase the Z axis rotation. This effects the bank/roll of the platform
 *  value is clamped to -360-360.
 *  @param amt number of degrees to change the bank
 *
 */
  public void increaseZRotate(double amt)
  {
   zAngle += amt;
   if(zAngle >= 360)
     zAngle -= 360;
   if(zAngle <= -360)
     zAngle += 360;

   integrateTransforms();
  }

  /**
 *  Increase the Y location. This effects the altitude of the platform.
 *
 *  @param amt number of degrees to change the altitude
 *
 */
public void increaseY(double amt)
  {
       platformVect.y += amt;
       integrateTransforms();
       if(followTerrain)
       {
        terrainFollowAltitude += amt;
        terrainFollowAltitude =
Math.max(MINIMUM_ALTITUDE,terrainFollowAltitude);
   }
  }
  public void Pan(double amt, int xyz){
    if (xyz ==1){
       platformVect.x += amt;
    }
    if (xyz ==2){
      platformVect.y += amt;
   }

    lookatDIR(platformVect,eyeCenter );

  }
  public void setEyeCenter(double amt, int xyz){
    if (xyz ==1){
       eyeCenter.x += amt;
    }
    if (xyz ==2){
       eyeCenter.y += amt;
    }
    if (xyz ==3){
      eyeCenter.z += amt;
   }



  }
  public void increaseXAngleP(double amt){
    xAngleP += Math.toRadians(amt);
    tiltAngle(0);
  }
  public double getXAngleP(){
    return xAngleP;
  }
  public void setXAngleP(){
    double dx = platformVect.x - eyeCenter.x;
    double dy = platformVect.y - eyeCenter.y;
    double dz = platformVect.z - eyeCenter.z;

    double r = Math.sqrt(dx * dx + dz * dz);
    if (r == 0){
      xAngleP = Math.PI/2;
    }
     xAngleP = Math.asin(dz/r);



  }

  public void lookatDIR(Vector3d oriPoint, Vector3d endPoint){
    platformVect.x = oriPoint.x;
    platformVect.y = oriPoint.y;
    platformVect.z = oriPoint.z;

    double dx = endPoint.x - oriPoint.x;
    double dy = endPoint.y - oriPoint.y;
    double dz = endPoint.z - oriPoint.z;

    if (dz == 0 && dx == 0){ // just move the camera to one point.
      return;
    }

    if (dz == 0 && dx!=0) {
      yAngle = 90;
      xAngle = (double) Math.toDegrees(Math.atan(dy/(Math.sqrt(dz*dz +
dx*dx))));
      integrateTransforms();
      return;
    }

    yAngle = (double) Math.toDegrees(Math.atan(dx/dz));
    xAngle = (double) Math.toDegrees(Math.atan(dy/(Math.sqrt(dz*dz +
dx*dx))));

    integrateTransforms();
  }
```

Code Example 2: GeoAgent

```
package com.geoAgents;

import java.util.Date;

import com.geoAgents.behavior.InitConcilationBehavior;
```

```java
import com.geoAgents.behavior.ResponseConciliationBehavior;
import com.geoAgents.ontology.Collaborative3DOntology;
import com.geoAgents.ontology.FloorType;
import com.geoAgents.ontology.InitQuery;
import com.geoAgents.ui.FloorSeetingUI;
import com.geoAgents.ui.Private3DView;

import jade.content.ContentElement;
import jade.content.ContentManager;
import jade.content.lang.Codec;
import jade.content.lang.Codec.CodecException;
import jade.content.lang.sl.SLCodec;
import jade.content.onto.Ontology;
import jade.content.onto.OntologyException;
import jade.content.onto.UngroundedException;
import jade.content.onto.basic.Action;
import jade.core.AID;
import jade.domain.DFService;
import jade.domain.FIPAException;
import jade.domain.FIPANames;
import jade.domain.FIPAAgentManagement.DFAgentDescription;
import jade.domain.FIPAAgentManagement.SearchConstraints;
import jade.domain.FIPAAgentManagement.ServiceDescription;
import jade.gui.GuiAgent;
import jade.gui.GuiEvent;
import jade.lang.acl.ACLMessage;
import jade.lang.acl.MessageTemplate;
import jade.lang.acl.MessageTemplate.MatchExpression;
import jade.util.leap.ArrayList;
import jade.util.leap.Iterator;
import jade.util.leap.List;

public class AgentAssistant extends GuiAgent{
     private ContentManager manager  = (ContentManager) getContentManager();
       // This agent "speaks" the SL language
       private Codec     codec    = new SLCodec();
       // This agent "knows" the Music-Shop ontology
       private Ontology   ontology = Collaborative3DOntology.getInstance();
       private AID       chairAID;
       private String SERVICETYPE = "Reconcilation";
       private String AGENTCHAIR = "Chair";
       Private3DView my3DView;
       AID [] AIDlist;
       AID myAID;
```

```
        DFAgentDescription[] dfds;

        public static final int InitSendMessage_EVENT = 1000; // init send
message
        public static final int ResponseSendMessage_EVENT = 1002; // response
message
        public static final int ResponseInitUsers_EVENT = 1003; // response
message
        public static final int ConfirmDeal_EVENT = 1001;

        //public static final int CancelDeal_EVENT = 1002;
        private InitQuery SendingMessage;
        private InitQuery IncomingMessage;

        private InitQuery CurrentMessage;

    protected void setup() {
        /*
         * register the language with the ContentManager
         */
        manager.registerLanguage(new SLCodec(),
FIPANames.ContentLanguage.FIPA_SL0);
        manager.registerOntology(ontology);
        /*
         *  Register chair itself
         */
        boolean registed = RegistFloorControlService(SERVICETYPE,AGENTCHAIR);
        if (!registed){
            this.doDelete();
        }

        chairAID = searchChair(SERVICETYPE,AGENTCHAIR);

        myAID = this.getAID();
        System.out.println("Local AID name: " + myAID);

        if (chairAID==null){
            System.out.println("Chair Agent not started, this assistant agent
is existing...");
            this.doDelete();
        }


        my3DView = new Private3DView(this);
        /*
         * Add behaviour:response other clients Init request
         */

        MessageTemplate acceptInitMT = new MessageTemplate(new
MatchExpression() {
            public boolean match(ACLMessage msg)
            {
                try
                {
                    ContentElement a =
AgentAssistant.this.getContentManager().extractContent(msg);


                    if (a instanceof Action )
                    {
                        Action aAction = (Action) a;
                        if (aAction.getAction() instanceof InitQuery){
                                return true;
                            }

                    }
                } catch (UngroundedException e)
                {
                    e.printStackTrace();
                } catch (CodecException e)
                {
                    e.printStackTrace();
                } catch (OntologyException e)
                {
                    e.printStackTrace();
                }
                return false;
            }
        });
        this.addBehaviour(new ResponseConciliationBehavior(this,
acceptInitMT));

        /*
         * Add behaviour: response other client's floor setting request
         */
    }
    @Override
    protected void onGuiEvent(GuiEvent ev) {
        // TODO Auto-generated method stub
        switch(ev.getType())
        {
        case InitSendMessage_EVENT:
            Iterator cloneParametersfloor = ev.getAllParameter();
            SendingMessage =(InitQuery)cloneParametersfloor.next();
            //sendMessage(SendingMessage);
            initSendMessage(SendingMessage);
            System.out.println("Init sending::::" +
SendingMessage.getIniM0());
            break;
        case ResponseInitUsers_EVENT:
            chairAID = searchChair(SERVICETYPE,AGENTCHAIR);
            System.out.println("Init users" );
            my3DView.updateAgentDFlist();
            break;
        //case ResponseSendMessage_EVENT:
        //  Iterator cloneParametersfloor1 = ev.getAllParameter();
        //  CurrentMessage =(InitQuery)cloneParametersfloor1.next();
        //  System.out.println("Response sending::::" +
SendingMessage.getIniTransForm());
        //  break;
        case ConfirmDeal_EVENT:
            break;
        }
    }
    public void updateIncomingContent(InitQuery incomingMessage){
        my3DView.updateIncomingMessage(incomingMessage);
    }
    public void updatePerformtive(int pt){
        my3DView.updatePerformtive(pt);
    }
    public void updateSender(String name){
        my3DView.updateIncomingSender(name);
    }
    public InitQuery getCurrentMessage(){
        return CurrentMessage;
    }
    public double[] getCurrentTransform(){
        return my3DView.get3DTransform();
        //double[] Trans3D = {0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15};
        //return Trans3D;
    }
    /*
    public ArrayList getCurrentTransform(){
        double[] Trans3D = {0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15};
        ArrayList myTransform = new ArrayList();
        for (int i=0; i<Trans3D.length;i++){
            myTransform.add(Trans3D[i]);
        }
        return myTransform;
    }
    */
    public AID getLocalAgentName(){
        return myAID;
    }
    /*
    private AID getDestinationAID(AID aidtext){
        int Alength = AIDlist.length;
        for (int i=0;i<Alength;i++){
            String name = AIDlist[i].getName();
            String AIDNAME = aidtext.getName();
            System.out.println("Agent AID name: " + name);
            System.out.println("Agent AID AID: " + AIDNAME);
            if (AIDNAME==name){
                return AIDlist[i];
            }
        }
        System.out.println("Agent:" + this.getName() + "Can not get destination
AID.");
        return null;

    }
    */
    private void initSendMessage(InitQuery message){
        ACLMessage msg = new ACLMessage(ACLMessage.REQUEST);
        //ACLMessage msg = new ACLMessage(message.getIniAction());
      msg.setProtocol(FIPANames.InteractionProtocol.FIPA_REQUEST);
      msg.setLanguage(FIPANames.ContentLanguage.FIPA_SL0);
      msg.setOntology(ontology.getName());
      //msg.setSender(this.getLocalAgentName());
//  We want to receive a reply in 10 secs
      msg.setReplyByDate(new Date(System.currentTimeMillis() + 10000));
      AID dAid= message.getIniReceiver();
      System.out.println("Agent AID receiver: " + dAid);
```

```
     msg.addReceiver(dAid);

     this.addBehaviour(new InitConcilationBehavior(this, msg,message));
   }
   protected void takeDown() {
//      Deregister from the yellow pages
        try {
           DFService.deregister(this);
        }
        catch (FIPAException fe) {
           fe.printStackTrace();
        }

        //      Close the GUI
        //floorcontrolGUI.dispose();
        //TakeDownAgent();
//      Printout a dismissal message
        System.out.println("Agent:" + this.getName() + "terminating.");
     }
   protected boolean RegistFloorControlService(String stype,String atype){
       // Register myself to DF, and subcribe counterparts,
       // This is a case study in a group meeting for urban planning
       //String serviceName = "Floor-Control-Service";
       String serviceName = stype;
       String agentName = atype;

       // Register the service
       System.out.println("Agent "+getLocalName()+" registering service
\""+serviceName+"\" of type \"reconcilation test\"");
       try {
          DFAgentDescription dfd = new DFAgentDescription();
          dfd.setName(getAID());

          ServiceDescription sd = new ServiceDescription();
          sd.setName(agentName);

          // sd.setType("weather-forecast");
          sd.setType(serviceName);
          // Agents that want to use this service need to "know" the
weather-forecast-ontology
          //sd.addOntologies("weather-forecast-ontology");
          //sd.addOntologies("FloorControl-ontology");

          //sd.addOntologies("FIPA-Agent-Management");
```

```
            // Agents that want to use this service need to "speak" the FIPA-SL
language
            sd.addLanguages(FIPANames.ContentLanguage.FIPA_SL0);

            dfd.addServices(sd);
            DFService.register(this, dfd);
            return true;
        }
        catch (FIPAException fe) {
            fe.printStackTrace();
            return false;
        }
    }
    /*
    public void RequireProfile(AID chairAID){
          ACLMessage msg = new ACLMessage(ACLMessage.INFORM);
          msg.setProtocol(FIPANames.InteractionProtocol.FIPA_REQUEST);
          msg.setLanguage(FIPANames.ContentLanguage.FIPA_SL0);
          msg.setOntology(ontology.getName());
          msg.addReceiver(chairAID);

          this.addBehaviour(new InitConcilationBehavior(this, msg));
    }
    */
    public void fillContent(ACLMessage msg,Action ac){
        try{
            manager.fillContent(msg,ac);
        }catch(Exception e){e.printStackTrace();}
    }
    public AID searchChair(String stype,String atype){

//      String serviceName = "Floor-Control-Service";
        String serviceName = stype;
        String agentName = atype;

        AID chair=null;
        DFAgentDescription dfd = new DFAgentDescription();
        ServiceDescription sd = new ServiceDescription();

        sd.setName(agentName);
        // sd.setType("weather-forecast");
        sd.setType(serviceName);
        dfd.addServices(sd);

        SearchConstraints sc = new SearchConstraints();
        // We want to receive 10 results at most
        sc.setMaxResults(new Long(2));

        try
        {
            dfds = DFService.search(this, dfd);
            String dfdcontent = dfds.toString();
            System.out.println("DF" + ":" + dfdcontent + "::" +dfds.length +
"::" );
            int length = dfds.length;

            if (length>0){
                chair = dfds[0].getName();
                System.out.println("AID chair:" + chair );
            } else {
                chair = null;
            }

            AIDlist = new AID[length];

            for (int i=0;i<length;i++){
                AIDlist[i] = dfds[i].getName();
                System.out.println("AID Name:" +  dfds[i].getName());

            }

        } catch (FIPAException e)
        {
            System.err.println(e.getClass());
            e.printStackTrace();
        }
        return chair;

    }

    public AID[] getAgentNamelist(){
        return AIDlist;
    }
}
```

Eric Zheng Chang’s CURRICULUM VITAE

Education

PhD, 2010, Geomatics Engineering, Department of Civil Engineering, *Ryerson University*, Canada

Topic: **Collaborative 3D GIS with multi-agent support**

Supervisor: Prof. Songnian Li

MSc, 2005, Geomatics Engineering, Department of Civil Engineering, *Ryerson University*, Canada

Topic: **Framework Design for the Collaborative GIS**

Supervisor: Prof. Songnian Li

MSc, 1997, State Key Lab for Information Engineering in Surveying, Mapping and Remote Sensing (LIESMARS), *Wuhan University*, China

Topic: **Time-space (4D) Database Design in Cadastre Management**

Supervisor: Prof. Jun Chen (President, National Geomatics Center of China)

Co-supervisor: Prof. Daosheng Du (LIESMRS)

BS, 1994, Surveying Engineering, College of Earth Science and Geomatics Engineering, *Wuhan University*, China

Publications

1. Chang, Z. and Li, S. [2009] Data Model Selection Criteria for Collaborative 3D GIS, ***24th International Cartographic Conference***, Nov. 15-21, Santiago, CHILE.
2. Chang, Z. and Li, S. [2009] Reconciling inconsistent perspectives in collaborative GIS using multi-agent method, ***17th International Conference on Geoinformatics***, Aug. 12-14, 2009 Fairfox USA
3. Chang, Z. and Li, S. [2008] Social Collaboration Model Support for Collaborative 3D GIS Environment Based On Multi-Agent Method, ***The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences***. Vol. XXXVII. Part B2. pp.1059-1063
4. Chang, Z. and Li, S. [2008] Architectural Design and Prototyping of a Web-based Synchronous Collaborative 3D GIS. ***Cartography and Geographic Information Science*** 35(2), pp. 117-132
5. Li, S, Guo, X. Ma, X. and Chang, Z. [2007] Towards GIS-enabled Virtual Public Meeting Space for Public Participation. ***PE&RS*** 73(4). pp. 641-650
6. Chang, Z. and Li, S. [2007] Collaboration enabled GIS Tools for Emergency Operation Centre, Geomatics Solutions for Disaster Management, ***Lecture Notes in Geoinformation and Cartography***, Springer-Verlag Berlin, Heidelberg, Pages. 149-163.
7. Chang, Z., Li, S., and Kirkwood, D. [2007] “Collaborative Sharing and Exploration of Geo-models”, ***9th GEOIDE Annual Scientific Conference***, Halifax, June 7-9, 2007 (Poster)
8. Chang, Z., Li, S., and Kirkwood, D. “3D Synchronous Collaborative Modeling”, ***8th GEOIDE Annual Scientific Conference***, Banff, May 31-June 2, 2006 (Poster)
9. Chang, Z. and Li, S. [2005] “A Framework design for collaborative GIS application: Based on Replicated Architecture”, ***98th Canadian Institute of Geomatics***, Ottawa, June 13~15, 2005.

10. Chang, Z. and Li, S. [2005] VRML-Based 3D Collaborative GIS: A Design Perspective, ***Web and Wireless Geographical Information Systems - Lecture Notes in Computer Science***, Volume 3428, Apr 2005, Pages 232 – 241
11. Li, S. and Chang, Z. [2005] “Developing Component-based Generic Tools for Group Visualizing, Manipulating and Exploring Spatial Information”, ***Proceedings of 2005 GeoTec Event***, Vancouver, B.C., February 13-16, 2005
12. Li, S., Ru, Y., and Chang, Z [2004] “GIS-based Internet NoticeBoard to Facilitate Public Participation in Municipal Developments”, ***Proceedings of the 20th ISPRS Annual Congress***, Istanbul, Turkey, July 12 – 23, 2004, 35(B2), pp. 269 – 274
13. Li. S., Gu, L., and Chang, Z. [2004] “On Mobile-enabled Collaboration GIS Systems”, ***Proceedings of the 4th International Symposium on Mobile Mapping Technology***, Kunming, China, March 29 – 31, 2004
14. Chang, Z. Zheng, X., Y. and Li, X., H. [2000] Urban Underground Pipeline Database Design and Development in Guangzhou City. ***Journal of Urban Construction Achieves***, 66(10); 98 (in Chinese)
15. Chen, J., Jiang, J. and Chang, Z. [1999] Spatio-Overlap and Time-Meet Relations in Land Subdivision, ***ISPRS "Dynamic and Multi-Dimensional GIS"***, Beijing, Volume 32, Part 4W12, pp. 137-140, Oct. 4-6, 1999
16. Chang, Z. Chen, J., and Du, D., [1997] Cadastral spatio-temporal database organization and query based on properties of land subdivision. ***Journal of Wuhan Technical University Survey and Mapping***, 22(3):216- 221 (in Chinese).