The Pennsylvania State University

The Graduate School

College of Earth and Mineral Sciences

SQSynC: SPATIAL QUERIES
IN SYNCHRONOUS COLLABORATION

A Thesis in
Geography
by
Kevin Scott Ross

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Master of Science

August 2010
The thesis of Kevin Scott Ross was reviewed and approved* by the following:

Alan M. MacEachren  
Professor of Geography  
Thesis Advisor

Alexander Klippel  
Assistant Professor of Geography

Karl S. Zimmerer  
Professor of Geography  
Head of the Department of Geography

*Signatures are on file in the Graduate School
ABSTRACT

Processes of knowledge-building, planning, and decision-making are frequently accomplished by groups of remotely located people working together. Research in the field of geocollaboration has been aimed at discovering ways to support this type of collaborative work using GIS technologies. One important topic that is yet to be explored is how to support synchronous collaborative spatial data queries. The goals of this research are (a) to develop a conceptual framework and create and assess the Spatial Queries in Synchronous Collaboration (SQSynC) proof-of-concept prototype to enable remotely located people to collaboratively query spatial data in real-time requiring only an internet connection and web browser and (b) to analyze the methods used by geographers to construct the queries in this environment to develop a deeper understanding of the process.

The SQSynC software prototype is a web-based application that can support both simultaneous and parallel collaborative activities between the users within the interface. In simultaneous mode, the software behaves as if there is only one interface that every user is concurrently interacting with, even though each user is remotely connected through individual computers. For instance, if one person were to pan and zoom the map, every other user would see the map change. In parallel mode, the software provides each user with a distinct, private interface that no other users can see or interact with. To enable more direct collaboration in the parallel mode, a second, public tab exists. Users can move items from their private tab over to the public tab in order to share them with the other users.
A user study was conducted, using the SQSynC prototype, to obtain targeted input from geographers regarding the two different approaches to collaborative spatial data query. Participants, working in counter-balanced pairs, were asked to collaboratively work through one set of tasks using the simultaneous application mode, then another set of tasks using the parallel mode or the reverse. In the simultaneous mode, each user could simultaneously interact with a shared query building interface. In the parallel mode users could communicate with each other and share data on the public tab, but there was no shared query building interface with which they could simultaneously interact, only a private interface. A total of 16 participants took part in the experiment and they each completed a follow-up survey to answer questions about which mode they preferred, report any critical incidents that occurred or features that they liked, and provide additional comments and suggestions. The main finding, along with all of the in-depth feedback from the participants, is that 56% of participants preferred the parallel mode of collaborative query building, while 44% preferred the simultaneous mode. The results of this experiment ultimately show that in geocollaboration software there is a place for both private and shared query building tools.
# TABLE OF CONTENTS

LIST OF FIGURES ........................................................................................................ vii

LIST OF TABLES ........................................................................................................... x

ACKNOWLEDGEMENTS ........................................................................................... xi

Chapter 1 INTRODUCTION ......................................................................................... 1

Chapter 2 LITERATURE REVIEW AND BACKGROUND ........................................... 5

2.1 Computer Supported Cooperative Work ......................................................... 5
2.2 Geocollaboration ............................................................................................... 7
  2.2.1 Community-oriented geocollaboration ................................................. 8
    Public participation GIS ............................................................................ 8
    Argumentation maps .............................................................................. 14
  2.2.2 Team-oriented geocollaboration ........................................................... 19
    Group spatial decision support design theories and examples .......... 22
    Visualization concepts ............................................................................ 25
    Same-time, different-place design theories and examples .................. 32
2.3 Collaborative Information Retrieval ............................................................... 38
2.4 Geocollaboration For Emergency Management ............................................ 42

Chapter 3 SOFTWARE DEVELOPMENT ................................................................. 48

3.1 Conceptual framework .................................................................................... 49
  3.1.1 Core features ......................................................................................... 49
  3.1.2 Spatial query features ........................................................................... 51
3.2 Requirements and solutions ............................................................................ 52
  3.2.1 Synchronous data transfer .................................................................... 53
  3.2.2 Public and private workspaces ............................................................. 56
  3.2.3 Communication .................................................................................... 58
  3.2.4 Geovisualization capabilities ............................................................... 60
  3.2.5 Usable query interface .......................................................................... 62
  3.2.6 Simultaneous and parallel approaches to collaborative spatial query .. 64
  3.2.7 Concurrency control ............................................................................ 66
3.3 Implementation results .................................................................................... 67

Chapter 4 USER STUDY ............................................................................................ 89

4.1 Methodology ................................................................................................... 89
  4.1.1 Participants ........................................................................................... 90
### Chapter 4: Results

#### 4.2 Results

<table>
<thead>
<tr>
<th>Subsection</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.2.1 Satisfaction and preferences</td>
<td>104</td>
</tr>
<tr>
<td>4.2.2 Critical incidents</td>
<td>110</td>
</tr>
<tr>
<td>Parallel approach</td>
<td>110</td>
</tr>
<tr>
<td>Simultaneous approach</td>
<td>111</td>
</tr>
<tr>
<td>4.2.3 Enjoyable features</td>
<td>112</td>
</tr>
<tr>
<td>Parallel approach</td>
<td>113</td>
</tr>
<tr>
<td>Simultaneous approach</td>
<td>113</td>
</tr>
<tr>
<td>4.2.4 Suggestions for improvements and other scenarios</td>
<td>114</td>
</tr>
</tbody>
</table>

### Chapter 5: Conclusion

<table>
<thead>
<tr>
<th>Subsection</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1 Discussion</td>
<td>117</td>
</tr>
<tr>
<td>5.1.1 Conceptual framework and prototype</td>
<td>118</td>
</tr>
<tr>
<td>5.1.2 Communication</td>
<td>118</td>
</tr>
<tr>
<td>5.1.3 Concurrency control</td>
<td>119</td>
</tr>
<tr>
<td>5.1.4 Missing features</td>
<td>119</td>
</tr>
<tr>
<td>5.1.5 Parallel approach versus simultaneous approach</td>
<td>120</td>
</tr>
<tr>
<td>5.2 Future work</td>
<td>122</td>
</tr>
</tbody>
</table>

### References

<table>
<thead>
<tr>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>126</td>
</tr>
</tbody>
</table>

### Appendixes

<table>
<thead>
<tr>
<th>Appendix</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>A Sample task given to participants to walk through during the tutorial video</td>
<td>143</td>
</tr>
<tr>
<td>B Task information sheet given to Participant 1 for the hypothetical emergency response scenario</td>
<td>144</td>
</tr>
<tr>
<td>C Task information sheet given to Participant 2 for the hypothetical emergency response scenario</td>
<td>145</td>
</tr>
<tr>
<td>D Task information sheet given to Participant 1 for the hypothetical real estate shopping scenario</td>
<td>146</td>
</tr>
<tr>
<td>E Task information sheet given to Participant 2 for the hypothetical real estate shopping scenario</td>
<td>147</td>
</tr>
</tbody>
</table>
LIST OF FIGURES

Figure 1. The four temporal and spatial categories of geocollaboration and an example application of each (MacEachren and Brewer, 2004) .......................... 1

Figure 2. Three-tier architecture for WPPS (Peng, 2001). ........................................... 11

Figure 3. Virtual Slaithwaite (source: http://www.ccg.leeds.ac.uk/projects/slaithwaite) ................................................. 12

Figure 4. Woodland Online Decision System (WOODS) (source: http://www.ccg.leeds.ac.uk/dales/) .............................................. 13

Figure 5. Argumentation map model (Keßler, Rinner, & Raubal, 2005). .................. 16

Figure 6. Keßler’s argumentation map prototype interface (Keßler et al., 2005). ..... 17

Figure 7. GeoDeliberator. (1) User and Group Panel, (2) Project Panel, (3) Map Panel, (4) Timeline View, and (5) Annotation Info Panel (Cai & Yu, 2009) ...... 18

Figure 8. “The GIS bottleneck” (Armstrong, 1994) ................................................... 21

Figure 9. EAST-2 framework. Adapted from (Jankowski & Nyerges, 2003) to include labels for the seven premises. .............................................. 23

Figure 10. Spatial Group Choice design architecture (Jankowski et al., 1997) ......... 24

Figure 11: Semiotic triangles for collaborative visualization (MacEachren & Brewer, 2004) ......................................................... 26

Figure 12: An updated version of DiBiase’s conceptual model for visual thinking and communication, extending it to consensus building in an evaluative realm (Armstrong & Densham, 2008) ........................................................................... 27

Figure 13. ImmersaDesk (MacEachren & Brewer, 2004) ................................. 30

Figure 14. HI-SPACE (MacEachren & Brewer, 2004). ........................................... 30

Figure 15. Prototype visual display interface (MacEachren & Brewer, 2004). ....... 32

Figure 16. Multiple-user, multiple view geocollaborative system (Convertino, Ganoe, Schafer, Yost, & Carroll, 2005). .............................................. 36

Figure 17. Illustration of an example scenario where DAVE_G is used (MacEachren & Cai, 2006) ................................................................. 46
Figure 18. Peer-to-peer model (left) and client-server model (right).............56

Figure 19. Tabbed interface for switching between the public view and the individual users’ views. .................................................................58

Figure 20. The SQSynC prototype in simultaneous mode. (1) Task Manager, (2) Whiteboard Tools, (3) Data Layers, (4) Connected Users List, (5) Query Editor, (6) Chat Panel.................................................................68


Figure 22. System architecture for SQSynC ..................................................70

Figure 23. A user named Kevin draws on the map. His cursor and his drawing are visible to the other users. The whiteboard menu is opened, showing the options to activate/deactivate and clear the whiteboard. .....................72

Figure 24. Query editor. ..............................................................................73

Figure 25. Layer select widget (for demonstration purposes, not all database layers are shown). ......................................................................74

Figure 26. Attribute field select widget. ..........................................................75

Figure 27. Two attribute fields have been added to the query: one has numeric data (top), the other has nonnumeric data (bottom).....................76

Figure 28. Two dropdown elements allow the user to select a type of spatial relationship (top) and layer with which this relationship exists (bottom). ........77

Figure 29. Query editor with attribute and spatial criteria set.......................78

Figure 30. Two layers have been added to the map: hospitals (points) and census tracts (polygons). Right-clicking on a layer name opens a context menu with additional options...............................................................80

Figure 31. Attribute table for census tracts, sorted by age_18_21. ....................81

Figure 32. A user named Kevin is typing in a value after clicking the ‘like’ button. .........................................................................................82

Figure 33. In the parallel approach, data layers are separated into private and public lists, and a new option exists in the right-click menu to share a private layer (make it public)......................................................84
Figure 34. Join Layers interface with data layers selected from two different sources .................................................................................................................. 85

Figure 35. Join Layers interface with data layers selected from the same original source ............................................................................................................. 86

Figure 36. Task Manager .................................................................................. 87

Figure 37. Within-subject comparison of responses to the first two survey questions. Circles around the points indicate that the participant rated the satisfaction level the same for each approach and the color indicates which approach was preferred (green for simultaneous, blue for parallel) ...................... 106

Figure 38. Histogram showing the number of participants who rated their satisfaction levels as (1) Very dissatisfied, (2) Dissatisfied, (3) Satisfied, or (4) Very Satisfied for the parallel and simultaneous approaches ......................... 121
LIST OF TABLES

Table 1. The four tasks that were given to the participants for the emergency response scenario, as well as the two different subsets of information that were provided for each. ................................................................. 96

Table 2. The four tasks that were given to the participants for the real estate shopping scenario, as well as the two different subsets of information that were provided for each. ................................................................. 97

Table 3. Follow-up survey. ......................................................................................................................... 99

Table 4. Frequency of satisfaction ratings for the two approaches.................................................... 105

Table 5. Number of participants, out of 9, who stated the given reasons for why they preferred the parallel approach over the simultaneous approach. Participants could provide more than one reason. .............................................................. 109

Table 6. Number of participants, out of 7, who stated the given reasons for why they preferred the simultaneous approach over the parallel approach. Participants could provide more than one reason. .............................................................. 109
ACKNOWLEDGEMENTS

I first of all want to thank Alan MacEachren for all of his guidance over the course of this research project. His vast expertise and his willingness to somehow find time to share it were invaluable. I also greatly appreciate the time and effort put in by Alex Klippel and Guoray Cai as well as the advice they provided. The Department of Geography at Penn State generously supported my research with the Ruby S. Miller Endowment for Geographic Excellence, which made it possible to compensate participants. Finally, I want to acknowledge my family for all of their support and especially my amazing wife Allison for keeping me going these past two years.
Chapter 1

INTRODUCTION

Geocollaboration is a field of research concerning ways to support “visually-enabled collaboration with geospatial information through geospatial technologies” (MacEachren & Brewer, 2004, p. 1). These technologies can serve to support various knowledge-building, planning, and decision-making tasks, such as education, regional planning, military command, and emergency management to name just a few. There are different temporal and spatial dimensions in which these types of activities can take place, as shown in Figure 1. Temporally, geocollaboration can occur at the same time or at different times. Spatially, geocollaboration can occur in the same place or in different places. The focus of this research is on same-time, different-place geocollaboration.

Figure 1. The four temporal and spatial categories of geocollaboration and an example application of each (MacEachren and Brewer, 2004).
There has been a great deal of research on how to support same-time, different-place geocollaboration exploring several important questions. Topics such as the types of tools that need to be available to the users (Armstrong, 1994; Jones, Copas, & Edmonds, 1997), how to control the access to these tools (Armstrong, 1994), how to enable communication between the users (including gestures) (Churcher & Churcher, 1999; Pichiliani & Hirata, 2006), how to accommodate multiple user roles and assist these users in establishing common ground (Convertino, Ganoe, Schafer, Yost, & Carroll, 2005; Convertino, Mentis, Rosson, Slavkovic, & Carroll, 2009), and how to support geovisualization in this type of system have all been investigated to some extent (MacEachren, Brewer, & Steiner, 2001; MacEachren & Brewer, 2004). One question that has not yet been explored is how to support collaborative data querying in a same-time, different-place geocollaboration environment.

Data have a very prominent role in any geocollaborative task. In geocollaboration the data will often be visualized as a map and will be a central part of the focus. It is therefore crucial to have an appropriate data set and to be able to work with the data. During the collaborative process, the need for new sets of data also often arises. To find and work with data, it is necessary to perform queries. It makes sense that software for facilitating geocollaboration should support collaborative data queries.

The goal of this research is to build on our understanding of how people collaborate with geospatial technologies by exploring methods to enable collaborative, structured queries of spatial data. Standard spatial data that would be used in a typical GIS are stored in a structured database of some sort. A common action performed by a GIS analyst is to query their spatial database based on selected criteria for attributes
and/or location of the data. For example, an analyst might retrieve all buildings that are zoned commercial, within a ten mile radius of a given point. In working towards supporting geocollaboration among teams of analysts, it is necessary to explore how these teams might collaboratively query their data.

Since structured data retrieval can be a complicated task for users and since it is such a common task in geographic research and decision making, supporting synchronous collaborative spatial data retrieval makes a lot of sense. Users with complementary expertise could assist each other in constructing the queries, both from a syntactical standpoint and from a role-based standpoint. For example, a domain expert who knows about a data set but not about constructing queries could work collaboratively with a GIS expert who knows how to put together spatial queries, but does not know about the data set.

The objectives of this research were twofold. First, I developed a conceptual framework for supporting collaborative spatial query and then built a software prototype based on this framework. This prototype acts as a proof-of-concept of my ideas (grounded in past research by others) for how these collaborative activities can be facilitated. The second main objective is to gain insight (through a focused user study) on how people prefer to carry out these collaborative spatial query tasks when comparing two distinct approaches, (1) simultaneous, collaborative query building (whereby users work together in real-time, with a single, shared query building interface) versus (2) parallel, collaborative query building (whereby users work within their own query building interfaces in parallel, and then can share and iterate over their individual results), along with insights on the strengths and weaknesses of each approach.
There is a substantial body of research upon which my own research is based, which spans various fields of geography and computer science. Computer supported cooperative work (CSCW) and human-computer interaction (HCI) studies are fundamental to the software design and evaluation aspects of my research. Research on collaborative information retrieval (CIR) is essential to understanding how synchronous collaborative data/information retrieval works and how to support it. Finally, research in the field of geocollaboration (in general, and more specifically geocollaboration for emergency management) is the most influential and relevant to my work. An overview of these research topics is provided in the next chapter.
Chapter 2

LITERATURE REVIEW AND BACKGROUND

The software I have developed as the focus of my research is the Spatial Queries in Synchronous Collaboration (SQSynC) software. The intent of this chapter is to give an overview of the research in the fields of Computer-Supported Cooperative Work (CSCW), Geocollaboration, and Collaborative Information Retrieval (CIR) which directly influence my own research on synchronous collaborative spatial information retrieval and its implementation in SQSynC. Section 2.1 provides a broad overview of CSCW and collaboration in general. Section 2.2 describes how CSCW has influenced geospatial technologies and tasks, resulting in the field of research on geocollaboration. Relevant research on CIR is discussed in section 2.3. Finally, the relevance of all this research and the potential application in emergency management is tied together in section 2.4.

2.1 Computer Supported Cooperative Work

CSCW is a field of research that developed in the 1980’s with the goal of discovering ways to support groups of people working together using computer technologies (Greif, 1988). It is important to distinguish among different levels of participation in this type of group work. A cooperative task is one whereby the
participants work independently on their own subtasks and share their results. A collaborative effort involves the participants working together on the same task or subtask (Jankowski, Nyerges, Smith, Moore, & Horvath, 1997). Balram and Dragićević (2006) agree that the distinction is subtle but important. Collaboration occurs with close integration, but with cooperation there is no such expectation. Jankowski and Nyerges (2001b) further break participation down into a cumulative hierarchy of four components. At the start is communication, followed by cooperation, then coordination, and finally collaboration. In discussing CIR, Shah (2008) presents a similar model where collaboration encompasses coordination, cooperation, and communication. It is clear that collaboration is the highest level of social interaction in this sense, and when filling the needs of geocollaboration and CIR, CSCW should probably be interpreted as computer-supported collaborative work.

CSCW provides the basis upon which geocollaboration and CIR rely heavily. CSCW technologies are often referred to as groupware, which can be defined as “computer-based systems that support groups of people engaged in a common task (or goal) and that provide an interface to a shared environment” (Ellis, Gibbs, & Rein, 1991, p. 40). It is groupware that supports geocollaboration and CIR tasks in general.

One way these computer-based groupware systems can be used to support geocollaboration and CIR is through the use of collaboratories (Kouzes, Myers, & Wulf, 1996). A collaboratory is a “center without walls, in which the nation’s researchers can perform their research without regard to geographical location - interacting with colleagues, accessing instrumentation, sharing data and computational resource, and accessing information in digital libraries” (Wulf, 1989 as cited in Cerf, et al., 1993, p. 7).
Geocollaboration software applications like SQSynC are necessary components of collaboratories that emphasize distributed use of geographic data. Interaction with colleagues is an obvious and necessary part of collaboration within a distributed collaboratory. Accessing instrumentation refers to the capability of researchers to remotely make use of tools necessary for their scientific research, such as microscopes. In the case of SQSynC, the tools are all integrated directly into the software, such as the map and query interfaces. Sharing data and accessing information is precisely the purpose of the collaborative data retrieval aspect of the system. Finally, SQSynC supports collaboration regardless of the geographical location of the researchers.

2.2 Geocollaboration

Geocollaboration is defined as “visually-enabled collaboration with geospatial information through geospatial technologies” (MacEachren & Brewer, 2004, p. 1). This describes an activity, but geocollaboration is also a field of research focusing on supporting that activity. There are several different perspectives within geocollaboration research, but there are no clear-cut categories for them. I have decided to organize these research perspectives into two broad categories: community-oriented geocollaboration (section 2.2.1) and team-oriented geocollaboration (section 2.2.2).
2.2.1 Community-oriented geocollaboration

Community-oriented geocollaboration systems are designed to support participation from a large-scale group of people. These systems are typically asynchronous for a couple of reasons. First of all, the number of people collaborating would make it logistically difficult to organize and support a synchronous meeting of everyone. Second, the purpose of community-oriented geocollaboration is generally a long-term planning or decision-making process. The most common term for describing community-oriented geocollaboration is public participation GIS (PPGIS). Other terms used include participatory GIS (PGIS) (Dunn, 2007; Carton & Thissen, 2009), collaborative planning system (CPS) (Shiffer, 1992; 1998), and planning support system (PSS) (Klosterman, 1999; 2005; Geertman & Stillwell, 2004; Jankowski, 2009; Kahila & Kyttä, 2009). Another prominent research perspective that applies mostly to community-oriented geocollaboration, but is separate from the primary perspectives of PPGIS research, is the topic of argumentation maps (Rinner, 2001). The following two subsections will present research on PPGIS and argumentation maps, respectively.

Public participation GIS

PPGIS began as a response to the criticism of traditional GIS as a top-down, non-participatory approach to planning and decision-making (Harris & Weiner, 1998) and the promise GIS holds as an agent for synthesizing expert knowledge with the experiences of the general public (Jankowski, 2009). Many researchers have seen the importance of including non-expert participants in planning and decision-making processes, such as the
dialogue on ‘GIS and society’ (Sheppard, 1995; Harris & Wiener, 1996) and the ‘rejection of GIS’ (Schlossberg & Mattia, 2003). After all, the professionals are not the only stakeholders in many geospatial plans and decisions. It is this need for these stakeholders’ voices to be heard in a democratic way that led to research in the area of PPGIS (Obermeyer, 1998).

A good overview of public participation in general is given by The International Association of Public Participation (2007) – IAP2. IAP2 describes five levels of public participation: inform, consult, involve, collaborate, and empower and in that order the level of impact a participant has on the overall process increases. Jankowski (2009) takes those levels and applies them to what he thinks the basic requirements for a PPGIS are. Maps, visualizations, and discussion forums might suffice in the lower levels, such as ‘inform’ and ‘consult’. In the higher levels, more analytical tools such as scenario evaluation and analysis utilities are required in addition to the visualization and communication tools.

One issue in PPGIS is the problem of scale. Everyone talking about PPGIS is quite clear about the need to bring in communities of people, but this necessitates a definition of the community (Dunn, 2007). The scale of the community to participate in the project must be considered carefully, to develop a clear understanding of the human-environment relationships relevant to the given goal (Stonich, 2002). Clearly, the system also must be scaled appropriately to be able to support the community once it is defined. The internet is certainly a possible option as the medium for such a system, as the scalability is essentially limitless and the accessibility to the public is high since there is no need for specialized software. Of course the downside is that an internet system
would not be accessible at all to people with no internet connection, putting low-income households at a disadvantage (Peng, 2001). Another possibility to help solve the potential cost-prohibitive nature of the internet is to set up publicly accessible GIS facilities in the community (Leitner, McMaster, Elwood, McMaster, & Sheppard, 2002).

There are certain features that are required for a system to successfully support public participation. To outline these requirements, Peng (2001) draws on some of the basic ideas from decision support system theories and argumentation map theories, both of which are discussed later in this chapter. He states that the functions necessary for the system are exploration (the participants must be able to explore information about the environment in consideration), evaluation (participants need a way to assess alternatives), scenario building (the capability for users to create their own scenarios is necessary), and forum (the abilities to discuss, express opinions of alternatives, and vote should be provided).

Peng (2001) developed a model for PPGIS modeled on a web-based public participation system (WPPS). Peng’s design is based on the three-tier architecture shown in Figure 2. The three tiers are client (the web browser), server (the web server), and application (the application servers). The GIS functionality resides on the servers in the application tier – this is where all the mapping and analysis is handled. The participants interact with the client tier by using their web browser to view the application interface. The server tier facilitates the connection between the participants and the GIS. With this architecture it is possible to create a PPGIS that is powerful, yet accessible and user-friendly.
One early implementation of a web-based PPGIS is Virtual Slaithwaite (Carver, Evans, Kingston, & Turton, 2001). Virtual Slaithwaite is a web-based system developed in Java, so anyone with an internet connection and a web browser with Java installed can access the tool and participate in the discussion. An updated version of the interface is shown in Figure 3 and illustrates how simple the system is. There is a map display with which the participants can zoom, pan, and select features. Once a feature has been selected, the participant can submit a comment about it. This tool provides the entire community with a simple and straightforward way to voice their opinions on features in their village, which would theoretically be taken into account in future planning processes.

The Virtual Slaithwaite system is in fact a digital version of a preexisting physical geocollaboration environment. As part of a Neighbourhood Initiatives Foundation (NIF) project for the area a three-dimensional model of the community was created. During a village fair, this model as well as eight computers running Virtual Slaithwaite were made available for public participation. Like in Virtual Slaithwaite, comments could be made
by the participants on the physical model, but this was done by writing the comment on a flag and appending that flag to the location. What Carver and colleagues found was that the biggest limitation to Virtual Slaithwaite was simply a lack of general knowledge about computer technologies. A benefit to Virtual Slaithwaite was that comments were not limited to just a couple of lines like they were on the physical model (Carver, Evans, Kingston, & Turton, 2001).

The Woodland Online Decision System (WOODS) is another web-based PPGIS with the intention of being easy to use. This ease-of-use is important for getting the
public involved. WOODS differs from Virtual Slaithwaite in the way users interact with it. It is set up more like a survey than like a traditional GIS. The participant is presented with a number of slides, each with a map of the study area depicting a different data set. The participant is asked whether trees should or should not be planted in the areas designated on the map, such as environmentally sensitive areas as shown in Figure 4. After making their selection for each of the maps presented to them, the final page displays a map of where the trees would be planted based on all of the input from the

*Figure 4. Woodland Online Decision System (WOODS) (source: http://www.ccg.leeds.ac.uk/dales/)*
participant. The participant can make changes to his or her final map before submitting it. This is another good example methodology for getting a community involved in a decision process that affects them (Woodland Online Decision System, n.d.).

**Argumentation maps**

Argumentation is a fundamental component of collaboration. In the collaboration literature, it is considered to be a process that involves participants sharing, discussing and debating their different views on the situation in question (Rinner, 2006). It is for this reason that argumentation maps were developed by Rinner (1999) to support geographically referenced discussion. His work is focused on a different-place, different-time setting. Since argumentation is inherently between more than one person, argumentation maps are inherently geocollaborative.

Debates can be formalized using argumentation models. Different argumentation models can be developed to fit a specific purpose. The simplest example of an argumentation model is a question/answer model, where a contribution is either a question (beginning a new discussion) or a response to either the question or to an answer to the question (Rinner, 2001). The argumentation model can be determined on a case by case basis.

Rinner (2001) presents four argumentation map use cases, which each correspond to a typical GIS use case. The four use cases are navigation, participation, exploration, and evaluation. The typical GIS functions that these use cases correspond to are
presentation, input, retrieval and analysis respectively. In navigation, a user browses a map that has preexisting argumentation contributions on it. These contributions are represented on a map by a symbol, and various symbols can be used to represent different types of contributions (e.g. a question mark for a question and an exclamation mark for a reply). By clicking on the symbol, the user is presented with the contents of the argumentation element. In the participation use case, the user is contributing a geographically referenced discussion element to the map. The exploration use case involves geovisualization of the existing argumentation elements on the map, with the ability to analyze different thematic maps displaying aggregated data about contributions. Finally, evaluation is a combination of argumentation analysis and spatial analysis to assess and/or modify the results.

Argumentation mapping is presented by Rinner (2001) as an object-based model for geographically referenced discussions. The model in Figure 5 shows the argumentation component on one end and the geographic component on the other. Argumentation components are logically related to each other. These relationships can be complex as they are based on hierarchy and argumentation type. The geographic components are spatially related and represent the footprint of the argumentation element. An argumentation element can be tied to either a preexisting geographic object, or an arbitrary graphic reference placed on the map by the user.
Argumentation mapping is an important contribution to geocollaboration theory. It enables place-based group planning, decision making, and knowledge building. Using argumentation maps, groups of people can work together toward developing knowledge and potentially a consensus (MacEachren, Gahegan, & Pike, 2004). Argumentation mapping systems are capable of supporting professionals as well as public stakeholders and argumentation plays a key role in the early stages of geocollaborative tasks.

One of the earliest argumentation map prototype applications was created by Keßler (2004). The argumentation model used for the prototype allows users to mark their contributions as question, suggestion, pro, con or neutral (as opposed to the simple question/answer model presented above). As shown in Figure 6, discussions are presented in a nested-tree format, with an icon to the left of each entry indicating its type (question, suggestion, pro, con or neutral). Geographically referenced objects on the map that have corresponding discussion components can be clicked to display the discussion, enabling interactive exploration. This tool is essentially a different-place, different-time
geocollaborative environment to enable participants to create annotations on a map to facilitate geographic discussion (Keßler, Rinner, & Raubal, 2005).

**Figure 6.** Keßler’s argumentation map prototype interface (Keßler et al., 2005).

Yu and Cai (2009) also present some research that fits within the category of argumentation mapping. They start out by presenting nine requirements that they have determined are necessary for a successful system: (1) the ability to create customized maps using various, heterogeneous data sources, (2) support for multi-modal communication options such as text, drawing and audio, (3) the ability to link arguments with geographic references, (4) support for structured arguments, (5) the ability to capture the context of arguments, (6) support for queries based on geographic scope as well as argument context, (7) sophisticated visual analysis, such as animation of a discussion
over time, (8) easy access and usability, and (9) the ability to control the degree to which a contribution is shared with other users.

Cai and Yu (2009) also discuss a software system they developed to meet these nine requirements called GeoDeliberator (Figure 7). The main functions of GeoDeliberator include the ability to manage discussion groups, the ability to customize the map for the discussion, the ability to look through and search for annotations, and the abilities to create new annotations and edit existing ones. The final main function of the

Figure 7. GeoDeliberator. (1) User and Group Panel, (2) Project Panel, (3) Map Panel, (4) Timeline View, and (5) Annotation Info Panel (Cai & Yu, 2009)
system is providing options to analyze annotations. The annotations can be viewed in a
tree-based layout to determine which annotations refer to which other annotations and
they can also be browsed using a timeline view. Another supported way to analyze the
annotations is by viewing their geographic references on a map. An aggregate view is
provided using graduated circles to make it easy to determine which locations are
discussed most.

2.2.2 Team-oriented geocollaboration

In contrast to community-oriented geocollaboration, team-oriented
geocollaboration is a category of research focusing on how small-scale teams of people
 collaborate with geospatial information and technologies. These teams are generally
experts who are working together to conduct research or to make a decision that is spatial
in nature and thus they would benefit from geocollaboration technologies. This type of
collaboration is also typically synchronous as opposed to asynchronous. One term
commonly used to describe these types of technologies is group spatial decision support
systems (GSDSS). Other terms are sometimes used to describe team-oriented
geocollaboration systems, such as collaborative visualization or collaborative spatial
decision making (CSDM) systems, which has the added confusion of sharing an acronym
with computer-supported decision-making (Karacapilidis, Papadas, & Egenhofer, 1995;
Armstrong & Densham, 1995; Brewer, MacEachren, Abdo, Gundrum, & Otto, 2000;
MacEachren, Cai, Brewer, & Chen, 2006). Semantics aside, what links these team-orientated systems is the intention to support small-scale teams of domain experts in
geocollaboration tasks. Much of this research is based on adapting decision support systems (DSS) to support spatial tasks.

DSS technology was initially developed to improve the decision process overall (Sprague & Watson, 1986; Benbasat & Nault, 1990). DSS technology and a need for software to support spatial decision-making more effectively than existing GIS created an interest in researching and developing spatial decision support systems (SDSS) (Densham, 1991; Nyerges, Montejano, Oshiro, & Dadswell, 1997). It was the idea of combining SDSS technology with CSCW technology that really sparked a research interest on team-oriented geocollaboration (Armstrong, 1993).

Armstrong (1993) lays out three stages that decision-makers pass through in the iterative process of seeking a solution to a problem. The stages are strategizing, exploration, and convergence, in that order. Strategizing involves utilizing background knowledge of the study area to formulate an initial solution process. In exploration, the decision-makers look at alternative solutions, which can be variations of the initial solution from the first stage, or a different potential solution altogether. Finally, in the convergence stage the focus of the analysis is narrowed and the assorted solutions are evaluated. It was Armstrong’s goal to update the SDSS model to provide guidance for exploring GSDSS applications.

The need for GSDSS can be attributed in part to the GIS bottleneck (see Figure 8). Spatial problems often require the efforts of multiple decision-makers in order to be solved. Prior to GSDSS, GIS was not capable of supporting this type of group process. GIS could only support a single user (the GIS Analyst in Figure 8), who had to act as an intermediary between the geospatial data and the decision makers. It is easy to see, then,
how this would severely impede progress. The need to overcome this bottleneck, coupled with the rapidly increasing capabilities of computer hardware and software, opened the door for GSDSS research (Armstrong, 1994). The focus of this thesis on supporting collaborative spatial query is targeted at this bottleneck problem as well.

![Image](image_url)

*Figure 8. “The GIS bottleneck” (Armstrong, 1994).*

There are several different design theories for what a team-oriented geocollaboration system should be. The following three subsections discuss (1) general GSDSS design theories and examples, (2) visualization theories for team-oriented geocollaboration, and (3) the requirements and some examples of same-time, different-place geocollaboration systems, respectively. There is a great deal of overlap, especially between the latter two topics, since geocollaboration relies so heavily on visualization.
Group spatial decision support design theories and examples

There are many concepts that various researchers have discussed as requirements for building a useful GSDSS. Armstrong (1993) suggests that the capability of generating alternative scenarios and presenting them to the rest of the group should be provided and that trace and reset functions should be incorporated into the system, such that decision-makers can backtrack through the steps of the processes without needing to start from the beginning again. A way of evaluating alternative scenarios should also be supported. Assistance in resolving conflicts and reaching decisions is another important feature, which can potentially be supported with voting capabilities (Armstrong, 1993; Karacapilidis, Papadias, & Egenhofer, 1995; Jankowski, Nyerges, Smith, Moore, & Horvath, 1997). Karacapilidis et al. (1995) also state that the ability to elicit and capture data is important. These are fundamental design requirements for GSDSS.

Jankowski and Nyerges (2001a) provide a much more comprehensive and complex theoretical framework for empirical GSDSS research, which they call EAST2. EAST2 is a revised version of their Enhanced Adaptive Structuration Theory (EAST) (Nyerges & Jankowski, 1997; Nyerges, 1999), which is in turn an enhancement to Adaptive Structuration Theory (AST) (DeSanctis & Poole, 1994) to accommodate collaborative GIS tasks. The EAST2 framework (shown in Figure 9) consists of eight constructs categorized as convening constructs, process constructs, and outcome constructs. The seven premises of EAST2 (labeled as P1 through P7 in Figure 9) describe the relationships between the constructs. Finally, there are 25 aspects within the
Figure 9. EAST-2 framework. Adapted from (Jankowski & Nyerges, 2003) to include labels for the seven premises.

constructs, which are organized in a way that reflects an organization of a group decision making process. In essence, EAST2 “suggests that people structure situations; in turn,
situations structure people’s interaction” (Jankowski & Nyerges, 2003, p. 16). One problem with EAST2 is its sheer size and complexity. It is the culmination of numerous studies, articles and books by Nyerges and Jankowski. The outline they created (Figure 9) is the easiest way to summarize their framework in its entirety in an economical manner (see Jankowski & Nyerges, 2001a for more on EAST2).

One of the earliest implementations of a GSDSS is called Spatial Group Choice and was developed by Jankowski et al. (1997). Their goal in designing this system was to enable exploration of a problem, sharing of perspectives for decision criteria, evaluation of possible solutions, and negotiation of the agreed upon solution. The system is an experimental prototype which is essentially the combination of a multi-criteria decision-making component called Group Choice and a GIS called ArcView 2 (see Figure 10). The basic process is that users view the spatial data using the ArcView 2.
component. They then vote on their decision criteria of choice using a private mode of the application. Their opinion is then submitted and this can be either open or anonymous. In the public mode of the application, submitted opinions can be explored, evaluated and analyzed. This functionality is basic, mostly due to technological limitations of the time.

There have been other implementations of geocollaborative systems that could be considered GSDSS as well, but a more complete context within which to consider the key aspects of these systems is required. The next two subsections outline the main visualization concepts that need to be considered, and the requirements of synchronous systems. More team-oriented geocollaboration systems are presented in each of these subsections.

**Visualization concepts**

Visualization concepts, such as techniques for creating visual displays and information about how people use and interpret visual displays, are very important to the field of geocollaboration, and are potentially applicable in all types of visual geocollaborative interfaces. The significance of visual displays to geocollaboration is primarily as the mediator for collaborative work. Much of the research is grounded in geovisualization as a more general field of research (MacEachren, Brewer, & Steiner, 2001; Dykes, MacEachren, & Kraak, 2005).
One theoretical perspective that has been adopted to support research in geovisualization is semiotics, the ‘science of sign systems’. A common way to conceptualize a sign is with a semiotic triangle, which is made up of the referent (what is being represented), the sign-vehicle (the symbol being used as the representation), and the interpretant (the meaning of the referent as derived from the sign-vehicle) (Pierce, 1955; MacEachren, 1995). The reason this is important from a geocollaborative perspective is because the interpretant is potentially different for every user of the system (Figure 11). Therefore, a geocollaborative system should support methods for the users to discuss their interpretations of sign-vehicles (maps and map objects in this case) as well as visual tools to enable them to work toward a shared understanding (MacEachren & Brewer, 2004).

Figure 11: Semiotic triangles for collaborative visualization (MacEachren & Brewer, 2004)
DiBiase’s (1990) conceptual model is a good starting point for understanding the processes of visual thinking, communication, and decision-making enabled through visual displays. DiBiase describes the process of going from visual thinking, in the private realm, to visual communication, in the public realm. The steps of this process are to explore the problem and confirm relationships in the private realm and then to synthesize and present the results in the public realm. Armstrong and Densham (2008) extend DiBiase’s model to add a consensus building part of the process following the visual communication (Figure 12). They call this the evaluative realm and this is the collaborative part of the process where team members re-confirm and summarize the findings to reach a consensus.

Figure 12: An updated version of DiBiase’s conceptual model for visual thinking and communication, extending it to consensus building in an evaluative realm (Armstrong & Densham, 2008)
MacEachren and Brewer (2004, p. 6) create a conceptual framework for supporting visually-enabled geocollaboration with the intention of providing “a basis from which to design, implement, and understand the use of collaborative geospatial information technologies, with particular attention to dynamic visual displays as mediators for group work.” Visual displays as mediators for geocollaboration are shared representations of geospatial information. This shared display can be used as context for discussion, as support for arguments or as a device for comparing perspectives. MacEachren and Brewer’s framework consists of six dimensions: problem context, collaboration tasks, perspective commonality, spatial and temporal context, interaction characteristics, and tools to mediate group work.

MacEachren and Brewer (2004) also look at several different interface options for visual display. The spatial level of collaboration (same-place or different-place) is a major determinant of the type of interface used. In same-place collaboration there should typically be one shared display, which is a hardware requirement. The display must be large enough to support all the participants. For different-place collaboration, each participant will usually have their own private display and the interface is typically a software application. Of course, there is no reason that large-screen displays cannot be used for different-place collaboration as well, particularly between groups of people rather than individuals.

One example of an early implementation of a large-screen interface to facilitate geocollaboration is a system for collaboratively exploring space-time climate data using an ImmersaDesk environment (Figure 13) (MacEachren & Brewer, 2004). An
ImmersaDesk is a large display that provides participants with a semi-immersive virtual reality environment (Otto & Thrasher, n.d.). Participants wear special goggles and the display appears to be three dimensional. Interaction with the display requires the use of a wand, which plugs into the goggles. As Figure 13 depicts, the ImmersaDesk supports up to five people positioned close enough to point at the display (in this case to a map). The wand also includes a laser pointer for the guide to point at specific locations on the map. While the ImmersaDesk display enables a group of people to perform same-time, same-place geocollaboration, it reduces some of the advantages of a same-place environment. The participants are all forced to wear goggles, which hinders their natural gestures. The wand also makes it cumbersome to hand off control to another person, as the two people would also be required to switch goggles (MacEachren & Brewer, 2004). The ImmersaDesk is not the ideal type of display for geocollaboration.

Another large-screen interface is a device called Human Information Workspace (HI-SPACE), which is an early implementation of ideas comparable to current multi-touch display surface technology (Figure 14). “HI-SPACE is a large-screen, rear projection, table display environment that supports hands-free, multi-user, untethered interaction with an electronic information space” (MacEachren & Brewer, 2004, p. 5). The purpose of HI-SPACE is to facilitate a more natural and productive collaboration environment for the participants. The idea is to participate in a round-table setting as opposed to being dispersed throughout a room or huddled against a wall. The untethered interaction enables the participants to control the interface with hand gestures without the requirement of a special glove. Finally, the system supports what are called ‘phicons,’ which means participants can place an ordinary, real-world object on the display and the
system would recognize it as a feature or placeholder (MacEachren, Brewer, Cai, & Chen, 2003).

Figure 13. ImmersaDesk (MacEachren & Brewer, 2004).

Figure 14. HI-SPACE (MacEachren & Brewer, 2004).

Large screen displays are continuing to evolve, and research on how to support collaboration and information visualization is still ongoing. Tang et al. (2006) take a look at the problem of supporting both independent and group work, specifically when
using a large screen, tabletop display. They compare three different techniques for organizing the viewing space of the display: “(1) lenses, which show information in spatially localized areas, (2) filters, which show information globally, and (3) ShadowBoxes, which allow spatially localized areas to be displaced” (Tang, Tory, Po, Neumann, & Carpendale, 2006, p. 1181). Through a user study, what they essentially found is that supporting collaboration is a complex problem because of numerous variations in collaborative work techniques of people. What they suggest is to provide flexible interfaces and support fluid transitions between different workspace views like the three listed above.

A prototype visual display environment for a typical personal computer (PC) display was created in the GeoVISTA Center at Penn State to support same-time, different-place geocollaboration, with the purpose of looking at several context questions, such as ‘who is interacting with what,’ ‘where is that person looking,’ and ‘what is being changed and how’ (MacEachren & Brewer, 2004). The two lower right panels in Figure 15 represent each participant and what they are doing. Everyone can tell who is currently in control and which part of the interface that person is interacting with. Also displayed is the proportion of time each person has been in control. Mouse movements and clicks are also tracked and the movements can be replayed. The workspace and interaction concepts implemented in this prototype are also foci of research presented in the next section, focusing specifically on same-time, different-place geocollaboration on a PC.
Same-time, different-place design theories and examples

There are several considerations to take into account when developing systems specifically for same-time, different-place geocollaboration. To start with, Pichiliani and Hirata (2006) outline what they consider the minimal requirements of a same-time, different-place collaborative system. First of all, the system must support communication. It can be in the form of text, audio, or video, but the team members need a way to exchange ideas with each other. In an interview of geographers, Brewer et al.
(2000) found that users prefer audio communication in a same-time, different-place geocollaboration environment; however, after 10 years of changing technology it is possible that these results might be different if a similar study were carried out today.

The second requirement listed by Pichiliani and Hirata (2006) is support for workspace awareness. Churcher and Churcher (1999) discuss the need to support natural gestures that users would typically make when collaborating in person, such as pointing, drawing, and other more subtle body and facial gestures. Heiser, Tversky and Silverman (2004) confirmed the usefulness of these types of gestures in an experiment on co-located geocollaboration, where teams collaborating on a shared map using gestures performed better than teams collaborating on individual maps without the ability to gesture to each other. Brewer et al. (2000) also found that many of the respondents to their interview stressed the importance of gestures and the ability to draw others’ attention. Churcher and Churcher (1999) suggest that knowing who is actively participating, what they are doing, and what their current area of interest happens to be is important. Each user needs to know not only the output of any action by another user, but also what that specific action is (Jones, Copas, & Edmonds, 1997; MacEachren, et al., 1999). One goal with same-time, different-place geocollaboration is to create an environment as close to a co-located collaborative environment as possible.

Another feature to help with workspace awareness is the use of what are often referred to as whiteboard controls (Armstrong, 1993; Carroll, et al., 2001; Heiser, Tversky, & Silverman, 2004). Whiteboard controls are free-form drawing tools that team members can use to draw in the interface and communicate with other team members. These tools are especially useful when combined with other visualizations, such as maps.
It is easy to imagine, for example, a team using the whiteboard features of a collaboratory to draw a route on a map in order to quickly and easily communicate that route to the rest of the team.

Providing what you see is what I see (WYSIWIS) functionality is another way to help support workspace awareness, and is often discussed as an essential feature of same-time, different-place collaboration software (Armstrong, 1993; Jones, Copas, & Edmonds, 1997; Churcher & Churcher, 1999; Gutwin & Greenberg, 1999; Park, Kapoor, & Leigh, 2000; Carroll, et al., 2001; Stefik, Foster, Bobrow, Kahn, Lanning, & Suchman, 1987). This concept means just what it sounds like; each person using the system sees precisely what every other person using the system currently sees. This can be true of the entire interface, or just for a portion of the interface. By seeing what each user is doing within the system, the whole team is able to be more aware of what is happening within the workspace.

In addition to a public WYSIWIS view users should also be given a private workspace view (Armstrong, 1993; Churcher & Churcher, 1999; Brodlie, Duce, Gallop, Walton, & Wood, 2004). Some users will feel more comfortable first working within a private view and then sharing results in a public view (Greenberg, Boyle, & Laberge, 1999). A private workspace allows users to experiment and make mistakes without worrying about their collaborators watching them. Convertino et al. (2005) take the idea a step further and suggest that each user should be given a private, role-specific view of the system that is tailored specifically to their needs based on their expertise.

The third main requirement in same-time, different-place geocollaboration is to implement appropriate concurrency control, which determines who is able to do what at
any given time (Ellis, Gibbs, & Rein, 1991; Jankowski, Nyerges, Smith, Moore, & Horvath, 1997; Jones, Copas, & Edmonds, 1997; Wood, Wright, & Brodlie, 1997; Guerrero & Fuller, 1999; Chang & Li, 2008). There are different ways to implement concurrency control. The first option is to basically have no concurrency control, which would mean anyone can perform any operations supported by the system at any time. This would likely become chaotic pretty quickly and there would be problems with conflicting interactions. For instance, if two people attempt to change a data value at the same time it would be problematic. The other end of the spectrum is to have a single facilitator method of floor control, whereby only one person has full control of the system at any one time (Jankowski, Nyerges, Smith, Moore, & Horvath, 1997). This can be a single person for the duration, or control can be passed off to other team members in a round robin fashion (Jones, Copas, & Edmonds, 1997). Allowing only a single person to interact with the interface at any given time is not a far step away from the GIS bottleneck described previously. A third option is to have a tiered method of concurrency control, where different users have different levels of control within the system. The most suitable concurrency control option can be dependent on the type of task and the team using the system, but it is an important consideration to make when designing any sort of synchronous collaboration environment.

Convertino and colleagues (2005) implemented a same-time, different-place geocollaboration prototype that implements a private, role-specific view for each user and a public, shared view for all the users (Figure 16). The idea of the role-specific view is to provide visualizations and tools tailored to the expertise and/or objectives of the specific user so that only knowledge that is relevant to the whole team is shared to all of the users.
This is well suited to interdisciplinary work, where experts from various fields can collaborate without bogging each other down with specialized knowledge that perhaps would not be well-understood by or of interest to everyone. The system supports user annotations in the role-specific map, which can then be selectively transferred onto the public view. This system is an excellent example of a mixture of WYSIWIS and a private workspace.

In geo-Collaboration through Information VI/ualization (CIVIL), Wu et al. (2009) built upon the previous work described above in a prototype designed to support annotations and sketches in order to refer to and communicate about objects on the map. They also mention support for information aggregation, but do not go into detail about this feature. To evaluate the system the authors conducted a lab study where 12 participants used CIVIL and responded to a questionnaire; overall the usability of the system was found to be satisfactory. An interesting note is that no concurrency control was instituted, which was pointed out as problematic by the participants.

*Figure 16. Multiple-user, multiple view geocollaborative system (Convertino, Ganoe, Schafer, Yost, & Carroll, 2005).*
Research and development in geocollaboration, while continuing, has progressed beyond laboratory prototypes to some commercial systems; Toucan Navigate is among the first. It is built on top of a preexisting CSCW environment, Groove. As such, it inherits the collaborative capabilities of Groove to provide many of the necessary geocollaboration utilities. Also, since Groove is driving the architecture that actually shares the data between users, Toucan Navigate can only be used as a peer-to-peer (P2P) system. In other words, there is no central architecture running the system; instead everything is run on the client machines of each user. This limits the capability of the system to supporting synchronous collaboration only, since at least two users must be online simultaneously in order to share their work.

A good overview of the Toucan Navigate system is given by Shafer et al. (2005) and they discuss it in the context of ‘who’, ‘what’, ‘where’, ‘when’, and ‘how’. Toucan Navigate gives information about the users, such as who is currently online and who is offline, who is using what tool, as well as identifying information about each user, similar to the ideas introduced in the prototype interface developed in the GeoVISTA Center at Penn State discussed previously (MacEachren & Brewer, 2004). For the ‘what’ question, Toucan Navigate provides alerts when something new has been done in the system and supports quick navigation to the change that was made. Toucan Navigate also provides the capability of sharing one’s location to the group using a GPS, an address, or by simply pointing to a location on the map. A history of events is also kept within the system, such that users can step backward or forward through alerts. Finally, the question of ‘how’ is addressed by Toucan Navigate in the availability of discussion tools.
Toucan Navigate stands as a good demonstration of how these same-time, different-place visualization concepts can be implemented into a geocollaborative system.

One concept that seems to be lacking from geocollaboration literature is that of collaborative spatial data retrieval. The actual geospatial data are clearly important parts of any geocollaboration task. Most of the research discusses ways to create and analyze data, but there are no studies on how teams can collaboratively retrieve a desired subset of data. It is possible that collaborative data queries are not the best way to retrieve spatial data within a collaborative tool and that queries should be performed individually in private instead, as suggested by Churcher and Churcher (1999). However, there is no evidence about the relative merits of this approach and alternatives that are collaborative and it is an important question that should be further explored. Fortunately, some computer scientists have researched the problem of collaborative information retrieval in general, which provides a good place to start. This research is discussed in section 2.3.

### 2.3 Collaborative Information Retrieval

Collaborative information retrieval (CIR) has been a focus of research within the field of computer science for at least the past decade and it is an important problem when dealing with collaboration systems. The central focus of collaboration involves data and information. A big part of many collaborative tasks is analyzing data to come up with useful information. Much of the research discussed so far has been about supporting these types of tasks through various analysis and visualization techniques. However, in real-world scenarios, researchers are often going to be interested in a subset of all the data.
or information they actually have. This is where information retrieval (IR) comes into play.

IR is a field of research dating back to the 1950’s (Singhal, 2001). The main objective of IR research has been the improvement of capabilities to find useful information from large collections of information. A majority of the research has been focused on the more complex problem of retrieving unstructured documents as opposed to retrieving structured data. From a technical standpoint it is more difficult to develop a system capable of effectively and efficiently retrieving documents, but from a user standpoint it is more difficult to come up with appropriate Boolean queries to retrieve structured data (Blair, 1984; Singhal, 2001). There is also a similar field of research that goes beyond just information retrieval, and looks at ways to support information seeking. Information seeking is a common process when the information needs of the user are not well-defined enough to be satisfied by information retrieval alone; support is needed to help analyze and manage the retrieved information (Marchionini & White, 2009).

There is a growing body of literature on CIR that has been useful in my own research, but there are some important distinctions to make. First, some research on CIR is not actually about how people explicitly collaborate with each other, but rather how logs of past queries can be used to improve a current query by an individual, which is more like human-computer interaction than human-human interaction (Fu, Goh, Foo, & Supangat, 2004; Khoussainova, Balazinska, Gatterbauer, Kwon, &Suciu, 2009). The focus of my research is human-human interaction. Another important distinction is the data-document distinction, which is discussed above (Blair, 1984; 2006). Most CIR research is focused on document retrieval, while my research is focused on data retrieval.
However, some points from research on CIR for document retrieval are still useful for collaborative data retrieval as well.

Golovchinsky, Qvarfordt and Pickens (2009), in acknowledging the potential for semantic confusion over the word *collaboration* in CIR literature, decided to classify CIR systems based on four dimensions, which are intent, depth of mediation, concurrency and location. The intent of the collaboration is either explicit or implicit. In explicit CIR, two or more people are intentionally working together toward a shared goal. With implicit CIR, the system might automatically assist an individual user based on previous, similar queries. The depth of mediation is the extent to which the system mediates the query and can either be at the interface level or at the database backend level. Explicit CIR is only mediated by the interface and once the users have constructed and submitted their query, the system does nothing but process the single query. In implicit CIR the mediation is done on the backend of the system, where other, previous queries and filters are automatically taken into account along with the single submitted query. The third and fourth dimensions are concurrency and location and these are the same as the spatial-temporal categories of collaboration previously discussed. These four dimensions of classification for CIR really help to put CIR research into perspective. The CIR aspects of my research fit within the explicit, interface mediated, synchronous, distributed dimensions.

One way of supporting synchronous CIR (SCIR) is by using a division of labor approach (Morris & Horvitz, 2007; Foley & Smeaton, 2009). In a division-of-labor SCIR, such as SearchTogether, a query can be issued by someone within the system and then the results of that query are equally divided among all the users currently logged in
(Morris & Horvitz, 2007). In this way the users can work together on exploring the results while not needlessly duplicating their efforts. The results can be rated by the users to distinguish which results are good and which are not. This type of system makes sense for a document retrieval task or fuzzy data queries, but it is not the best way to go about retrieving a specific data set.

When looking for a specific set of data based on given criteria, the Boolean model of IR is often an appropriate one to use. Using the logical operators AND, NOT, and OR, Boolean queries return a set of results that are an exact match for the query (Foley, 2008). An existing CIR system that uses this model is TeamSearch (Morris, Paepcke, & Winograd, 2006). TeamSearch is a same-time, same-place CIR interface using a large tabletop display. Each team member sits around the table and they have a virtual pile of what are called query tokens. Using the query tokens, the users can designate search criteria to be used as the query to retrieve photographs. The system retrieves all photographs whose attributes match the given criteria. TeamSearch supports two modes of collaborative queries: (1) collective querying, where the results must match all of the query tokens from all of the users, and (2) parallel querying, where each user receives a result set based only on their own query tokens. This is a simplistic version of a Boolean model and the main goal with this research was to explore a novel method of visual query in a collaborative environment.

While the Boolean model of query is widely used for retrieving data from a structured database, it is not without criticism. Several researchers argue that Boolean queries are not sufficient for all spatial data query tasks, especially ones that involve uncertainty. Often, decisions need to be made that involve data for locations with
unspecified or imprecise boundaries (e.g. English speaking areas). These fuzzy queries involve data sets with gray areas, as opposed to the strictly black and white approach of the Boolean model. While fuzzy logic may more accurately account for uncertainty in data sets than Boolean logic, it does require more specialized software than a typical relational database (Schneider, 2001; Morris & Jankowski, 2001; Robinson V. B., 2003).

There is still a lot to learn about CIR, especially when dealing with structured, spatial data. It is a piece of the puzzle that is still missing from geocollaboration research. It is, however, a very important piece since deriving meaningful information out of raw spatial data is such a crucial part of most geocollaboration applications.

2.4 Geocollaboration For Emergency Management

There are numerous potential applications for geocollaboration. For example, shopping for real estate, site selection for new development, military strategic planning, public policy problems, and education could all potentially benefit from geocollaboration technologies. One specific application where geocollaboration methods and tools have a potential to make a positive impact and which is part of the focus of this research is emergency management and response.

Emergency management is a very broad topic that can include various activities. Emergencies, as recognized by Federal Emergency Management Agency (FEMA) (2009), can include severe weather conditions such as hurricanes, tornados and winter storms, as well as human-induced emergencies such as chemical spills, nuclear power plant emergencies and terrorist attacks. The actual management activities can be long-
term planning prior to an emergency event or time-critical response to a recent or ongoing emergency event (Federal Emergency Management Agency, 2008). Whether it is the planning stage or the response stage, geospatial information and technologies can be extremely helpful in the process of dealing with emergencies.

Not only is emergency management an inherently spatial task, it is also typically collaborative in nature. Teams of people often work together to plan for or respond to an emergency situation. This is confirmed in FEMA’s documentation for emergency management, which includes activities such as alert and notification; deployment of response teams; incident action planning; operation coordination; direction and control; information collection, analysis and management; facilities management and several other tasks that require collaboration and would benefit from geospatial information and technology (Federal Emergency Management Agency, 2008). These types of collaborative efforts, which are spatial in nature, demonstrate how emergency management teams clearly stand to benefit from geocollaboration applications.

There are multiple requirements to consider when developing technologies for supporting emergency management tasks. One key requirement is that the system should be interoperable with preexisting infrastructure and compliant with existing legislation. It also needs to be sufficiently flexible to be scaled to all levels of emergency management, from local to international. In addition, communication must be supported by the system and will ideally be multimodal. The system should use common terminology to ensure interoperability and avoid confusion. It also needs to provide decision support tools and information sharing. Finally, the system needs to be sufficiently secure and provide
In addition to the fundamental requirements of a system for supporting emergency management in general, there are additional functions that need to be supported for time-critical emergency response. Cai (2005) argues that access to spatial data, support for effective information retrieval, and support for team work are essential. MacEachren et al. (2005) conducted a series of phone interviews and questionnaires in order to determine which features are essential for a GIS in supporting emergency response and found that “participants identified support for zoom, pan, buffer, display, and selection of geospatial data as the core functions that a GIS-based emergency response environment should have” (MacEachren, et al., 2005, p. 307). There are also demands of immediacy and relevancy on the system, meaning it needs to work in a timely manner and provide information that appropriately fits the needs of the team (Cai, Sharma, MacEachren, & Brewer, 2006).

When meeting in person to collaborate on emergency management tasks, traditional tools are not particularly ideal. Paper maps and paper annotations, while reasonably capable of supporting input from multiple people, are inflexible and limited in scope (Convertino, Mentis, Rosson, Slavkovic, & Carroll, 2009). A traditional GIS provides more powerful tools to accomplish the task, but does not support collaboration. For these reasons, there has been research focusing specifically on co-located geocollaboration for emergency management.

Rauschert et al. (2002) suggest that a same-place geocollaboration system to facilitate emergency management should be multimodal as opposed to a traditional
keyboard and mouse interface, or only one modality, such as speech. To support this multimodal interaction, they developed a Dialogue Assisted Visual Environment for Geoinformation (DAVE_G). The two modes of interaction supported by DAVE_G are speech and natural gestures. DAVE_G has a large screen display for presenting the information to the users, as well as audio and video inputs for capturing speech and gestures. It can interpret user inputs in order to perform data queries, buffer creation, zooming and panning of the map, and drawing (Fuhrmann, MacEachren, & Cai, 2008). The dialogue-based data query functionality in DAVE_G is a type of CIR based on human-computer interaction, but is also similar to human-human interaction, since the computer acts as an agent. This query functionality is also more similar to my research than most other CIR in that it supports data retrieval as opposed to document retrieval. A nice example of how DAVE_G would be used is illustrated in Figure 17.

In team work for emergency management tasks there is not always the convenience of having each team member in the same location. This is especially true of time-critical emergency response scenarios where there simply is not enough time for everyone to meet in a single location, and certain team members are out in the field. In these situations, a system capable of supporting same-time, different-place geocollaboration is essential. Several of the previously discussed same-time, different-place prototypes could potentially fit this need.
The research presented in the literature discussed in this chapter provides an excellent foundation to begin my own research on SQSynC. Theories from CSCW, geocollaboration and CIR research are taken into account in the design for SQSynC. Workspace awareness, WYSWIS features, public and private workspaces, communication support, and geovisualization capabilities have all been shown to be essential for a successful same-time, different-place geocollaboration environment. It is also clear that spatial data query is an important task in geocollaboration, but there are still questions to be answered about how to best support spatial data query in a same-
time, different-place geocollaboration environment. The following chapter describes a conceptual framework and software prototype, combining my ideas with many of the key ideas presented in this chapter, to help explore these questions of how to support collaborative spatial data query.
Chapter 3

SOFTWARE DEVELOPMENT

There are two main goals of this thesis: (1) to develop a conceptual framework for supporting collaborative spatial query and implement a proof-of-concept application using these ideas, and (2) to evaluate two different approaches within this framework for supporting spatial queries of structured data in a same-time, different-place collaborative workspace. The two approaches to be compared are: (1) the *simultaneous approach*, which enables the collaborators to view and share control of a single query builder with which they work together to construct their queries and (2) the *parallel approach*, which enables the collaborators to each use their own, private query builders to construct their queries in parallel and then share and iterate over their results. Phase one of this project is the conceptualization and development of the prototype software that will support these two collaborative spatial query approaches. This chapter discusses this software development phase, starting with the conceptual framework, then outlining the requirements and discussing alternatives for meeting those requirements, and finally presenting the actual implementation of these ideas with the SQSynC software. Phase two of this research, the user study to evaluate the two approaches, is discussed in Chapter 4.
3.1 Conceptual framework

The goal with the framework presented here is to expand upon the body of knowledge presented in the previous chapter, to outline how spatial queries should be facilitated in same-time, different-place geocollaboration. The framework identifies four core components required by all same-time, different-place geocollaboration environments: (1) synchronous data transfer between users and the interface, (2) public and private workspaces, (3) multimodal communication support, and (4) geovisualization capabilities. In addition, there are three features that are essential to support of collaborative spatial query: (1) an easy to use query building interface, (2) an implementation of the query building interface that can be used simultaneously by the collaborators, and one that can be used privately by the collaborators in parallel, and (3) appropriate concurrency control for the query building interfaces. In section 3.1.1 the core features of the framework will be described in detail, followed by detailed descriptions of the additional features for spatial query in section 3.1.2.

3.1.1 Core features

The first feature of this framework is support for synchronous data transfer between the users and the interface. This feature is what will enable most of the workspace awareness and WYSIWIS functionality that has been noted to be useful in a same-time, different-place geocollaboration environment (Armstrong, 1993). The ideal software should handle all input from each user and respond with the appropriate output for all connected users. For instance, when one user inputs a command to change the
scale of the map in a WYSIWIS view of the system, the system should respond by changing the scale of the map for not only the user who issued the command, but for all of the other connected users as well. Workspace awareness should be supported by functionality that allows users to see what other users are interacting with, functionality that enables users to easily demonstrate an area of focus on the map to other users, and functionality that enables users to easily differentiate between all of the other users and their interactions (Gutwin & Greenberg, 1999).

The second main feature of this framework is support for both public and private workspaces for all users (Armstrong, 1993). The public workspace should make full use of the synchronous interaction features described above. All users should be able to see the interactions of all the other users when they are working within the public workspace and the tools should be accessible to everyone. Within the same software interface, there should be a space that is completely private for each user. No users other than the owner of the private workspace should be able to see or interact with that space unless the owner gives them permission (permission settings should enable workspace owners to set their space to private, viewable, or viewable and editable). Users should also be able to share data from their private workspace by moving them into the public workspace (Greenberg, Boyle, & Laberge, 1999).

Another feature that should exist in this framework is multimodal communication support (Pichiliani & Hirata, 2006). Users should be able to talk to each other via audio communication, as if they are speaking on the telephone. This allows quick and easy communication and has been shown to be favored over other modes of communication by users (Brewer, MacEachren, Abdo, Gundrum, & Otto, 2000). In addition to voice
communication, there should be support for text-based communication as well. Text enables users to share rich and exact descriptions (Yu & Cai, 2009). Both modes of communication are enhanced by the workspace awareness features, such as drawing on the map.

Fourth, geovisualization capabilities are also necessary in this framework. An interactive base map should be a central focus of the interface. This base map, in addition to providing general geographic information about the world such as political boundaries, street maps, and aerial photography, should also facilitate the dissemination of and communication about more specific, user-defined spatial data. Users should be able to retrieve a specific data set based on their needs, and that data set should be displayed appropriately on the base map with minimal effort from the users.

3.1.2 Spatial query features

The previous section outlines general guidelines for geocollaborative systems, but within this conceptual framework there are also more specific guidelines for supporting collaborative spatial query. First, in order to retrieve spatial data, there needs to be a usable interface for constructing and executing spatial data queries. There needs to be support for the retrieval and the display of any spatial data from a standard structured database, and not just limited, proprietary data types. The query interface must be accessible to a wide range of people and not just experts of database administration.

Second, as outlined above, there are two different approaches to using the query interface to collaboratively query spatial data. In the simultaneous approach, the query
interface is used as a tool within the public workspace of the system. This means the query interface is a shared tool; every user can interact with the interface, provide input, and see the interactions and input of every other user simultaneously. In the parallel approach, the query interface is used within the private workspace, by an individual user. Only the owner of the workspace can interact with the query interface and no other users can see the workspace owner’s interactions or input. In this approach, the users must work in parallel, then share and iterate over their results in order to collaboratively query their data.

Third, in order to make the query interface usable in a same-time, different-place geocollaboration environment, appropriate concurrency control models must be used for each approach. In the simultaneous approach users should be able to simultaneously share and interact with the query interface as a whole, but without worry of interrupting other users. Conflicts with simultaneous user input should be limited without the need to allow only one user to have total control at a time. In the parallel approach, concurrency control for the query interface is more straightforward; only the owner of the private workspace has control of the query interface.

3.2 Requirements and solutions

The SQSynC prototype was built with the conceptual framework from the previous section in mind, but also with the intention to use it for the user study that will be detailed in Chapter 4. Therefore, the requirements for SQSynC are very similar to the
features outlined in the conceptual framework, but with a few adjustments. This section discusses these requirements in detail.

There are seven key requirements for SQSynC, based on the past research presented in Chapter 2, the conceptual framework outlined in the previous section of this chapter, and the needs of the user study that SQSynC was used to facilitate. These requirements are: (1) support for synchronous data transfer between users and the interface, (2) public and private workspaces, (3) multimodal communication support, (4) geovisualization capabilities, (5) an easy to use query building interface, (6) an implementation of the query building interface that can be used simultaneously by the collaborators, and one that can be used privately by the collaborators in parallel, and (7) appropriate floor control for the query building interfaces. The approach taken in SQSynC to addressing each of these seven requirements, as well as some potential alternatives, is described in the following seven subsections.

3.2.1 Synchronous data transfer

The first requirement for SQSynC is to include support for synchronous data transfer between the users and the interface. By incorporating this functionality, workspace awareness and WYSIWIS views will be possible to implement, which are integral features of same-time, different-place geocollaboration environments (Churcher & Churcher, 1999; Carroll, 2002). Workspace awareness will be supported in three main ways: (1) each user should be assigned a color to enable the other users to quickly and easily identify which of their collaborators are responsible for various interactions within
the interface, (2) in public workspaces, each user’s mouse cursor should be visible and identified for the other users, and (3) whiteboard drawing tools should be available to enable users to draw on the map to enhance their communication with their collaborators. Each user’s mouse cursor and anything they draw on the map should be displayed in the color that is assigned to them when they connect. WYSIWIS functionality should be implemented in all public workspaces within SQSynC, such that all interactions can be seen by all users. For example, if one user zooms in on the map, the zoom level should be changed accordingly for all of the other users as well.

One way to support these synchronous interactions would be to use special view-sharing software to enable multiple users to make use of general software that is originally intended for use by a single person (Greenberg, 1990). For instance, it would be possible to use existing, single-user GIS software along with existing view-sharing software to enable groups of remotely located users to collaboratively use the GIS software. Using the view-sharing software, one user could give another user control of her instance of the GIS software, and it would be possible for the other users to view all of the interactions in real-time. This has the advantage of being a readily available solution using software that already exists. Nothing new would need to be developed, and it would be possible to use a GIS that people are already familiar with. One downside to this solution is that each user would need to install specialized and potentially costly software. The main issue though, is that view-sharing software effectively just broadcasts the computer screen (in part or in its entirety) of one user to the other users. This removes the benefit of having one centralized environment developed specifically for geocollaboration, where public and private views can be
integrated and data can be shared among them. Also, since one user is broadcasting her
screen to every other user, bandwidth limitations could cause problems. In order to
maintain tolerable broadcasting speeds, there is often a need to greatly reduce the
resolution of the screen being shared, which makes it difficult or impossible to work with
detailed visualizations.

Instead of using view-sharing software and/or trying to adapt an existing
application to meet the needs of this project, the decision was made to develop the
SQSynC prototype from scratch. It would be possible to build this software as a desktop
application using synchronization tools, similar to what was done with Toucan Navigate
(Schafer, Ganoe, Xiao, Coch, & Carroll, 2005) and with BRIDGE and CORK
(Convertino, Ganoe, Schafer, Yost, & Carroll, 2005; Isenhour, Rosson, & Carroll, 2001).
A downside to desktop software is that each individual user needs to have the tools
installed. This can be an inconvenience to users and there is potential for compatibility
problems (with different versions of the software, or between the software and the user’s
operating system). It also limits the ability for impromptu collaboration sessions.

The best solution for SQSynC was to develop it as an entirely web based
application with synchronous interaction capabilities directly integrated, rather than using
view-sharing software, or desktop synchronization software. One key benefit to this is
that there is one, centralized application that handles all of the synchronization between
the users, fitting the client-server model as opposed to the peer-to-peer model that the use
of view-sharing software would create (Figure 18). With a single, centralized application
it is possible to keep all data and views synchronized and facilitate seamless sharing of
data between different views and users. Another advantage to this web based architecture
is the ability for users to connect to and use the application from any computer that has an internet connection and a web browser; no specialized software is required for the users. The support for synchronous interactions is implemented directly in the web based SQSynC software and facilitates the workspace awareness and WYSIWIS functionality described previously; the implementation details are described in section 3.3.

![Diagram of peer-to-peer model (left) and client-server model (right).]

3.2.2 Public and private workspaces

The second requirement for SQSynC is to support both public and private workspaces for the users (Armstrong, 1993; Brodlie, Duce, Gallop, Walton, & Wood, 2004). The cursor tracking, whiteboard drawing, and WYSIWIS functionality described in the previous subsection all must be integrated into the public workspace, but not in the private workspace. The tools in the public workspace, specifically the query building interface, should be accessible to all of the users. Access to each private workspace
should be restricted to only the owner of that workspace; no other users should be able to see or interact with it.

Each user needs to be able to access the shared public view and their individual private view at their discretion. There are a few different ways to support this. One way is to have separate windows for each view. This has the advantage of allowing the users to organize their views in whatever way suits them best. However, in the conceptual framework described previously, there would be a view generated for every user connected to the system, since each user has the ability to adjust the privacy settings of her view to enable others to view and or interact with it. This could result in a large number of separate windows if several users are collaborating, which could make it difficult for the users to keep all of the views organized and under control.

Another solution, one employed by Convertino et al. (2005) and Wu et al. (2009), is to place the public workspace right next to the private workspace, all within a single interface view. A great benefit to this is that it enables each user to easily see both the public and private workspaces simultaneously, which is especially helpful in establishing common ground (Convertino, Mentis, Rosson, Slavkovic, & Carroll, 2009). One downside to this alternative is that overall display space for the interface is sacrificed. To fit two views on a screen at once instead of just one, each view can only be half as large. This could present a problem for users who have smaller displays, such as laptops and mobile devices. Similar to using separate windows, display space becomes an even greater issue when multiple individual user views are displayed, such as in the conceptual framework.
The solution that is used in SQSynC is the implementation of a tabbed interface. There is one, unified software application that runs within a single web browser window. This keeps each view organized for the users, while allowing them easy access to any of the views by clicking on their tabs. Using tabs for the views also provides the most flexibility for supporting an indefinite number of views. Figure 19 shows an example of a tabbed interface for organizing the different views when four users are collaborating. Again, each user controls whether she wants to allow the other users to view and/or interact with her workspace. When a user views the tab of a workspace she does not have access to she will only see a message notifying her that she does not have access and must ask the owner of that workspace for permission.

3.2.3 Communication

Communication is a fundamental part of collaborative processes. There are three main modes of communication that can be supported in a same-time, different-place geocollaboration environment: text, audio, and video.
Text is a mode of communication commonly supported in internet applications. It can be supported asynchronously, such as email and message boards, or it can be supported synchronously, such as chat rooms. In a same-time, different-place environment it makes the most sense to support synchronous text communication, as in GeoJabber (Hardisty, 2009) and the Context Discovery Application (Tomaszewski & MacEachren, in press). In geocollaboration processes it is possible that certain information is more easily communicated via text than other methods, such as a data attribute that is not a natural, spoken word. For example, a user might have an easier time typing out “Find features in the census_tracts layer with a value of 2500 for the age_18_21 attribute” than verbally communicating that command, especially if it is important to relay symbols, such as underscores. A live chat component is included in SQSynC to support text-based communication; users can type their messages, click send, and they instantly appear to the other users.

Audio is another mode of communication that can be used in geocollaboration software. This is a mode of communication that comes very naturally to people since it is so frequently used in all types of collaborative activities; people speak face to face when they are in the same location and people often speak via telephones to collaborate remotely. It is no surprise that audio is the preferred mode of communication by users in a same-time, different-place geocollaboration environment, and support for this functionality needs to be provided in some way (Brewer, MacEachren, Abdo, Gundrum, & Otto, 2000). Digital, streaming audio communication is supported in SQSynC such that users can talk and listen to each other through headsets connected to their computers.
The third common mode of communication that can be supported in same-time, different-place geocollaboration is video. One benefit to video is the added awareness that it gives each user about their collaborators. Being able to see other users can provide a better sense of thoughts and feelings throughout the collaboration process, by way of visualizing their unspoken gestures (Dix, Finlay, Abowd, & Beale, 2004). One drawback to providing video communication is the amount of screen space that is required to support it. A portion of the interface needs to be allotted for each user’s video display; the more users that are collaborating, the less space there is for the other important parts of the interface. Another downside to video communication is the amount of bandwidth that is consumed by streaming video. Even if the central server has a sufficient connection to handle all of the video streams, each end user is another potential point of failure. Some users might not have an internet connection capable of handling all of the video feeds, which could result in a poor experience for them. This point is true for the system as a whole, since it relies on an internet connection, but the problem is exacerbated with streaming video. For these reasons, it was decided that video communication would not be supported in this prototype.

3.2.4 Geovisualization capabilities

The fourth key requirement of this system is to provide geovisualization capabilities, as discussed in the outline of the conceptual framework. An interactive map is a core feature that is necessary in a geocollaboration environment like SQSynC. There are several alternatives for how to incorporate a base map into a web based application.
One option for a base map is to use ArcGIS Server from ESRI (ESRI, n.d.). ESRI offers a Flex application programming interface (API) that could be integrated into SQSynC to provide base maps and visual analysis capabilities. This API was brand new when I started developing SQSynC, so little was known about it. It is also a commercial product, so it would potentially cost money to use. There were other free and more established web mapping APIs to choose from, so ArcGIS Server was not the best choice for this project, though it is worth a second look for future work.

Another web mapping API available that can be used in Flex applications is one called OpenScales (OpenScales, 2010). A great advantage to OpenScales is that it is free and open source; there is no cost to use the API and there is great flexibility in what can be done with it. Similar to the ArcGIS Flex API, the OpenScales project was just getting started when I was deciding on which base map to incorporate into SQSynC. At that point, OpenScales was not quite ready for use, but with the recent release of a full version, it is certainly worth consideration for future projects.

There are three other major providers of web mapping APIs that can be used in Flex: Google Maps, Microsoft Bing Maps, and Yahoo! Maps, of which Google is the most dominant player (Roth & Ross, 2009; Zang, Rosson, & Nasser, 2008). All three of these APIs include road map and aerial photography base map layers for all of Earth. These map layers provide a great starting point for providing context and a base for geovisualization. With Google Maps being the most widely known of the three APIs, it is the one that is most likely to be appealing or at least familiar to the most users. Given Google Maps’ good usability, long term stability, and my personal familiarity with the API, it was the best choice for the base map in SQSynC.
While the Google Maps API provides the foundation of an interactive map within the interface, it is still necessary to provide additional support for visualization of user-defined spatial data. As users retrieve various data sets to use within SQSynC, there needs to be a way to visualize these data on the map. Depending on the nature of the data points, lines, or polygons must be rendered on top of the Google Maps base map layer in their appropriate geographic locations. For the purposes of this project, this will enable users to view their data on a map, in addition to tabular format views, so they can verify that it is the data they were seeking, but this level of visualization support is just the base upon which additional functionality could be added in future, extended versions of the software.

3.2.5 Usable query interface

One key requirement for the query interface in SQSynC is to make it fairly simple to use while still enabling the types of queries that are the focus of this project. Namely, the support is required for query of structured data that are stored in a relational database (e.g. selecting all of the hospitals from a table that stores data about all types of buildings). The standard current method of performing these queries is to specify them using high level query languages, such as structured query language (SQL) (Ahlberg, Williamson, & Shneiderman, 1992). In addition to standard SQL queries, SQSynC must also support these types of queries for spatial data (e.g. selecting all of the hospitals that are within 25 miles of a specified point of interest). With a database that supports it, it is possible to perform spatial SQL queries (Egenhofer, 1994). This introduces one
possibility for supporting queries in SQSynC, which is to enable users to write SQL queries themselves and execute them within the interface.

A problem with SQL is that it can be complex and is often difficult for non-expert users to grasp (Ahlberg, Williamson, & Shneiderman, 1992). The users of geocollaboration software like SQSynC are likely to have an understanding of spatial data and the types of questions they want to ask, but they are not all going to be expert database administrators who know how to write SQL queries.

An alternative to having users write and execute their own SQL queries is to provide them with a more usable graphical user interface (GUI). With an appropriate GUI, there is no longer a need for users to have knowledge of SQL. The GUI should enable users to select the data they want using dropdown menus and sliders modeled on those developed by Shneiderman, Plaisant, Cohen, and Jacobs (2009).

To be successful, the query builder GUI needs to meet certain requirements. It needs to support a specific and straightforward work flow for the users, but be flexible and dynamic enough to support various user-defined queries. Users should be able to begin a query by using a dropdown element to select the table from which to retrieve data. From there, the query interface should adapt on the fly to all of the users’ inputs so that the chance for mistaken input is reduced. For instance, data input elements should automatically be populated with only options that make sense based on the previous inputs of the users. The interface should automatically determine if a field chosen by a user contains nonnumeric or numeric data and then provide them with the appropriate type of input element. For nonnumeric data, a dropdown element should be presented to the user and for numeric data a slider element should be used, which have been shown to
be effective for this type of dynamic query building (Li, Bao, Song, Zhang, & North, 2003; Williamson & Shneiderman, 1992). The entire query interface should be presented to the users as an easily understood form that they need to fill out. If the GUI is well-designed it should be easier than SQL for most people to use, as shown by Ahlberg, Williamson, and Shneiderman (1992), while still allowing for the types of queries that are the focus of this project.

Once the query is submitted by the user, SQSynC will need to automatically translate the user input to SQL. The SQL query will then need to be executed on the database and the results of the query must be returned. The query results should be displayed to the user on the map interface, with the option to view the data in tabular form. Once the user submits the query, the results should be displayed within a few seconds depending on the size of the result set.

3.2.6 Simultaneous and parallel approaches to collaborative spatial query

Another requirement of the SQSynC prototype is to be able to support both the simultaneous and the parallel approaches to collaborative query building. This is where the actual SQSynC prototype needs to deviate most from the conceptual framework. While in an ideal environment both of these approaches could be supported via proper implementation of public and private workspaces, for the purposes of this study the two extremes are being compared. Thus SQSynC needs to support two different modes of use.
One mode needs to enable the simultaneous approach to collaborative query building. In this mode there needs to be one public workspace and nothing else. Users must only be able to perform spatial queries simultaneously, within the public workspace; there must be no private workspaces.

The second mode of SQSynC must facilitate the parallel approach to collaborative query building. Tabs, similar to those discussed for supporting public and private workspaces previously, should be used to separate the main interface into two workspaces. Each user must have a private tab to work in and when working within it, their activities must not be visible to others. The second tab should follow the model for the public tab as described so far, with workspace awareness and WYSIWIS features, but (to support the user study described below) it must not have a query interface. The users will have to build their own queries in parallel with each other. It must be possible to share data layers from the private workspace to the public workspace, and to be able to use those public data in future queries that are performed in the private workspace, in order to share and iterate over results.

The audio and text communication capabilities need to be identical for each mode. This enables SQSynC to be used in two entirely different ways: one that forces simultaneous collaborative query building and one that forces parallel collaborative query building.
3.2.7 Concurrency control

The final key requirement for SQSynC is to include appropriate concurrency control for the query interfaces. The concurrency control for the query interface in the parallel mode of SQSynC is very straightforward; each user has their own query interface that only they can see and interact with. For this use case every user has control in the public workspace of the parallel mode, though different methods might make more sense in different cases (e.g. a team with a single leader might desire a single facilitator floor control model in the public workspace). Thus, users have equal access to the options of toggling the display of layers on the public map and removing layers from the public workspace. The concurrency control for the query interface in the simultaneous mode of SQSynC requires more consideration.

One option for concurrency control in the simultaneous approach is to have no concurrency control. In this situation, users would be able to do anything within the interface at any given time, without regard to what other users are currently doing. This would not work well for this application, as there would be nothing to stop users from simultaneously issuing conflicting commands or inputs. This would likely make it a frustrating experience for the users and would probably cause errors within the system.

Another concurrency control option is to have a single facilitator at any time who has full control of the system (Jankowski, Nyerges, Smith, Moore, & Horvath, 1997). No other users would be able to interact with the interface until the facilitator passes control on to them. Again, this is not the ideal solution in this case since the goal is to support simultaneous collaborative data query.
A third option for concurrency control is to use a tiered approach. Different users could be given different levels of control of the interface. For example, one user could be restricted to only being able to set attribute-based criteria for the query, and another user could be restricted to only be able to set spatial criteria. This could be an interesting concurrency control model to explore within an iteration of SQSynC at some point, but it did not support the core questions to be addressed in the user study, so was not implemented.

The concurrency control model most suitable for this project is somewhere between no concurrency control and a single facilitator model of floor control. The interface needs to adopt a more granular approach, such that users can assume control of individual elements of the interface at a time. When a user begins interacting with an element, such as a slider element in the query editor, that element needs to become locked for all other users. This means if one user is currently interacting with a slider element, nobody else should be able to interact with that specific slider element until the first user is done. Users should, however, still be able to interact with any other part of the interface that is not currently in use. This enables users to simultaneously work within the shared query interface without conflict.

### 3.3 Implementation results

The first main result of this research is the proof-of-concept SQSynC software prototype. SQSynC is a functional implementation of my ideas for designing a software application to support same-time, different-place geocollaboration with a focus on spatial
data query. The measure of success for this phase of the research comes from the requirements laid out in the previous section of this chapter and the need for SQSynC to facilitate the user study in Phase 2. The two different modes of use (simultaneous and parallel) of the SQSynC prototype are shown in Figure 20 and Figure 21. The visually noticeable differences are minor, but all of the differences in functionality between the two modes will be made more evident throughout this section.

Figure 20. The SQSynC prototype in simultaneous mode. (1) Task Manager, (2) Whiteboard Tools, (3) Data Layers, (4) Connected Users List, (5) Query Editor, (6) Chat Panel
SQSynC is entirely web-based; the only requirement for users is having a computer and an internet connection. It is built in Flex with a Java backend connected to a PostgreSQL database (Figure 22) (Adobe Systems Incorporated, 2010; The Apache Software Foundation, 2010; PostgreSQL Global Development Group, 2010). PostGIS is installed on the PostgreSQL server to enable spatial data transactions (PostGIS, 2010).
Google Maps API for Flash is used to provide the base map upon which all other spatial data is overlaid (Google, 2010). The types of spatial data that are supported by SQSynC are vector points, lines and polygons. The geographic coordinates of these data, as well as any attributes, are stored in the PostgreSQL database. This enables the data to be queried based on their spatial locations as well as any other attributes of the data.

![Figure 22. System architecture for SQSynC](image)

Being a same-time, different-place collaboration tool, one of the first major challenges when developing the foundation of the software was building support for synchronous interactions between the users. This was accomplished by setting up real time data services on the server side of the application, using BlazeDS (Adobe Systems Incorporated, 2009). Put simply, a channel was established on the server that constantly listens for new messages. When a message comes in, it gets pushed out to any subscribers of the channel. The Flex application was set up to use this channel to send
and receive messages. Each instance of the client then becomes a publisher and a subscriber to the streaming channel set up on the server. This provides the base functionality to enable all of the real-time interactions between the users of the interface.

With this synchronous data sharing functionality in place, it was possible to begin building the WYSIWIS elements of the software. For instance, when a user pans the map, this interaction gets sent as a message to all the other connected clients. Within the message is all of the information needed for all of the other clients to emulate the exact action that took place on the original client instance. This also applies to all other map interactions, such as zooming, toggling layers, and switching base maps. WYSIWIS functionality was implemented for the simultaneous mode of the software, as well as for the public tab of the parallel mode. In the simultaneous mode, all of the interactions with the shared query builder are also WYSIWIS.

Along with all of the WYSIWIS functionality, the streaming data services were also used to support some of the workspace awareness features. Mouse cursor locations were captured and shared with all of the clients. This enables all connected users to see where every other user is currently pointing, or what tools they are currently interacting with. Cursors are displayed as a rectangle with the user’s name in it and a black arrow in the upper left corner, as shown in Figure 23. Upon connecting to the SQSynC software, each user is assigned a color, which becomes their signature for all of the WYSIWIS features. The color assigned to Kevin in this example (Figure 23) is purple. A whiteboard tool was also built into the prototype. This tool allows users to quickly and easily draw on the map. As a user draws, everyone else can see the annotations in real time, making it quick and easy to communicate about specific locations. Figure 23 shows
a user named Kevin drawing on the map. The options in the whiteboard dropdown menu are to activate or deactivate the whiteboard and to clear all of the whiteboard drawings. Deactivating the whiteboard hides, but does not erase, the current drawings and allows the users to interact with the map again.

Figure 23. A user named Kevin draws on the map. His cursor and his drawing are visible to the other users. The whiteboard menu is opened, showing the options to activate/deactivate and clear the whiteboard.

A key feature of SQSynC in relation to this thesis is the query editor (Figure 24). This is the part of the interface where the users construct and edit their data queries, and is the main focus of the user study. The query editor had to be usable for people who are not experts at writing SQL queries by hand, yet also flexible and powerful enough to retrieve any set of data from the database given the criteria declared by the user for the data’s attributes and spatial relationships. The result is a novel approach to a spatial data query interface, based on previous research on interface design and dynamic query showing the usefulness of interface widgets like sliders and dropdown elements in lieu of
The query editor enables users to go through a logical workflow to construct queries without the need to type anything. This is accomplished in part by having the interface adapt to the input from the user to help guide the user through the process and...
also limit their input choices to only those that are valid. The user starts by using the first dropdown element to choose the layer from which they want to select data (Figure 25). This can be any of the tables that exist in the database, or any layers that have already been added to the map from previous queries. When a layer is chosen, a new query is started and the second dropdown element is automatically updated to include all of the attribute fields from the selected layer. This element, shown in Figure 26, is given more visual prominence, since it can be used multiple times to set all of the attribute criteria for the query.

When a user selects an attribute field, a new widget gets added to the query editor interface. When a new field is selected, the software automatically does a couple of processes in the background. The first process is to determine if the selected field contains numeric or nonnumeric data. If the field contains numeric data, the widget

![Layer select widget](image)

*Figure 25. Layer select widget (for demonstration purposes, not all database layers are shown).*
provides a slider for setting the criteria for the attribute. The user can also select whether they want to find data that is exactly equal to, is not equal to, or is within the range of their input. Again, the interface adapts to the user input; if the ‘range’ option is selected, the slider provides two thumbs and if the ‘equal’ or ‘not equal’ options are selected, the slider provides one thumb. If the selected attribute field contains nonnumeric data, the widget has a dropdown element from which the user can select the criteria for the attribute. The user can set whether they want data that is, is like, or is not equal to the value they provide. If ‘like’ is selected, the dropdown menu adapts to allow the user to type in a value. The second process that happens when an attribute field gets selected is to find all of the possible values for the field. For sliders, the minimum and maximum values are determined and used as the lower and upper bounds of the slider bar. For dropdown menus, all of the possible values are included as options. In Figure 27, two attribute fields have been added to the query: the first one has numeric data, the second one has nonnumeric data.
The next two dropdown elements in the query editor allow the user to set a spatial relationship that must exist for the data they are retrieving. In the first of these dropdown elements, the user can select between a list of eight different spatial relationship types: (1) intersect, (2) are within a distance of, (3) completely contain, (4) are completely within, (5) have their centroid in, (6) touch the boundary of, (7) are identical to, and (8) do not intersect. If ‘are within a distance of’ is selected, the user is provided with an input element that they can type a distance into, as well as a dropdown element to select the unit of measurement (centimeters, feet, inches, kilometers, meters, miles, or yards). The second dropdown element allows the user to select the data layer with which this spatial relationship exists. These two dropdown elements for setting the spatial relationship are shown in Figure 28.

Since the query editor adapts ‘on the fly’ in order to accommodate appropriate user input, it remains flexible to the types of queries it can perform. Any data that exist
Figure 28. Two dropdown elements allow the user to select a type of spatial relationship (top) and layer with which this relationship exists (bottom).

in the database, which can include any vector point, line, or polygon data, can be queried using this interface. The only input that is required to run a query is the base data layer to select features from. If no other criteria are set, the system will simply retrieve all of the data in that layer. Since users are able to select layers straight from the database or from previously retrieved map layers, they are able to perform multiple iterations of subqueries. For example, a user can select a new subset of data based on a spatial relationship with a previously selected subset of data that was retrieved by another user. Figure 29 shows the query editor with criteria set for the attributes and spatial relationship of the data to be selected from the census_blocks layer. The equivalent of this query typed as SQL code would be:

```sql
SELECT census_tracts.*
FROM census_tracts, (SELECT hospitals.* FROM hospitals WHERE [hospital criteria]) as spatial_table_alias
WHERE (census_tracts.age_18_21 BETWEEN 0 AND 3500)
AND (census_tracts.cnt_name = 'MIAMI-DADE')
AND ST_Contains(census_tracts.the_geom, spatial_table_alias.the_geom)
```
where [hospital criteria] would be replaced with additional SQL code to establish any criteria specified for the original hospital layer. Instead of requiring the user to type this query in SQL, the query editor allows the users to perform the same query in a simpler way through the visual interface described. The form-based query can be easily constructed, read, and understood as “Select features from census_tracts where age_18_21 is between 0 and 3500 and

![Query editor with attribute and spatial criteria set.](image)

*Figure 29. Query editor with attribute and spatial criteria set.*
cnt_name is MIAMI-DADE that completely contain the features in the hospitals layer.”

The final element in the query editor is the ‘Run Query’ button. When this button is pressed, SQSynC translates all of the user input to SQL code, executes the query on the database, then returns the resulting data layer to the client. The data layer is added to the map and a new checkbox is added to the data layer list (Figure 20-3 and Figure 21-4) to enable the users to toggle the map layer on or off. For the tasks in the user study, this process would generally take no more than five seconds, depending on the amount of data being returned. The map layer is automatically given a unique name, by appending a number to the end of the name of the original table in the database. The user is able to change the name of the layer by right-clicking on it. In the right-click menu, there are also options to open the attribute table of the layer, to remove the layer from the application, and to zoom to the extent of the layer. Figure 30 demonstrates two layers added to the map: hospitals represented by the purple points and census tracts represented by the teal polygons, as well as the menu that appears when a user right-clicks on one of the layer names. The attribute table enables the user to view all of the data for the selected layer in tabular form, and to sort by each of the columns (Figure 31).
Figure 30. Two layers have been added to the map: hospitals (points) and census tracts (polygons). Right-clicking on a layer name opens a context menu with additional options.
In order to facilitate the user study in Phase 2 of this project, SQSynC had to support two main modes of use: one for the simultaneous collaborative query building approach, and one for the parallel collaborative query building approach. Both modes use the same interface for the query editor; the key difference is that in the simultaneous mode there is one, shared, WYSIWIS query editor and in the
parallel mode, each user has his own, private query editor that no one else can see or interact with.

For the shared query editor of the simultaneous approach, it was necessary to institute some form of concurrency control to prevent conflicts if multiple users try to change a single input element at the same time. As outlined above, a mixed approach was used that is less restrictive than giving a single user total concurrency control, but more restrictive than a free-for-all mode. When any user interacts with an element of the query editor, that element gets locked for all other users so that nobody can interact with an element that is currently in use, but they can still interact with all of the other elements in the interface. When an element is locked to a user, it gets highlighted with the color that was assigned to the user upon connecting to SQSynC, making who has control of the element clear to the other users (Figure 32).

![Figure 32](image)

*Figure 32. A user named Kevin is typing in a value after clicking the ‘like’ button.*
For the parallel approach, no concurrency control is needed for the query editor, since the query editor is only accessible in the private tab. Otherwise, the way in which the system works is very similar to the simultaneous approach. The query editor works the same way aside from not allowing interactions from other users. When a user retrieves a data layer, it is placed in the panel labeled ‘Your Data Layers’. A new option is added to the right-click menu to share the selected layer. Upon clicking this option, the layer is duplicated on the public tab of the interface, and ‘_p#’ is appended to the name, where # is a unique number. Another panel exists in the private tab, labeled ‘Public Data Layers’, which has a list of all of the data layers that have been shared by any of the users. This enables the users to interact with any of the public layers within their private tab, such as displaying them on the map, viewing their attribute tables, or using them in their own queries. For example, one user might retrieve all of the hospitals based on criteria that they know to be relevant, then share that layer so that the other user can find all of the relevant census tracts that contain those hospitals. Each user has equal control over the public workspace, so they have the ability to remove any layers from the public list (but they can be shared again by the creator). Figure 33 shows the private and public data layer lists and the new option to share data layers.

The public tab in the parallel mode of the interface is very similar to the simultaneous mode with two exceptions: (1) there is no query editor and (2) there is a new option to join two layers together. By right clicking on a layer in the public tab and clicking ‘Join’, the users have the ability to join the selected layer to another layer. Upon setting the two layers to be joined, as well as the corresponding fields in each layer that
In the parallel approach, data layers are separated into private and public lists, and a new option exists in the right-click menu to share a private layer (make it public).

The join should be based on, and then executing the join, a third layer is created. The third layer is composed of the coincident data between the two layers, based on the fields that were selected (i.e. an intersection of the two layers). For example, Figure 34 shows the census_blocks layer being joined to the schools layer. A new layer will be created of all the schools that have a county value that matches a county value from the census_blocks layer, and all of the attributes from both of the layers will be included in the attribute table of this new layer where the counties match. Alternatively, the user can
decide to keep all of the records from one of the tables no matter what, and just bring in the attributes of the other table where there is a match.

If two layers that came from the same original source are selected to be joined, all of the other options are disabled since it is a much more straightforward join (Figure 35). For example, if one user retrieves a subset of schools based on a given set of criteria and another user retrieves a different subset of schools based on other criteria, the users can then join those two subsets of schools together to create a third, cumulative layer (i.e. a union of the two input layers). This functionality was provided in the public tab of the parallel interface to give the users the ability to collaboratively iterate over the data layers they retrieved. The query editor can be used in a similar fashion in both the parallel and simultaneous approach, by performing a subquery on a previously queried data layer.

*Figure 34. Join Layers interface with data layers selected from two different sources*
The final feature included in SQSynC in order to facilitate the user study is the task manager. When a user clicks the ‘Tasks’ button (Figure 20-1 and Figure 21-2), the task manager interface is opened (Figure 36). The purpose of this feature is to collect the results for each of the tasks that the participants of the user study are asked to complete, but it is presented to the users as an agenda and report builder. It has a tab for each of the four tasks, which displays the main objective of the task (which is also provided to the participants on a sheet of paper) and provides an area that the participant can drag data layers into. The participants are instructed to drag the layer or layers that satisfy the requirements of the tasks into this interface. A layer that fully satisfies the task is a subset of data retrieved from the database that meets all of the criteria laid out in the task.
instructions. Upon submitting the report, their responses for the four tasks are saved to a database where they can be reviewed later.

With all of the described functionality of SQSynC working, it stands as a proof-of-concept of ideas outlined in this thesis for how to support spatial data queries in a synchronous geocollaboration environment. It has been shown that it is possible, with current technology, to implement these features into a functional, web-based software application. After developing this working prototype, the next step was to use it to facilitate the user study to gain insight about the usability of the parallel and simultaneous
approaches to collaborative spatial data query, which is described in the following chapter.
Chapter 4

USER STUDY

The SQSynC prototype enables a user study to evaluate the two different approaches to performing spatial queries of structured data in a same-time, different-place collaborative environment. The foci of the user study are to (a) obtain user input on functionality of and preferences for the two interface approaches and (b) to add to understanding of how interface choices in geocollaborative systems influence use of the systems for collaborative activities. To answer these questions, participants were asked to use SQSynC in a controlled laboratory setting while relevant data was collected through logs of their interactions with the prototype as well as through a follow-up survey. The study is a formative study, designed to provide guidance on future development of SQSynC and related geocollaborative applications and thus relies primarily on qualitative data through which insights about system use can be derived. The methods used to conduct the user study are presented in the next section, followed by the results of the study.

4.1 Methodology

This section outlines the methodology of the user study, starting with a description of the participants in subsection 4.1.1. Subsection 4.1.2 describes the data used to support the tasks included in the study, followed by an overview of the research
environment in 4.1.3. The use scenarios and tasks that were completed by the participants as part of the study are presented in 4.1.4. The types of data that were collected as the participants worked through the study are described in 4.1.5 and then the post session questionnaire is presented in 4.1.6. Subsection 4.1.7 outlines the overall procedure of the study, which was informed in part by the lessons learned in the pilot study described in 4.1.8.

4.1.1 Participants

Participants were solicited from Pennsylvania State University (PSU) students and researchers. The only requirements were: (1) participants had to be 18 years or older and (2) participants had to have taken at least one GIS related course, since these are the target users of this type of software and so that all participants had at least some familiarity with the types of tools and concepts they would encounter in this user study. A monetary incentive of 15 dollars was offered to the participants for their time.

A total of 16 people participated in the user study. The participants were all students of the Department of Geography at PSU and were between the ages of 18 and 30 years old. Four of the 16 participants were female. The participants worked in pairs during the user study, so after agreeing to participate each individual was sent a link to a customized schedule hosted by Doodle (http://www.doodle.com). The Doodle schedule enabled the participants to declare all of the time slots that they would be available and willing to participate in the user study. To reduce the chance of friends signing up to work together, no participant could see the entries of any of the other participants in the
schedule and the pairs were chosen randomly from the pool of people who had overlapping availability.

4.1.2 Data

SQSynC was set up with a previously assembled set of data prior to the user study; participants did not need to find, download, or import any data and all participants had the same data available to them. The selection of the data to use was based on the availability of various types of data to the public for a single location. A location outside of Pennsylvania was selected for which to assemble the data to reduce any bias that may exist due to the participants being students in Pennsylvania and thus being potentially familiar with data about the state. Florida was chosen as the geographic focus of the use scenario around which tasks were constructed and 20 different publicly-available GIS layers were selected from Florida, including census blocks, housing, transportation, and schools.

4.1.3 User study environment

The user study environment was set up to simulate a synchronous, distributed collaboration scenario while still enabling a single moderator to be available to assist the participants and keep them on task. To accomplish this, the user study was carried out in a single computer lab with multiple sessions. Each session included one team of two participants. Each participant was seated at a computer and provided with a headset
including a microphone. Participants were seated with their backs facing each other and they were only able to see their own monitor, since remotely located collaborators would not be able to directly view their teammate’s face or computer monitor. All communication was done through their headsets or text chat via the chat support built into SQSynC, again to simulate a distributed collaboration scenario.

Each group worked together using SQSynC after receiving brief instructions on how to use the system and about the various options that were available to them. These instructions were provided in the form of a video, so that consistency was ensured for every participant (the video can be viewed at http://ksross.com/sqsync/tutorial and the sample walkthrough task is in APPENDIX A).

Since there were two different approaches to collaborative query building being compared, there were two options for the user study: (1) evenly divide the participants and have half of them use one approach and have the other half use the other approach or (2) have all of the participants use both of the two approaches. Dividing the participants has the benefits of allowing participants to do more tasks with the particular interface used in a fixed length session and of removing the possibility of any order effects where participants might be more knowledgeable and prepared for the second approach they use. On the other hand, having all of the participants use each approach enables within-subject comparisons of the two approaches. Since a core goal in this phase of the research is to understand which approach is preferred by users as well as to obtain other relative comparisons, all of the participants used both approaches. To control for order effects, half of the participants started with the simultaneous approach and the other half started with the parallel approach.
4.1.4 Use scenarios and tasks

Two different sets of tasks were created that the participants completed using SQSynC. All of the tasks were designed to be equivalent in terms of what was required of the users. For instance, each task had between three and five different attribute criteria (e.g. ‘houses must have 4 bedrooms’) and one spatial relationship (e.g. ‘roads must intersect the hurricane track’). Every task can be completed with a total of two queries: one to retrieve the layer to which the data is spatially related, and one to retrieve the main data layer, though more than two queries can also be used to complete each task. Making complexity and the nature of the sets of tasks as close to equivalent as possible reduces the potential for the preferences of the participants to be biased depending on which set of tasks was completed with which approach.

The objective with the two sets of tasks was to enable comparison between the approaches while avoiding having participants simply repeat the same tasks in the second part of their session. The first set of tasks involves an emergency response scenario and the second set of tasks involves a real estate shopping scenario. These are two scenarios that commonly involve people with varied knowledge or expertise working together to solve an inherently spatial problem (National Research Council, 2007; Federal Emergency Management Agency, 2008; Williamson & Shneiderman, 1992). Having two scenarios from completely different domains reduces the potential of the participants becoming too familiar with the types of tasks and data by the time they start the second approach. Also, to account for order effects with the two scenarios, the order in which
they were given to the participants was varied, along with the order of which approach was used first.

With two data sets and two orders, there was a total of four different user study setups that were used with equal frequency: (1) simultaneous approach with emergency response scenario first, (2) simultaneous approach with real estate scenario first, (3) parallel approach with emergency response scenario first, and (4) parallel approach with real estate scenario first. Each scenario included four tasks that the participants were asked to complete. Each task prompted the users to find the appropriate subset of data that met the task criteria from the vast amount of data available in the database. In order to successfully complete the tasks and find the correct answers, the participants were required to perform spatial queries in SQSynC.

As discussed in the background chapter, one logical reason why people might collaboratively query spatial data is to take advantage of their varied domain expertise. Therefore, participants were divided into groups of two and each person represented a different set of expertise. In order to simulate this, each participant was given a subset of information that their collaborator did not receive. Each piece of information given to the participants was necessary to complete the task. This divided information ensured that the teams needed to collaborate in order to be successful.

The tasks that were given to the participants, as well as the two different subsets of information for each task are shown in Table 1 and Table 2. The task instructions, shown in bold in each table, were exactly the same for both participants in each session. The following two subsets of information for each task were divided, such that one participant got only the first subset, and the other participant got the other subset. For
instance, for the first task in the hypothetical emergency response scenario, both participants in each session knew that they had to find all medical centers within the critical area that provided general health care services and had a sufficient number of beds. However, only one participant knew which counties made up the critical area, and only the other participant knew to look for hospitals of type ‘ACUTE CARE’ with at least 200 beds. This ensured that the participants had to collaborate in order to successfully complete the tasks. The full task sheets as given to the participants are available in APPENDIX B, APPENDIX C, APPENDIX D, and APPENDIX E.
Table 1. The four tasks that were given to the participants for the emergency response scenario, as well as the two different subsets of information that were provided for each.

<table>
<thead>
<tr>
<th>Task 1</th>
<th>Find all medical centers within the critical area that provide general health care services and have a sufficient number of beds.</th>
</tr>
</thead>
</table>
| Information provided to participant 1 | -counties are in the counties table, medical centers are in the hospitals table 
-the critical area includes Broward, Martin, Miami-Dade, and Palm Beach counties (name field in the counties table) |
| Information provided to participant 2 | -counties are in the counties table, medical centers are in the hospitals table 
-general care means hospitals of type ACUTE CARE (type field) 
-sufficient number of beds is more than 200 (beds field) |

<table>
<thead>
<tr>
<th>Task 2</th>
<th>Parts of this database are being kept up to date in real time. Which rescue personnel are within range of the critically injured victim and are capable of assisting with a spinal injury?</th>
</tr>
</thead>
</table>
| Information provided to participant 1 | -the critically injured victim is in the spinal_injury table, the personnel are in the personnel table 
-the personnel must have a helicopter to air lift the injured party |
| Information provided to participant 2 | -the critically injured victim is in the spinal_injury table, the personnel are in the personnel table 
-appropriate range is 25 miles 
-the vehicle field in the personnel table describes the vehicle the person is currently using |

<table>
<thead>
<tr>
<th>Task 3</th>
<th>Find all high-capacity evacuation routes that intersect the hurricane track.</th>
</tr>
</thead>
</table>
| Information provided to participant 1 | -the hurricane track is in the hurricane_track table, routes are in the roads table 
-sufficient capacity is greater than 2000 (capacity field) 
-the routes must be at least 2 lanes (lanes field) |
| Information provided to participant 2 | -the hurricane track is in the hurricane_track table, routes are in the roads table 
-the routes must be egress routes (egress field = true) |

<table>
<thead>
<tr>
<th>Task 4</th>
<th>Find all facilities housing unsafe amounts of dangerous types of hazardous materials within the critical area.</th>
</tr>
</thead>
</table>
| Information provided to participant 1 | -the critical area for hazardous materials is within 15 miles of the hurricane track 
-the hurricane track is in the hurricane_track table 
-unsafe amounts include anything above 50 
-dangerous types of materials include Class 1, Class 2, and Class 3 |
| Information provided to participant 2 | -hazmat facilities are in the hazmat_facilities table 
-facilities that function as storage or disposal are the ones currently housing materials (function field) 
-the amount field describes the amount of materials currently at the facility 
-the type field describes the type of materials currently at the facility |
Table 2. The four tasks that were given to the participants for the real estate shopping scenario, as well as the two different subsets of information that were provided for each.

<table>
<thead>
<tr>
<th>Task 1</th>
<th><strong>Find affordable and appropriately sized waterfront property.</strong></th>
</tr>
</thead>
</table>
| Information provided to participant 1 | -property is in real_estate table, water bodies are in waterbodies table  
-`waterfront` is within 1 mile of a water body (not including the ocean)  
-the water body feature cannot be a swamp or marsh (feature field) |
| Information provided to participant 2 | -the property should be empty land (type field)  
-the size should be at least 1 acre (acres field)  
-the price should be between $100,000 and $250,000 |

<table>
<thead>
<tr>
<th>Task 2</th>
<th><strong>Find affordable homes that are on a large beach.</strong></th>
</tr>
</thead>
</table>
| Information provided to participant 1 | -beaches are in beaches table, homes are in real_estate table  
-large beaches have an area of at least 3,000,000 (area field in beaches table) |
| Information provided to participant 2 | -affordable is less than $355,000 (price field)  
-the type of home desired is a condo (type field)  
-the home should have a swimming pool (pool field) |

<table>
<thead>
<tr>
<th>Task 3</th>
<th><strong>Find accessible homes, with a garage, near a church.</strong></th>
</tr>
</thead>
</table>
| Information provided to participant 1 | -homes are in real_estate table, churches are in religious_centers table  
-accessible homes have no more than one floor (floors field)  
-near a church means within 10 miles |
| Information provided to participant 2 | -the garage should fit at least 2 cars (garage field)  
-the church should be non-denominational (type field) |

<table>
<thead>
<tr>
<th>Task 4</th>
<th><strong>Find large homes that are close to small schools.</strong></th>
</tr>
</thead>
</table>
| Information provided to participant 1 | - schools are in schools table, homes are in real_estate table  
-small schools have an enrollment (enrollment field) of less than 500  
-close means within 15 miles |
| Information provided to participant 2 | -large homes:  
-are over 3000 square feet (sqft field)  
-have at least 4 bedrooms (bed field)  
-have at least 3 bathrooms (bath field)  
-school type should be private (op_class field) |
4.1.5 Data collected

In order to be able to examine exactly what each participant did and said during each task session, all of the interactions that each participant had with SQSynC were captured on video and audio. To accomplish this, Camtasia Studio screen recorder software was used (TechSmith Corporation, 2010). This step was taken as a way to log this potentially important information, so that if any critical incidents occurred or specific comments of interest were made by participants, it is possible to view and listen to the recording of exactly what happened. The software also kept a log of all of the results that satisfied each task, as declared by the participants. This was supported through the use of the task manager interface described in the previous chapter. Upon submitting the answers, they were stored in a database.

4.1.6 Post-session questionnaire

Following the collaborative spatial query activity, the participants were given a survey to complete about the user study and SQSynC (Table 3). The first three questions of the survey were structured questions with fixed response choices designed to assess the participants’ preferences of the simultaneous and the parallel approaches to the query building exercise. Questions 1 and 2 had them rate their level of satisfaction of each of the two approaches on a multiple-choice Likert scale of 1: very dissatisfied, 2: dissatisfied, 3: satisfied, 4: very satisfied. Question 3 asked the participants to select which of the two approaches they preferred and then give an explanation of why they
Table 3. Follow-up survey.

<table>
<thead>
<tr>
<th></th>
<th>1. On a scale of 1 to 4, how would you rate your level of satisfaction with the parallel query building approach?</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>- 1. Very dissatisfied</td>
</tr>
<tr>
<td></td>
<td>- 2. Dissatisfied</td>
</tr>
<tr>
<td></td>
<td>- 3. Satisfied</td>
</tr>
<tr>
<td></td>
<td>- 4. Very satisfied</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>2. On a scale of 1 to 4, how would you rate your level of satisfaction with the simultaneous query building approach?</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>- 1. Very dissatisfied</td>
</tr>
<tr>
<td></td>
<td>- 2. Dissatisfied</td>
</tr>
<tr>
<td></td>
<td>- 3. Satisfied</td>
</tr>
<tr>
<td></td>
<td>- 4. Very satisfied</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>3. Overall, which of the two modes of collaborative query building did you prefer?</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>- Parallel query building</td>
</tr>
<tr>
<td></td>
<td>- Simultaneous query building</td>
</tr>
<tr>
<td></td>
<td>Why?</td>
</tr>
</tbody>
</table>

|   | 4. Can you remember specific incidents when the parallel query building procedure was confusing or difficult? If yes, please describe them and explain how you dealt with them. |

|   | 5. Were there any features of the parallel query building interface/procedure that you particularly enjoyed? If yes, please describe them. |

|   | 6. Can you remember specific incidents when the simultaneous query building procedure was confusing or difficult? If yes, please describe them and explain how you dealt with them.  |

|   | 7. Were there any features of the simultaneous query building interface/procedure that you particularly enjoyed? If yes, please describe them. |

|   | 8. Do you have any suggestions for improving the overall collaborative query-building process? |

|   | 9. Are there any other real-world scenarios that you can you think of where collaborative query building, either simultaneous or parallel, might be particularly necessary or useful? |
preferred it to the other approach. The remaining questions were open-ended focused on
pros, cons, potential improvements, and potential applications. For questions 4 and 6 the
participants were asked to describe any critical incidents or points of confusion that they
encountered with each of the two approaches. As a complement, for questions 5 and 7 the
participants were asked about any features of the two approaches that they
particularly liked. The next question prompted the participants for any suggestions about
how the overall collaborative query building process could be improved. Finally the
participants were asked for their opinions about any other situations that would be well
served by collaborative query building tools.

The survey was administered using SurveyMonkey, a web-based survey
management tool, to make it easy for the participants to fill out while they were already
using a computer (SurveyMonkey, 2010). This also made it possible to organize the
questions based on the order in which the participants used the two different modes of the
software. Table 3 demonstrates the way the survey was organized for a session in which
the parallel approach was used first. When the simultaneous approach was used first, the
order of questions 1 and 2 was switched, the order of the answer choices in question 3
was switched, and questions 6 and 7 came before question 4.

4.1.7 Overall procedure

Each user study session was limited to one hour. When the participants arrived,
they were given a copy of the informed consent form, which included a brief overview of
the user study. After ensuring they understood everything in the informed consent form,
the participants were seated at two different desks, each with a computer for them to use, and they were shown the tutorial video. The video went over all of the key aspects of the software that they would need to be familiar with, and went through a sample task similar to the ones that they were asked to complete. A printed copy of this sample task (APPENDIX A) was given to each participant so that they could follow along with the video. This served as an example of how to use the query builder interface, as well as a demonstration of how each participant had a different subset of information that was necessary to successfully complete each task. The tutorial video was 10 minutes long. Following the video, the participants were given 20 minutes to work on the first set of tasks, using the first mode of the software (simultaneous or parallel). The next 20 minutes were used to work through the second set of tasks in the second mode of the software, and the remainder of the time was allotted to complete the survey. In the two 20 minute work sessions, the participants were told to complete as many of the four tasks as they could in that amount of time; they were cut off at 20 minutes regardless of how many tasks were completed.

4.1.8 Pilot study

To help with determining exactly how to set up each piece of the user study described above, a pilot study was conducted. A research faculty member and a PhD student from the Department of Geography at PSU were asked to run through the designed scenarios using SQSynC. This pilot study proved quite valuable, as it provided some insight about the software prototype and the design of the user study.
The original design for this user study included the intent to collect more quantitative data about the performance of each team of participants, to see which of the two approaches resulted in faster and more accurate completion of the tasks. One of the reasons for dropping those metrics was the number of participants that would have been required (and the limited funds available to compensate participants). More importantly, the pilot study determined that the tasks as initially designed were more challenging to complete than expected. One of the original concerns was that the tasks would be too easy for the participants to complete. This was not the case and in fact the opposite was true, most likely because one hour was not enough time for the participants to watch the tutorial video, become familiar with the software, get comfortable with the idea of collaborating with their assigned partner, complete all of the tasks using both approaches, and finish the follow-up survey. Since the initial plan was for user performance in each of the two approaches to be measured by (a) how many tasks were completed accurately and (b) how fast the tasks were completed, the fact that the expert geographers in the pilot study were unable to complete the tasks within an hour presented a problem.

To account for the time constraints, three options were considered: (1) make all of the tasks much simpler to complete, (2) increase the maximum time of each session to two hours instead of one, or (3) focus more on the qualitative aspects of the study, such as user preferences and opinions, as opposed to comparing the performance for each approach. The problem with the first option was that the complexity of the tasks is necessary to force collaboration between the participants, which is essential to the study. The second option would have been ideal if it was possible to find enough participants, but this was not possible within time and funding limitations. The third option has the
disadvantage of not allowing for the collection of more generalizable quantitative results, but the advantage of enabling participant responses with greater breadth and depth to be collected. The third option was also the most viable given the constraints of this project; because of these advantages, it was chosen over the other two options.

The decision was made to keep the limit for each session at one hour, reduce the number of tasks in each scenario from five to four, and to take the pressure of measured performance off of the participants and instead focus on their responses to the survey. Instead of focusing on statistical analysis of the causal relationships between the variables (e.g. approach used and number of accurately completed tasks), more open-ended questions with potential for rich responses were posed to participants. The compensation for each participant was also raised to 15 dollars from its initial value of 10 dollars, since fewer participants would be needed. These turned out to be good decisions, as it proved to be difficult to find many people willing to participate, even when offering 15 dollars for one hour; increasing the time length without additional funding while also expecting to have enough participants for statistically significant results would not have worked out.

4.2 Results

This section presents the results of the user study, based on the responses to the survey (Table 3). In total, 16 geography students from PSU participated in the user study. As the participants worked in pairs, there were a total of eight sessions, which each lasted one hour. The results are the 16 individual responses to the nine question
The organization of this section is based on the themes of the survey questions. Analysis of the participants’ levels of satisfaction and preferences for the two collaborative spatial query building approaches is presented in section 4.2.1. Section 4.2.2 goes over the critical incidents and points of confusion for each approach. The features of the two approaches that participants found enjoyable are outlined in section 4.2.3. Finally, section 4.2.4 discusses the suggestions that were made by participants regarding potential improvements to the collaborative query building process, as well as ideas about other real-world scenarios where tools like SQSynC might be useful.

4.2.1 Satisfaction and preferences

The first two survey questions asked the participants to rate their level of satisfaction with both the parallel approach and the simultaneous approach to collaborative spatial query building on a Likert scale: (1) very dissatisfied, (2) dissatisfied, (3) satisfied, and (4) very satisfied. The results are shown in Table 4. Overall there were no participants who were very dissatisfied with either of the two approaches. There were seven participants who were dissatisfied with the parallel approach and only five who were dissatisfied with the simultaneous approach; however, there were also more participants who were very satisfied with the parallel approach compared to the simultaneous approach. Overall, the ratings are fairly close between the two approaches; there is no clear winner as far as one approach receiving a far greater satisfaction rating than the other approach.
Table 4. Frequency of satisfaction ratings for the two approaches.

<table>
<thead>
<tr>
<th>Response</th>
<th>Parallel Approach</th>
<th>Simultaneous Approach</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 – Very Dissatisfied</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2 – Dissatisfied</td>
<td>7</td>
<td>5</td>
</tr>
<tr>
<td>3 - Satisfied</td>
<td>5</td>
<td>8</td>
</tr>
<tr>
<td>4 – Very Satisfied</td>
<td>4</td>
<td>3</td>
</tr>
</tbody>
</table>

Another way to analyze the results from the first two survey questions is to look at a within-subject comparison of the responses from the 16 participants (Figure 37). This chart shows that it was typical for a participant to rate their satisfaction level as 2 (dissatisfied) for one of the approaches and then rate their satisfaction level for the other approach as either a 3 (satisfied) or 4 (very satisfied). It also makes it easy to see that there was only one participant who was dissatisfied with both the parallel and the simultaneous approaches; the rest of the participants were either satisfied or very satisfied with at least one of the approaches. Overall, the results of the first two survey questions show that the participants were generally satisfied with at least one of the two approaches, they typically had a higher level of satisfaction for one of the approaches over the other, and there were an equal number of participants who rated their satisfaction level for the parallel approach higher than that for the simultaneous approach as there were participants who rated their satisfaction level for the simultaneous approach higher than that for the parallel approach (six of each).
Figure 37. Within-subject comparison of responses to the first two survey questions. Circles around the points indicate that the participant rated the satisfaction level the same for each approach and the color indicates which approach was preferred (green for simultaneous, blue for parallel).

The third question of the survey more directly assesses the preferences of the participants when comparing the two approaches. Of the 16 participants, nine preferred the parallel approach while seven preferred the simultaneous approach. This again shows a fairly even split as far as which approach was preferred. One factor that could possibly contribute to this even split in preferences is the order in which each approach was used. Eight of the participants used the parallel approach first while the other eight participants used the simultaneous approach first. 11 of the 16 participants stated a preference for the
second approach they used, which might suggest that the more time they spent using the SQSynC software in general, the more satisfied they became.

A follow-up to Question 3 asked the participants to explain why they preferred the approach they chose. One reason commonly given by participants for their preference was that the approach they preferred allowed for greater speed and/or efficiency in performing the queries. Interestingly, three participants who preferred the parallel approach stated speed/efficiency as a reason for the preference, and three participants who preferred the simultaneous approach did as well.

A few other reasons were given by participants for why they preferred the parallel approach over the simultaneous approach. Four of the nine participants felt it was easier to build the queries in parallel because there was no worry about interrupting their collaborators or “stepping on each others’ toes.” Even two of the participants who preferred the simultaneous approach acknowledged this as an issue when explaining their preference. Three participants, each of whom preferred the parallel approach, stated that they felt more comfortable working independently in their private workspaces and sharing only what they wanted to share with their collaborators. One participant wrote “[the private workspace] allowed me to make mistakes in building my queries before showing it to my partner,” which is an often cited reason for why private workspaces are necessary in synchronous collaboration environments (Armstrong, 1993; Convertino, Ganoe, Schafer, Yost, & Carroll, 2005; Greenberg, Boyle, & Laberge, 1999). Table 5 summarizes the number of participants that stated each of the three main reasons for why they preferred the parallel approach over the simultaneous approach, out of the nine who did.
Along with the opinion that the simultaneous approach was more efficient than the parallel approach, there were two other main reasons given for preference of the simultaneous approach. The most common was that participants liked being able to instantly see what their collaborators were doing, and conversely they liked to be able to show their collaborators their input while they were talking about it. Another benefit to being able to see the input of their collaborator, as mentioned by a participant is that they did not need to feel “worried that they may unnecessarily duplicate work, or leave something out,” as they did with the parallel approach. The second main reason the simultaneous approach was preferred was that the users had instant access to the data layers as they were retrieved, and it was easier to keep track of which layers were which. The numbers of participants who gave each of these three main reasons for why they preferred the simultaneous approach are shown in Table 6.
Table 5. Number of participants, out of 9, who stated the given reasons for why they preferred the parallel approach over the simultaneous approach. Participants could provide more than one reason.

<table>
<thead>
<tr>
<th>Reason for preferring the parallel approach</th>
<th>Number of participants (out of 9 total)</th>
</tr>
</thead>
<tbody>
<tr>
<td>It was faster and/or more efficient.</td>
<td>3</td>
</tr>
<tr>
<td>It eliminated the possibility of interrupting or being delayed by their collaborator.</td>
<td>4</td>
</tr>
<tr>
<td>It provided a more comfortable working environment with the private workspace.</td>
<td>3</td>
</tr>
</tbody>
</table>

Table 6. Number of participants, out of 7, who stated the given reasons for why they preferred the simultaneous approach over the parallel approach. Participants could provide more than one reason.

<table>
<thead>
<tr>
<th>Reason for preferring the simultaneous approach</th>
<th>Number of participants (out of 7 total)</th>
</tr>
</thead>
<tbody>
<tr>
<td>It was faster and/or more efficient.</td>
<td>3</td>
</tr>
<tr>
<td>Participants could see their collaborator’s interactions and input in real-time.</td>
<td>5</td>
</tr>
<tr>
<td>It provided instant access to all of the data layers as they were retrieved, making them more convenient to use and easier to keep track of.</td>
<td>2</td>
</tr>
</tbody>
</table>
4.2.2 Critical incidents

Two of the questions in the survey asked the participants to identify any specific incidents when the parallel or the simultaneous query building procedures were confusing or difficult. If there were any such incidents for either approach, they were asked to describe the incidents and explain how they dealt with them. The following two subsections discuss the responses to this question for the parallel approach and the simultaneous approach, respectively.

Parallel approach

For the question regarding critical incidents with the parallel approach, many participants gave no response, or mentioned something positive such as “it was very useful.” Of the responses that did mention an aspect of the parallel approach that was confusing, the most common point was that it was difficult to closely interact with the other collaborator without being able to see everything they were doing. In fact, comments along this line were made by five of the seven participants who preferred the simultaneous approach. A different critical incident mentioned by a participant who preferred the simultaneous approach focused on the lack of the ability to drag a layer from their private tab into the public tab. This participant said it would have been helpful to have been able to use the drag and drop metaphor to share his layers instead of right-clicking and clicking ‘Share this layer.’ The screen recording of this session shows the participant clicking on a layer and attempting to drag it down to the public layer list several different times.
Of the nine participants who preferred the parallel approach, only one made a specific comment about critical incidents. He said that he was confused by the fact that there was no query editor in the public tab of the interface. This is just the result of a misunderstanding of how the user study was designed, as the tutorial video explains that for the study there is no query editor in the public workspace.

**Simultaneous approach**

By far the most common point of confusion or frustration with the simultaneous approach was the fact that there was only a single query editor that had to be shared by the participants. Nine of the participants responded with something similar (six of which preferred the parallel approach, three preferred the simultaneous approach). They were typically either annoyed that they had to wait for their collaborator to finish what they were doing, or they were confused about what could and could not be done without interrupting what the other person was doing. One critical incident that occurred along these lines, which was mentioned in the response to this question by one of the participants, was that one person attempted to start a new query while their collaborator was in the middle of constructing one. The action of starting a new query in the shared query editor eliminated all of the input of the other user, as confirmed by the screen recording.

Another confusion encountered by one participant was about who would lead a task in the simultaneous approach. Obviously one person has to take the first step toward
constructing a new query, and there is more potential for apprehension about who should take that initial leadership role than there is in the parallel approach.

A limitation in the software design was brought up by one participant as a point of confusion or frustration. While most of the interactions are shared between the users connected to SQSynC in the simultaneous approach, the options that expand when a dropdown element is activated do not get displayed to anyone other than the person interacting with the element. The participant who noted this rightly points out that this is a flaw of the software (rather than a limitation of the approach), since it does not enable both collaborators to simultaneously view the options of a dropdown element, such as the values that are available for a selected attribute field. Instead, they only see that the dropdown element is locked to the other participant and they must wait before they can open it to view the values.

4.2.3 Enjoyable features

There were also two questions in the survey prompting the participants to identify and describe any features of either of the two approaches that they particularly enjoyed. One feature that is worth pointing out as an observed result, but that was not mentioned by any participants, is the audio communication feature. Every participant used audio communication throughout each session, while the text communication was not used once. The next subsection discusses the responses about enjoyable features of the parallel approach, and the following subsection looks at the responses for the simultaneous approach.
**Parallel approach**

One interesting response to this question about particular features of the parallel approach that the participants enjoyed was about the ability to join two data layers. One participant said “I liked that you were able to join layers if necessary.” What is interesting about this is that the join layer functionality in the public tab of the parallel mode of SQSynC was not used once by any of the 16 participants. It is possible that this participant saw the potential value of this functionality, but never felt the need to use it for any of the tasks in the user study. It is also possible that the user was referring to the ability to use the query editor to spatially join two different data layers.

Aside from the response about joining layers, there was not much new information provided in the responses to this question. The participants generally reiterated points that they had made previously, such as how they liked to have a defined, private workspace that only they could see; they did not have to worry about interrupting their collaborator and they could “experiment and be wrong without anyone seeing it,” as one participant put it.

**Simultaneous approach**

Again with this question about enjoyable features of the simultaneous approach, there were a lot of comments that were similar to those made in previous parts of the survey. For instance, several participants again mentioned how they liked being able to see their collaborators’ actions. One participant responded “I liked being able to see what
my partner was doing as if I was sitting next to them,” which is a feature to strive for in any same-time, different-place collaboration application.

One new point that was raised by three different participants in their responses to this question was that it was easier to accomplish tasks in fewer, more complex queries, which lends some support to the argument that the simultaneous approach is more efficient. Two of these three respondents preferred the parallel approach to the simultaneous approach overall.

Another feature of the simultaneous approach that was commented on in a couple of the responses to this question was the concurrency control functionality. Two participants specifically said that they liked how elements would get locked when their collaborators were interacting with them.

One comment stood out as an interesting counterpoint to the perspective that it is better to have a private workspace so that mistakes can be made without others knowing. A participant said “if I was running the query my partner would be able to see me doing it and correct any mistakes I might be making.”

4.2.4 Suggestions for improvements and other scenarios

The final two questions in the survey asked the participants to think about how the collaborative query building procedure in general could potentially be improved, and solicited their opinions about other real-world scenarios where a tool like SQSync would be useful.
One suggestion made by a participant focused on the placement of the query editor within the interface. The decision was made to have the query editor be collapsible so that it could be mostly hidden when it was not in use. When the query editor is expanded, it covers up a portion of the right side of the map, though it does not interfere with the functionality of the map, such as panning, zooming, and toggling data layers. The participant suggested that the query editor should not block a portion the map, so that the users can see their query results on the map with each query they perform. Another participant made the suggestion to include the ability to take snapshots of the work done in SQSynC at any time in order to document the process, as well as the ability to save the work. A third suggestion was that users should be able to retrieve previously built queries and edit them using the query editor. The remaining 13 participants had no new suggestions to add for how to improve the collaborative query building process.

There were several suggestions made about other real-world scenarios where collaborative spatial query tools like SQSynC could potentially be necessary or useful. Emergency response and management was the most commonly mentioned type of scenario; 10 of the 16 participants included some type of emergency situation in their responses. Other scenarios mentioned were inter-department collaboration within government and industry, event coordination, defense applications, scientific fieldwork research, mission control centers for spacecraft, national park management, wildlife rescue, package delivery and tracking, and traffic accident analysis. More generally, several participants commented on the potential usefulness of SQSynC in any project involving geographic data where people in different parts of the world need to
collaborate. Two participants also commented on the usefulness of enabling experts to assist nonexperts in completing a task.

In summary, this chapter has presented the results of the user study, which are mostly from the responses given by the participants in the follow-up survey. It was found that 56% of participants preferred the parallel mode of collaborative query building, while 44% preferred the simultaneous mode. Various opinions were given for why one approach was preferred over the other, and there were also several comments about critical incidents and enjoyable features of each approach. Finally, suggestions for how support for spatial query in same-time, different-place geocollaboration could be improved and suggestions for potential real-world applications for tools like SQSynC were given. The following chapter discusses all of these results and concludes with discussion about potential ways to continue with this research in the future.
Chapter 5

CONCLUSION

The main goals of this research were to (a) develop a conceptual framework and create and assess a proof-of-concept software prototype for supporting spatial queries in a same-time, different-place geocollaboration environment and (b) compare two different approaches to collaborative spatial data query; a parallel approach and a simultaneous approach. The following section summarizes and discusses the results of the project in the frame of these two goals. The final section presents ideas for future work to carry on this line of research.

5.1 Discussion

There were several interesting results from this research, which are discussed in this section. In section 5.1.1 the conceptual framework and software prototype are discussed. The communication support of SQSynC is discussed in 5.1.2, followed by the concurrency control functionality in 5.1.3. Section 5.1.4 discusses features that participants felt were missing from SQSynC, and finally the satisfaction with the parallel approach versus that of the simultaneous approach is discussed in 5.1.5.
5.1.1 Conceptual framework and prototype

The first phase of this project was to develop a conceptual framework for same-time, different-place geocollaboration with support for spatial data queries. Derived from a comprehensive review of literature in the fields of CSCW, geocollaboration, and CIR, the conceptual framework outlines a set of core features required for a same-time, different-place geocollaboration environment in general and then extends this with a set of additional features specifically for supporting spatial query. The successful development of the SQSynC prototype shows that the software design based on this framework is feasible. Developing a web-based application with synchronous data sharing capabilities enabled workspace awareness functionality such as color-coded cursor tracking, whiteboard drawing, and WYSIWIS views in the public workspaces. The combination of these features resulted in a usable same-time, different-place geocollaboration system, evidenced by several positive comments by the participants.

5.1.2 Communication

The results of the user study also corroborated findings by Brewer et al. (2000) that audio is the preferred mode of communication in same-time, different-place geocollaboration, at least when the only alternative is text. The interaction logs of each session show that every single participant used the audio communication functionality frequently. Each pair of participants was talking throughout the entire session. The text communication feature, however, was not used by a single participant. This is not to say that text communication is completely useless in this environment; it would likely see
more use when there are more than two people collaborating. Multiple people can transmit text simultaneously, but several people talking at once does not work as well.

5.1.3 Concurrency control

The concurrency control model implemented in SQSynC turned out to work well, at least functionally. Locking individual elements of the interface to only enable control for a single user at a time successfully prevented any conflicts, aside from one issue with starting a new query before finishing the existing one (which clears the current query since there can only be one query at a time). Despite this mostly successful concurrency control, there were still nine comments made by participants about confusion or annoyance with the simultaneous query interface. These nine participants either did not know what they were able to interact with while their collaborator was interacting with the interface, or they felt that they had to wait until their collaborator was finished before being able to do anything, which annoyed them. It is possible that the concurrency model employed by SQSynC is not the most usable option and that another model would be better overall. It could also just be part of the learning curve that users would need to get over in order to successfully use the software (Ellis, Gibbs, & Rein, 1991).

5.1.4 Missing features

There were some features that were not incorporated into SQSynC that some participants expected to see included, but overall the prototype was still able to
successfully facilitate the tasks within both experimental scenarios. One participant expected to be able to drag layers from their private tab and drop them into the public tab as a way to share their layers, which is not possible in SQSynC. This added drag and drop functionality would certainly be an improvement to the system. Another item that was more clearly a flaw versus a missing feature was the inability for the participants to see the options of a dropdown element that another participant was interacting with. Beyond these issues, SQSynC functioned successfully as intended.

5.1.5 Parallel approach versus simultaneous approach

The fact that all but one of the participants were either satisfied or very satisfied with at least one of the collaborative query building approaches lends support to the conclusion that the software development phase of this project was a success. It is less straightforward, however, to compare the parallel query building approach to the simultaneous query building approach using the satisfaction ratings. The histogram in Figure 38 shows that there might be slightly stronger feelings one way or the other about the parallel approach since there were two more participants that were dissatisfied with it than the simultaneous approach and one more participant that was very satisfied. Overall, though, the sample size is not sufficient to assess the statistical significance of differences; thus it is not possible to make definitive conclusions about the relative preferences for the two approaches.
Along with the satisfaction ratings of the two approaches, the user study elicited additional valuable input from the participants. The results of the survey question about which of the two approaches the participants preferred show that there is no clear winner, and that in fact, each approach is preferred nearly equally among the 16 participants. Just two more participants preferred the parallel approach than preferred the simultaneous approach. With preferences for one approach over the other split nearly down the middle, the results may suggest that there is a place for both private and public query building tools in a same-time, different-place geocollaboration environment, so that the decision of which approach to use can be made by the user and they can even switch between the approaches if they so choose.

The ideal software design likely should include versions of both the parallel and the simultaneous approaches to supporting collaborative spatial queries that were
presented in this thesis. It has been shown that it is certainly feasible to support both options in a web-based application. The private workspace should include a query editor, and based on reactions from a few participants, I believe that the public workspace should also include a shared query editor that all of the users can see and interact with concurrently.

The responses to the final question of the survey show that there are a multitude of potential real-world applications for software like SQSynC. Many different suggestions were made by the participants, spanning a wide range of domains. There are people and organizations doing important work that stand to benefit from improved geocollaboration technologies.

5.2 Future work

Overall, SQSynC stands as a successful proof-of-concept implementation of the ideas presented in this thesis, and the user study resulted in interesting feedback from the participants. One of the main limitations of this research though was the small number of participants used for the user study. A sample of 16 total participants in eight sessions is not large enough to assess statistical significance of results, especially when the number of people who prefer each of the two approaches is so similar. While the results of this research show that there is potentially a place for both approaches implemented for spatial data query in same-time, different-place geocollaboration, a larger sample would enable a more reliable assessment of preferences and characterization of advantages and disadvantages of each approach to be made.
Another advantage to running a similar user study with a larger sample would be the ability to make additional comparisons between the two approaches. With more participants and longer sessions, comparisons could be made between the performance levels of users in each approach. Instead of limiting participants to 20 minutes to use each approach, they could be given a set of tasks that they must complete before moving on. In this way it would be possible to measure (a) the time taken to complete tasks and (b) the number of tasks completed successfully, so that time and accuracy could be compared between the two approaches. A larger sample size and longer sessions were not used for this user study due to funding and time limitations, as well as the availability of willing participants.

Along with running further user studies similar to the one presented here, another path for continued research is to continue to develop the SQSynC software and/or work on other, better tools to support same-time, different-place geocollaboration. The responses to the survey provide a good starting point. A couple of flaws in SQSynC were found and a few suggestions for how to improve the interface were made. The first functionality flaw that was found is that the dropdown menus of the interface elements do not follow the principle of WYSIWIS. This is an impediment to simultaneous collaborative query building process and a solution needs to be found in any future versions of the software. Another comment was made about the ability to drag and drop data layers from the private view to the public view as a way to share them. The screen recording shows that this user attempted this interaction multiple times throughout the session, suggesting that this is more intuitive than right-clicking and clicking ‘Share this layer’, at least for this particular user and likely other people as well. Since this
functionality provides added benefit to the user without adding any additional complexity like extra menus or buttons in the interface, it would be a good idea to include it in the future. A third issue was with the way a new query is initiated; one participant was initially confused that selecting a new table starts a new query and thus erases an existing query if it has not yet been submitted. Alternatives could be explored, such as displaying an alert window that a user must confirm that they want to clear their current query and start a new one, or including an extra button that the user must click to start a new query before they can choose the table to query.

Beyond the above, ideas that were submitted by participants about including the ability to take snapshots at any time to document the workflow, the option to save work so that it would be possible to return to it at a later time, and the ability to go back to a previously executed query and change its parameters in the query editor were all excellent suggestions. Finally, there was a concern about the position of the query editor within the interface; several different options for the layout of the interface could be explored. The best way forward to address these flaws and suggestions would be to implement solutions and then perform a far more in-depth usability study of the software.

Ultimately, the research presented here offers a good starting point for researching the problem of how to support spatial data queries in a same-time, different-place geocollaboration environment, by providing a conceptual framework, a proof-of-concept software prototype, and the results of a user study comparing two different approaches. It is now clear that (at least given the interface options introduced here) some people prefer to build spatial queries in parallel, in separate, private workspaces and then collaboratively iterate over their results, while others would rather simultaneously build
spatial queries in a WYSWIS environment. This difference in styles of work is similar to findings by Robinson (2008) on tendencies to adopt different organizational metaphors for information in collaborative, analytical tasks. The SQSynC prototype and the results found in this study can be built upon to explore further questions, such as which approach is more efficient, which leads to more accurate results, and how to support moving between approaches in real work in order to meet different and changing needs of a range of users.
REFERENCES


Harris, T., & Wiener, D. (1996). GIS and society: The social implications of how people, space, and environment are represented in GIS. *Scientific Report for the Initiative 19 Specialist Meeting* (pp. 96-97). NCGIA.


*Communications of the ACM*, 30 (1), 32-47.


APPENDIX A

Sample task given to participants to walk through during the tutorial video

Find available assisted-living homes in community-oriented neighborhoods.

| community-oriented neighborhoods are in the `front_porch` table | the type of assistance needed is for persons with disabilities |
| assisted-living homes are in the `assist_housing` table          | `pop_served`: describes the type of assistance provided |
| `occstatus`: field in the `assist_housing` table describes the availability | |


APPENDIX B

Task information sheet given to Participant 1 for the hypothetical emergency response scenario

Emergency Response
These tasks are based on a hypothetical scenario involving a hurricane in Florida, with one collaborator who works in local government and another who is an emergency response expert. The blue text beneath each task represents your domain knowledge and is required, along with your partner’s domain knowledge, to successfully complete the tasks. When you have completed the tasks, please submit them in the tasks interface and/or let the researcher know.

1. Find all medical centers within the critical area that provide general health care services and have a sufficient number of beds.
   - counties are in the counties table, medical centers are in the hospitals table
   - the critical area includes Broward, Martin, Miami-Dade, and Palm Beach counties (name field in the counties table)

2. Parts of this database are being kept up to date in real time. Which rescue personnel are within range of the critically injured victim and are capable of assisting with a spinal injury?
   - the critically injured victim is the spinal_injury table, the personnel are in the personnel table
   - the personnel must have a helicopter to air lift the injured party

3. Find all high-capacity evacuation routes that intersect the hurricane track.
   - the hurricane track is in the hurricane_track table, routes are in the roads table
   - sufficient capacity is greater than 2000 (capacity field)
   - the routes must be at least 2 lanes (lanes field)

4. Find all facilities housing unsafe amounts of dangerous types of hazardous materials within the critical area.
   - the critical area for hazardous materials is within 15 miles of the hurricane track
   - the hurricane track is in the hurricane_track table
   - unsafe amounts include anything above 50
   - dangerous types of materials include Class 1, Class 2, and Class 3
APPENDIX C

Task information sheet given to Participant 2 for the hypothetical emergency response scenario

Emergency Response
These tasks are based on a hypothetical scenario involving a hurricane in Florida, with one collaborator who works in local government and another who is an emergency response expert. The green text beneath each task represents your domain knowledge and is required, along with your partner’s domain knowledge, to successfully complete the tasks. When you have completed the tasks, please submit them in the tasks interface and/or let the researcher know.

1. Find all medical centers within the critical area that provide general health care services and have a sufficient number of beds.
   - counties are in the counties table, medical centers are in the hospitals table
   - general care means hospitals of type ACUTE CARE (type field)
   - sufficient number of beds is more than 200 (beds field)

2. Parts of this database are being kept up to date in real time. Which rescue personnel are within range of the critically injured victim and are capable of assisting with a spinal injury?
   - the critically injured victim is the spinal_injury table, the personnel are in the personnel table
   - appropriate range is 25 miles
   - the vehicle field in the personnel table describes the vehicle the person is currently using

3. Find all high-capacity evacuation routes that intersect the hurricane track.
   - the hurricane track is in the hurricane_track table, routes are in the roads table
   - the routes must be egress routes (egress field = true)

4. Find all facilities housing unsafe amounts of dangerous types of hazardous materials within the critical area.
   - hazmat facilities are in the hazmat_facilities table
   - facilities that function as storage or disposal are the ones currently housing materials (function field)
   - the amount field describes the amount of materials currently at the facility
   - the type field describes the type of materials currently at the facility
APPENDIX D

Task information sheet given to Participant 1 for the hypothetical real estate shopping scenario

Real Estate
These tasks are based on a hypothetical scenario of a real estate agent helping people search for real estate in Florida based on their needs and desires. The green text beneath each task represents your domain knowledge and is required, along with your partner’s domain knowledge, to successfully complete the tasks. When you have completed the tasks, please submit them in the tasks interface and/or let the researcher know.

1. Find affordable and appropriately sized waterfront property.
   - Property is in real_estate table, water bodies are in waterbodies table
   - Waterfront is within 1 mile of a water body (not including the ocean)
   - The water body feature cannot be a swamp or marsh (feature field)

2. Find affordable homes that are on a large beach.
   - Beaches are in beaches table, homes are in real_estate table
   - Large beaches have an area of at least 3,000,000 (area field in beaches table)

3. Find accessible homes, with a garage, near a church.
   - Homes are in real_estate table, churches are in religious_centers table
   - Accessible homes have no more than one floor (floors field)
   - Near a church means within 10 miles

4. Find large homes that are close to small schools.
   - Schools are in schools table, homes are in real_estate table
   - Small schools have an enrollment (enrollment field) of less than 500
   - Close means within 15 miles
Task information sheet given to Participant 2 for the hypothetical real estate shopping scenario

**Real Estate**

These tasks are based on a hypothetical scenario of a real estate agent helping people search for real estate in Florida based on their needs and desires. The blue text beneath each task represents your domain knowledge and is required, along with your partner's domain knowledge, to successfully complete the tasks. When you have completed the tasks, please submit them in the tasks interface and/or let the researcher know.

1. Find affordable and appropriately sized waterfront property.
   - the property should be empty land (type field)
   - the size should be at least 1 acre (acres field)
   - the price should be between $100,000 and $250,000

2. Find affordable homes that are on a large beach.
   - affordable is less than $355,000 (price field)
   - the type of home desired is a condo (type field)
   - the home should have a swimming pool (pool field)

3. Find accessible homes, with a garage, near a church.
   - the garage should fit at least 2 cars (garage field)
   - the church should be non-denominational (type field)

4. Find large homes that are close to small schools.
   - large homes:
     - are over 3000 square feet (sqft field)
     - have at least 4 bedrooms (bed field)
     - have at least 3 bathrooms (bath field)
     - school type should be private (op_class field)