

GIS-BASED ACCIDENT LOCATION AND ANALYSIS SYSTEM (GIS-ALAS)

PROJECT REPORT: PHASE II

CTRE Management Project 97-6

**Prepared by
Center for Transportation Research and Education
Iowa State University**

Iowa State University Research Park
2625 North Loop Drive, Suite 2100
Ames, IA 50010-8615
Telephone: 515-294-8103
Fax: 515-294-0467
<http://www.ctre.iastate.edu>

Submitted to
Office of Transportation Safety
Iowa Department of Transportation

Principal Investigator
Reg Souleyrette

Project Manager
Tim Strauss

Graduate Research Assistants
Brad Estochen
Michael Pawlovich

April 1999

| | |
|---|-----------|
| EXECUTIVE SUMMARY..... | V |
| INTRODUCTION..... | 1 |
| REFINEMENT AND ENHANCEMENT OF GIS-ALAS | 1 |
| ARCVIEW ALAS SUGGESTED IMPROVEMENTS LIST | 2 |
| WEB SITE DEVELOPMENT | 2 |
| ABBBC PROCEDURE | 3 |
| INDEPENDENT EVALUATION: BOB SCHULTZ..... | 3 |
| INDEPENDENT EVALUATION: CLAY SPINUZZI..... | 3 |
| ACCIDENT RATE METHODS ASSESSMENT | 3 |
| TRANSLATOR TOOL DEVELOPMENT | 4 |
| ENHANCED LINK-NODE TO LATITUDE-LONGITUDE CONVERSION..... | 5 |
| “AVENUE OF SAINTS” EMERGENCY RESPONSE PATTERN CHANGES..... | 5 |
| BACKGROUND:..... | 6 |
| METHODS: | 6 |
| PROCEDURE/PRODUCTS:..... | 6 |
| FUTURE PATHS: | 7 |
| TECHNOLOGY TRANSFER | 8 |
| DOCUMENTS: | 8 |
| PRESENTATIONS:..... | 9 |
| RELATED PROJECTS | 10 |
| CTRE PROJECTS | 10 |
| IOWA DOT PROJECTS..... | 12 |
| FUTURE ACTIVITIES | 13 |
| APPENDIX A | 14 |
| ARCVIEW ALAS SUGGESTED IMPROVEMENTS LIST | 14 |
| APPENDIX B | 20 |
| EXPLORER ALAS DATA FILE CREATION (ABBBC) PROCEDURE | 20 |
| APPENDIX C | 23 |
| INDEPENDENT EVALUATION: BOB SCHULTZ..... | 23 |
| APPENDIX D..... | 25 |
| RESULTS OF GIS-ALAS USEABILITY RESEARCH..... | 25 |
| APPENDIX E | 31 |
| CTRE LINK-NODE TO LATITUDE-LONGITUDE (MPT2SHP) PROCESS | 31 |

EXECUTIVE SUMMARY

This report summarizes progress made in Phase II of the GIS-based Accident Location and Analysis System (GIS-ALAS) project. Phase II efforts build on Phase I activities, which were focused on improving the current PC-based Accident Location and Analysis System (PC-ALAS) used by the Iowa Department of Transportation (Iowa DOT); these improvements were realized through the development of a GIS-based system that facilitates advanced spatial query and display capabilities in a graphical environment. Phase II efforts then focused on the refinement and enhancement of GIS-ALAS in both Explorer ALAS and ArcView ALAS, and on the use of GIS-ALAS in specific applications. This report documents several Phase II activities including the refinement and enhancement of GIS-ALAS using a list of suggested improvements; changes in the data conversion processes; a specific case study; and several technology transfer efforts. Several appendices with more detailed information are provided at the end of the full report.

INTRODUCTION

This report summarizes progress made in Phase II of the GIS-based Accident Location and Analysis System (GIS-ALAS) project. Phase II efforts build on progress made in Phase I to improve the existing accident location and analysis system (PC-ALAS) used at the Iowa Department of Transportation (Iowa DOT).

In Phase I, efforts were focused on reproducing most of the query and reporting functions of PC-ALAS, but in a GIS-based graphical environment that facilitates advanced spatial query and display capabilities. During Phase I, most of the base system was created. Phase II efforts were directed toward refining and enhancing the core GIS-ALAS product developed in Phase I, as well as exploring the use of GIS-ALAS for specific analyses.

Phase II of the GIS-ALAS effort is documented in this report. GIS-ALAS refinement and enhancement is discussed, including changes for both Explorer ALAS and ArcView ALAS. A list of suggested enhancements is provided, and implemented improvements are explained. The improvements are prioritized and those with highest priority are listed separately. Changes in the data conversion processes are discussed, including both a new method to convert link-node to latitude-longitude and the development of a translator from latitude-longitude to link-node. This translator allows future use of PC-ALAS and Microsoft Access-ALAS. Also, a specific case study is discussed. This case study examines changes in emergency service response patterns along the “Avenue of Saints” due to corridor construction and realignment. Several technology transfer efforts are then outlined, including a user’s manual for ArcView ALAS, a web site, and several related papers and presentations. Finally, related projects that support, or are supported by, GIS-ALAS are briefly discussed. These include the Location Tool/SmartMap effort, the Crash Outcome DataEvaluation System (CODES) project to evaluate injury outcomes through the linking of medical and crash data, and the analysis of access management, winter weather safety, and deer crashes. In addition, several appendices with more detailed documentation are provided at the end of the full report.

REFINEMENT AND ENHANCEMENT OF GIS-ALAS

Task 1 of the GIS-ALAS Phase II proposal states that “the interface will be improved based on input from initial users.” In Phase II, suggested improvements were requested from several ALAS users including personnel at the Iowa DOT’s Office of Transportation Safety; planning, engineering, and law enforcement personnel; Bob Schultz, an independent consultant working with the Office of Transportation Safety; CTRE GIS-ALAS team members; and Clay Spinuzzi, an Iowa State University graduate student in rhetoric and professional communication doing research on computer system usability issues. The suggested improvements identified thus far have been categorized and are listed below.

Recently, copies of ArcView and ArcView ALAS have been distributed to additional users and a formal beta test is currently underway under the Phase III contract. It is

anticipated that ideas for further improvements will be identified during the formal beta test.

ArcView ALAS Suggested Improvements List

Throughout the development and enhancement of GIS-ALAS, a number of suggested improvements have been identified, some by the team and some by independent evaluators. These suggested improvements have been compiled into a list, shown in Appendix A. Each of these improvements has been classified into a number of categories regarding the interface, reports, printing, etc. Each item in the list also has an indication of whether it has been completed or, if not, a priority level. Additional items may at times be added to this list, whether through the course of further development and enhancement or through the beta test stage. The priority levels may change based on these additions or shifting emphasis as indicated by users. Major enhancements completed include several relating to the user interface, such as speed and appearance, and to the functionality, such as types of searches and reports. High-priority items remaining include (listed by category):

- Interface – User-friendliness
 - Allow users to start afresh with selection and queries without closing AV-ALAS.
 - Provide a back button for parameter selections.
 - Provide functional button dialogs.
- Query
 - Enable non-functional query types.
 - Address county-wide search issues.
- Report
 - Insert refinement parameters selected into report.
 - Fill in crash rate.
 - Insert date, region, etc. into report for identification.
- Printing
 - Enable report printing.
- Documentation
 - Develop online help, training programs and manuals, and documents detailing current use and analyses.
- Miscellaneous
 - Develop Intersection Magic interface.
 - Enable statewide operation.

Web Site Development

The web site has been developed and the current site address is:

http://www.ctre.iastate.edu/projects/tp_is/AV-ALAS_BetaTest/betatest.htm

Currently the page has the latest User's Manual (in Microsoft Word 97 format), a download choice for the most current ArcView ALAS project file, and a download choice for the installation modules. Potential additions to the web site include the user's

manual in online form, examples of how GIS-ALAS is being used, and answers to commonly asked questions, as well as the most recent map/data downloads. It may someday include case studies in which users relate how they have dealt with certain situations.

ABBBC Procedure

The crash database has three separate record formats, one for general crash information (A Record), one for driver and vehicle information (B Record), and one for injury information (C Record). ArcView ALAS has been coded to facilitate searches among each of these record formats; however the separate formats are difficult to use with ArcExplorer. To enhance Arc Explorer usage, the data obtained from the Office of Transportation Safety, and subsequently attached to features using the MPT2SHP procedure (outlined in Appendix E), were run through a process to create yearly aggregate files for each county containing one A record, up to three B records, and one C record for each crash. This enables users of Explorer ALAS to select crashes graphically and obtain, in almost all cases, all the information about the crashes selected. The ABBBC procedure is outlined in Appendix B.

Independent Evaluation: Bob Schultz

Bob Schultz, an independent contractor for the Office of Transportation Safety, beta tested ArcView ALAS and came up with several suggestions, most of which have been incorporated into the list in Appendix A. The suggestions in the list primarily focused on the user interface and query functionality. The list of suggestions itself is in Appendix C.

Independent Evaluation: Clay Spinuzzi

Clay Spinuzzi, an Iowa State University graduate student in rhetoric and professional communication, also beta tested ArcView ALAS. The focus of the evaluation was on computer system usability issues. Several developers and potential users were surveyed and a group of Iowa State students was observed using the system. He sent a memorandum concerning his study and findings to CTRE staff. The last section of this memorandum lists his suggestions; these suggestions have been incorporated into the list in Appendix A. The memorandum itself can be found in Appendix D.

ACCIDENT RATE METHODS ASSESSMENT

The accident rate methods assessment was to examine ways to identify crash rates for highway sections or intersections based on specific crash locations and traffic counts. The Phase II proposal regarding this section states that “the usefulness of accident point locations of GIS-ALAS and estimates of average daily traffic (ADT) variations along a segment in providing finer-scale analyses of accidents will be assessed.” However, in consultation with the project monitor, this task was postponed until Phase III to allow more pressing concerns, such as the translator tool (discussed in the next section), to be addressed.

TRANSLATOR TOOL DEVELOPMENT

Development of a tool to translate latitude-longitude coordinates for crashes into the ALAS link-node system is necessary for future operation of PC-ALAS and MS Access-ALAS, neither of which have been coded to work with latitude-longitude. This task replaced the accident rate methods assessment. A procedure was developed within ArcView utilizing the Avenue coding language to facilitate this translation.

The X,Y to Link-Node translator tool (or XY2LN procedure) was developed to facilitate the continued use of Access-ALAS and PC-ALAS as Iowa moves toward collection of x,y coordinates for its crash location process. The tool was designed to take crashes located by x,y coordinates and represented in a GIS coverage and determine the nodes with which these crashes are associated. Both Access-ALAS and PC-ALAS utilize these nodes to identify crashes when a user specifies a location.

Code was written, working from the assumption that in the future, crashes would be located by enforcement personnel using a combination of GPS receivers and the location tool/smart map in conjunction with Advantage Safety. Other assumptions regarding the data were that an x,y location as well as a route number would be provided.

The code, written in ArcView's Avenue scripting language, automatically determines the valid nodes for each crash, both for node crashes and link crashes. For node crashes, the code determines the nearest node to the crash and assigns that node to the crash attributes. For link crashes, the code initially selects from the nodes file all nodes that fall along the route indicated by the crash. The code then determines the nearest node from that node selection to the crash and assigns that node to the crash attributes. The code then determines the next closest node in the opposite direction from the closest node and assigns that node to the crash attributes as well. The code finally determines the distance from the nearest node and assigns that value to the crash attributes.

The process was tested, utilizing the code, crash feature files obtained from the Iowa DOT Office of Transportation Data, and crash attributes obtained from the Iowa DOT Office of Transportation Safety that had been brought into a GIS format by CTRE. To test the code, first the crash feature files and crash attributes had to be merged. This was done for years 1987 and 1995 for Johnson County using the GIS join function after key fields had been produced in both datasets. The code was then run on the resultant database utilizing a variety of tolerance distances, and the results were analyzed.

Results varied between the node crashes and the link crashes. For the node crashes, the capture rate (success rate) was in the upper 90 percent range. Two types of error were identified: incorrect node placement and incorrect crash placement. Incorrect node placement occurs primarily on primary roads within city boundaries. Upon closer examination, the crashes which had incorrect nodes identified were not directly on top of their nodes, due to an error in the database. The nodes are offset from the intersection, whereas the crashes are placed on the intersection by the Office of Transportation Data

method. The result of this incorrect node placement is that sometimes the crash is closer to another node than to the correct node. Incorrect crash placement, when it occurs, has the same result.

For the link crashes, the capture rate was much lower, varying between 50 and 70 percent. This was largely dependent on the tolerances chosen for buffering the crashes and roadways. The smaller the tolerances, the smaller the capture rate. However, smaller tolerances result in faster operation of the code and reduced chance of Type I errors (i.e., capture of incorrect nodes). Currently, researchers are working to quantify the different types of errors in order to refine the process in the future.

Errors currently occur due both to database issues and code issues. Database issues include roadway layer problems such as having only a centerline for divided highways, road names varying along the length of a road, and the presence of curves. Code issues include the setting of tolerances to minimize Type I errors and the consideration of curved road segments and ramps. Consideration of these issues is imperative to improving the translator code.

The process, although not perfect, can yield improved efficiencies in the Iowa DOT's crash data management procedures. Significant reduction in effort by the "locators" in the Iowa DOT Motor Vehicle Division can already be realized simply because the node crash success rate is high and node crashes represent roughly 50 percent of crashes statewide. With the addition of the link crash success rate, over half the effort involved in crash location currently will be handled by the translator tool. Additionally, because the crashes will be represented as features, their spatial locations will be available on map layers which will allow "locators" to visually determine or check the nodes associated with the crashes. This represents an improvement over past methods in which the locator must first determine the location using sometimes obscure literal descriptions, find the location on a map, and assign nodes based on that location.

ENHANCED LINK-NODE TO LATITUDE-LONGITUDE CONVERSION

This task addressed the problem of locating accidents on curved line segments and missing links. Initially, CTRE was to rewrite its MPT2SHP process (detailed in Appendix E). However, the Iowa DOT Office of Transportation Data determined that it could effectively and efficiently convert the data using the GIS package (MGE) at its disposal. Therefore, a process named "walking-the-line" was developed and is currently being refined for accuracy.

"AVENUE OF SAINTS" EMERGENCY RESPONSE PATTERN CHANGES

The "Avenue of Saints" task assessed changes in emergency response patterns due to the construction of the "Avenue of Saints" corridor in eastern Iowa. CTRE worked with the ISU Fire Services Institute to assess the impacts of Highway 218 corridor relocation and expansion in the Avenue of Saints. A synopsis of efforts follows:

Background:

The Iowa Fire Service Institute was interested in creating various maps of the fire service districts for Henry, Johnson, Lee, and Washington counties in southeast Iowa. These maps would be used for several purposes including: 1) providing a visual reference of the boundaries of each fire service district with respect to the adjoining fire service districts, 2) demonstrating how this visual information may be used in the emergency response planning efforts while Highway 218 is undergoing a realignment, and 3) introducing various ways of providing information about the local contacts for each fire district.

The Iowa Fire Service Institute was also evaluating the interaction between the local and state entities in the planning and construction of this corridor. This interaction is not part of the GIS-ALAS Phase II efforts. The major emphasis of this project was the creation of these maps in a GIS environment and integration of the data elements with the current data available within the GIS-ALAS project.

Methods:

Maps of each of the four counties' fire service districts were presented to the GIS-ALAS team. The maps varied from computer generated E911 maps with the service districts highlighted to rough photocopies with approximate boundaries sketched in.

Procedure/Products:

The fire service districts were created within ArcView. On average each county took two hours to complete. Using the roadway network provided from the Iowa DOT as a basis, along with the four maps of the counties (see Figure 1), polygon coverages were created for each fire service district. A short period of time was spent cleaning these polygon coverages. Based on input from George Oster and his team of investigators, researchers determined that the concept and a rough visualization would be sufficient for this stage of the process. There are areas between polygon coverages that were not assigned to a fire district. These appear as white slivers when displayed on a piece of paper.

Very little attribute data was added to these polygon coverages. The only attribute added to this data set was the fire district name. Several of these maps were printed out on a color plotter. These plots were 48 inches by 36 inches with various colors to represent each fire service district. One set of maps was produced for the Fire Service Institute team to use as a display. A set of 15 maps was created for each county. These were given to Jackie Comito, a team member of the Fire Service Institute team, who was responsible for dispersing them to the appropriate people within each county. Four sets of maps for Henry, Lee, and Washington counties were created and mailed to Larry Jackson, an engineer with the Iowa DOT Southeast Iowa Transportation Center, at the request of George Oster. Overall, 72 of these maps were created on the color plotter.

TECHNOLOGY TRANSFER

Several documents related to the GIS-ALAS project, including this report, were produced over the last year. Many of these documents are journal papers and all are listed below:

Documents:

- GIS-ALAS Users' Guide Version 1.0, Center for Transportation Research and Education (CTRE), Sept. 1, 1998, 74 pages.
- Pawlovich. "A Typology for Studying Crash Dependence on Demographic and Socioeconomic Data," Compendium of Student Papers presented at the 1998 Transportation Scholar's Conference, Mid-America Transportation Center/Center for Transportation Research and Education, Nov. 20, 1998.
- Estochen. "An Assessment of Emergency Response Vehicle Pre-Deployment Using GIS Identification of High-Accident Density Locations," Compendium of Student Papers presented at the 1998 Transportation Scholar's Conference, Mid-America Transportation Center/Center for Transportation Research and Education, Nov. 20, 1998.
- Pawlovich, Strauss, and Estochen. "The Application of Geographic Information Systems to Improve Rural Highway Safety", *Rural Advanced Technology and Transportation Systems 1998 International Conference Proceedings*, University Park, PA, August 1998, CD-ROM.
- Pawlovich, Souleyrette, and Strauss. "A Typology for Studying Crash Dependence on Demographic and Socioeconomic Data," *Crossroads 2000 Conference Proceedings*, Iowa Department of Transportation/Center for Transportation Research and Education, Aug. 19–20, 1998, pp. 209–215. (see <http://www.ctre.iastate.edu/pubs/>)
- Estochen, Strauss, and Souleyrette. "An Assessment of Emergency Response Vehicle Pre-Deployment Using GIS Identification of High-Accident Density Locations," *Crossroads 2000 Conference Proceedings*, Iowa Department of Transportation/Center for Transportation Research and Education, Aug. 19–20, 1998, pp. 221–226. (see <http://www.ctre.iastate.edu/pubs/>)
- Souleyrette, Strauss, Pawlovich, and Estochen. "Safety Data in a GIS Environment: New Tools for the Four Es," *Proceedings of the 24th International Forum on Traffic Records and Highway Information Systems*, Minneapolis, MN, July 26–28, 1998.
- Souleyrette, Strauss, Pawlovich, and Estochen. "GIS-ALAS, The Integration and Analysis of Highway Crash Data in a GIS Environment," *AASHTO GIS Conference Proceedings*, Salt Lake City, Utah, April 1998.

- Souleyrette, Strauss, Pawlovich, and Estochen. GIS-based Accident Location and Analysis System (GIS-ALAS) Project report: Phase 1, Iowa Department of Transportation/Center for Transportation Research and Education, April 1998.

Additionally, several presentations related to the GIS-ALAS project were made over the last year. These presentations are listed below:

Presentations:

- *Crash Analysis Using GIS-ALAS and Integration with Intersection Magic*, Iowa Traffic Engineering and Traffic Safety Forum, December 9, 1998.
- *A Typology for Studying Crash Dependence on Demographic and Socioeconomic Data*, 1998 Transportation Scholar's Conference, Mid-America Transportation Center/Center for Transportation Research and Education, Nov. 20, 1998.
- *An Assessment of Emergency Response Vehicle Pre-Deployment Using GIS Identification of High-Accident Density Locations*, 1998 Transportation Scholar's Conference, Mid-America Transportation Center/Center for Transportation Research and Education, Nov. 20, 1998.
- *Geographic Information System-based Accident Location and Analysis System*, 1998 Fall Iowa Traffic Control and Safety Conference (ITSCA), Ames, Iowa, October 22–23, 1998.
- *The Application of Geographic Information Systems to Improve Rural Highway Safety*, Rural Advanced Technology and Transportation Systems 1998 International Conference, University Park, PA, Aug. 30–Sept. 2, 1998.
- *A Typology for Studying Crash Dependence on Demographic and Socioeconomic Data*, Crossroads 2000 Conference, Iowa Department of Transportation/Center for Transportation Research and Education, Scheman Building, ISU Center, Aug. 20, 1998.
- *An Assessment of Emergency Response Vehicle Pre-Deployment Using GIS Identification of High-Accident Density Locations*, Crossroads 2000 Conference, Iowa Department of Transportation/Center for Transportation Research and Education, Scheman Building, ISU Center, Aug. 20, 1998.
- *Safety Data in a GIS Environment: New Tools for the Four Es*, 24th International Forum on Traffic Records and Highway Information Systems, Minneapolis, MN, July 26–28, 1998.
- *GIS-ALAS, The Integration and Analysis of Highway Crash Data in a GIS Environment*, AASHTO GIS Conference Proceedings, Salt Lake City, Utah, April 1998.

RELATED PROJECTS

Several projects either related to the GIS-ALAS project or using crash data are ongoing or have been recently completed. These include CTRE projects conducted for the Iowa DOT as well as Iowa DOT projects.

CTRE Projects

Location Tool/Smart Map project

CTRE is developing the SmartMap Crash Location Tool for the Motor Vehicle Division. This software program is being integrated into Advantage Safety (the next generation of the Officer Information Manager), a computerized data entry system used to collect crash data. The location tool will be used by law enforcement personnel, both in police vehicles and in office settings, and in other agencies to capture GIS crash location data. With SmartMap, a county map will appear during the Advantage Safety data entry process. Through a simple user interface, the user will zoom in to the approximate area and input the specific crash location with the click of a mouse (or the touch of a pen on the screen). The x,y location, the route identifier, and other relevant information will then be transferred to the main Advantage Safety module and incorporated into the crash database. This information will be available for immediate use in GIS by local agencies, and the statewide data management process at the Office of Driver Services will be greatly simplified.

(taken from the Location Tool, Phase II proposal, CTRE)

Crash Outcome Data Evaluation System (CODES)

The CODES project will design, develop, and implement a Crash Outcome Data Evaluation System for the identification, collection, and evaluation of traumatic injuries in Iowa resulting from motor vehicle crashes. By linking motor vehicle crash, ambulance, and health care records into one record, isolated pieces of information are joined together into a more complete picture. Crash and occupant variables will be related to injury outcomes, allowing assessment of the impact of such risk factors as alcohol impairment or lack of safety belt use on injury type and inpatient charges. Though crash records provide an estimate of injury severity, only health records capture the location of the injury, the hospital charges, the expected payment source, and the discharge disposition. Conversely, the motor vehicle crash records provide a wealth of information about the nature of the crash and the risk factors for being injured.

The Center for Transportation Research and Education at Iowa State University and the University of Iowa Injury Prevention and Research Center will assist in the analysis and application of the linked data to:

- Determine acute care charges and severity of injury for use vs. non-use of safety equipment in motor vehicle crashes
- Determine acute care charges and severity of injury for helmeted vs. non-helmeted crashes involving slow moving vehicles, such as farm machinery
- Determine acute care charges and severity of injury among different motor vehicle crash speeds
- Determine the validity and reliability of recorded personal identifiers in the traffic records
- Provide a research database to communities, EMS regions, EMS agencies, state agencies and research institutes to monitor and evaluate highway safety programs

The goal of CODES is to provide detailed, reliable, and readily accessible information on a motor vehicle crash victim. These data will allow assessment of prevention activities, quality of care, highway safety system evaluation, and continued research in the area of motor vehicle crashes. Information obtained from the CODES data will enable the state to approach the legislature and other governmental bodies in more meaningful ways as we develop laws and public policies dealing with motor vehicle crash injuries.

The Bureau of EMS will coordinate the CODES grant in collaboration with the Governor's Traffic Safety Bureau; Iowa DOT, Division of Driver Services; Association of Iowa Hospitals and Health Systems; Center for Transportation Research and Education at Iowa State University; and the University of Iowa Injury Prevention & Research Center.

(taken from "Iowa EMS Announces CODES Grant" by Richard Harmon, EMS Coordinator, Iowa Department of Public Health)

Access Management

Access management is a process that manages access to land development while seeking to preserve the flow of traffic on the surrounding road system. Sound access management practices can lead to safer roads that provide better service to motorists. In 1996, the Iowa Department of Transportation established an Access Management Task Force as part of its Safety Management System development effort. The Task Force is working with the Center for Transportation Research and Education at Iowa State University and other researchers at the University of Northern Iowa to improve awareness of how access management can lead to safer,

better functioning highways, roads, and streets in Iowa. The Task Force and Research Team are also investigating the effects that access management projects have on local business vitality.

(taken from CTRE's web site)

Mobility and Safety Impacts of Winter Storm Events in a Freeway Environment

The safety and traffic-flow service qualities of a freeway segment are a function of its prevailing conditions. Nationally accepted procedures for freeway level of service (LOS) and mobility analyses currently account for the impact of prevailing traffic and roadway characteristics (e.g., heavy vehicles, lane width, etc.). Good weather, adequate pavement conditions, and an incident-free environment are assumed. Unfortunately, there are many time periods of varying length in which these assumptions are incorrect, and the mobility and safety of a freeway segment are compromised.

A more comprehensive knowledge of how poor weather, snow, and ice events in particular, impacts the safety and service provided by a freeway will improve the decision-making capabilities of the traveler and those responsible for roadway operations and maintenance.

(taken from the Mobility and Safety Impacts of Winter Storm Events in a Freeway Environment proposal, CTRE)

Iowa DOT Projects

MS Access-ALAS

MS Access-ALAS is an effort to update the Iowa DOT PC-based Accident Location and Analysis System (PC-ALAS) by coding the PC-ALAS capabilities into MS Access. Access-ALAS is currently in the beta test stages and will be distributed soon.

MARS/OIM and Advantage Safety

The Officer Information Manager (OIM) automates the completion of driver and vehicle related reports by law enforcement officers in the field using pen-based computers. Software has been developed for OIM to provide the computer interface through which crash information is recorded. This software, the Mobile Accident Reporting System (MARS), supports the link-node location of crashes that has been historically used in Iowa. It is designed to support the development of GPS technologies to

locate accidents. The new version of OIM/MARS, Advantage Safety, will have expanded capabilities.

(taken from the Location Tool, Phase II proposal, CTRE)

Deer crash study

Researchers in the Iowa State University Animal Ecology department performed a literature review regarding deer crashes as Phase I of a Iowa DOT study regarding deer crashes. Phase II was to be an in-depth study of Iowa's deer crash problem; however, Phase II was not funded. A report is available through the Iowa DOT Office of Transportation Safety.

FUTURE ACTIVITIES

During Phase III, three tasks will be addressed to further enhance GIS-ALAS. The first task involves further system enhancements as well as statewide safety program support. The second task efforts are to develop intersection database management and analysis capabilities. The third task is to examine two case-study applications of the system. For more information, refer to the Phase III proposal or contact CTRE personnel.

APPENDIX A

ArcView ALAS Suggested Improvements List

Following is a list of improvements that may be made to ArcView ALAS as well as some that have already been made. Those items that have been completed are indicated. If the item has not yet been completed a priority level has been indicated as either low, medium, or high. Low prioritization indicates that though this item would be nice, it isn't as important as finishing other items. Medium prioritization suggests that a little more emphasis should be placed on completing the item. High priority indicates that for the system to be viable, the item should be addressed as quickly as possible.

A. Interface

1. Appearance
 - a. Add parentheses to run location requests (completed)
 - b. Change railroad line type to something appropriate (low)
 - c. Color instructions for color-blind people (low)
 - d. Create ability to generate graduated sizes/colors of dots for numbers of accidents (low)
 - e. Disable commands that currently cannot be utilized (low)
 - f. Eliminate AV-ALAS scripts opening on errors -> encrypting? (medium)
 - g. Eliminate underscores from menus/dialogs/etc. if possible (low)
 - h. Fit text into windows/dialogs/etc. (medium)
 - i. If no data in directory, take off selection list or vice versa (completed)
 - j. Master report sheet with all report options listed (medium)
 - k. Menus only show up when project window is open, needs to show up when views (maps) or tables are active (completed)
 - l. Name the view that is opened something that indicates what is contained (e.g., county name, RPA) (completed)
 - m. Slow! Speed up? ? index, search only applicable years, search for all crashes at same time (completed)
 - n. Stackable crashes (completed)
 - o. When querying multiple counties place the query results into 1 table for the A records, 1 table for the B records, and 1 table for the C records and not have a table for the A, B, and C records for each county (medium)
 - p. When selecting crashes within a distance of a feature create a circle or other shape to reflect what that actual distance is from that point. In the example below the crashes 100 feet from the node (large blue dot) were selected, however providing a graphical reference allows a much clearer picture of where this 100-foot area is. (low)
2. Operation
 - a. Speed
 - 1) Link request inclusive of nodes is too slow. (medium)
 - 2) Redraw (invalidate) display after each county opens? (low)
 - 3) Select only from those layers that are pertinent. (completed)

- 4) Use point selection to grab all layers related to selected crashes (i.e., A, B, C records) (medium)
 - 5) When session has already identified available counties, need not identify them again every time user opens another county. (completed)
- b. User-friendliness
- 1) Allow automated creation of thematic maps (e.g., Road ADT, type of collision) (low)
 - 2) Allow users to start afresh with selection and queries without closing AV-ALAS. (high)
 - 3) Automatic recognition of earliest and latest dates available (completed)
 - 4) Automatically recognize whether nodes/features have been selected (completed)
 - 5) Back button for parameter selections (high)
 - 6) Code dialogs to work (high)
 - 7) Create configuration files for: (medium)
 - a) counties available
 - b) RPAs, MPOs, ITCs, statewide available
 - c) Columns available
 - d) Column names
 - e) Years available, years desired, dates desired – each county?
 - f) Root directory, program directory, data directory, query directory, reports directory
 - g) Defaults saved to files, called up when AV-ALAS opens
 - 8) Creating 3 stages of hierarchy for users (i.e., beginner, advanced, and power user) by doing this it would allow functionality to be deactivated for the low end users (low)
 - 9) Delete previous queries when a new query is specified. Save old queries if user wants to. (medium)
 - 10) Eliminate month and day option when requesting dates desired (completed)
 - 11) Enable/change help information (medium)
 - 12) If options have eliminated all crashes, inform user and request different data refinement (medium)
 - 13) Incorporate a choice for selecting all options (low)
 - 14) Provide a visual check on what selection parameters have been chosen and also allow a back button to change parameters selected (i.e., if you select injury and accident information and you start to select the additional parameter and realize that you want to include driver information there is no way to go back and change your initial selection without starting completely over) (medium)
Reset AV-ALAS for new location specifications (medium)
 - 15) Wait messages (completed)
 - 16) Rename A, B, and C themes with more descriptive names, e.g. “Co. 85 Yr. 94 - Injuries”. If this isn’t possible, explain the differences among them in the online help and provide help buttons in the dialog boxes. (medium)

- 17) Widen the Themes pane and the Identify Records window so that users can read the theme names without resizing. (low)
 - 18) Take multiple parameters in the same window so that users will be able to review their choices in relation to each other. These windows might look similar to old PC-ALAS dialog boxes. (medium)
 - 19) Present series of windows as Wizards so that users can easily go back to review and correct previous steps. (medium)
 - 20) Plan window placement so that windows do not obscure each other. For instance, the Identify Records window could always appear in the upper left quadrant of the screen, while the report tables could always appear in the lower right quadrant. (low)
 - 21) Plan window size so that windows won't have to be resized. Widen the Identify Records window so that each theme's name is completely displayed. Resize the report table windows so that they can be read without scrolling. If possible, resize each report window to fit the specific report table (e.g. the Fixed Objects Struck window would be a different size from the Parameter Sheet window). (low)
3. Functionality
- a. Code string of nodes, milepoint, milepost (completed)
 - b. Combine (append) years of interest together during session? (low)
 - c. Eliminate saving changes to AV-ALAS, but allow changes to saved as another project. (low)
 - d. Eliminate the default setting for default root directory and default directory name (completed)
 - e. Eliminate Transportation Center/Statewide options or create regionwide coverages or speed it up (low)
 - f. Enable selection of crashes nearest (within distance of) user-specified point. (medium)
 - g. Enable user specified regions, points, etc. (medium)
 - h. Facilitate selection of crashes using buffering of selected features. (medium)
 - i. Functional selection of accidents using actual accident locations (completed)
 - j. In order to open more than one county, needs more coding (completed)
 - k. Modify flag notation (actual accident locations) ? flags doesn't do anything? (completed)
 - l. Only ask for logic specification once and inform that this is now the default and must be changed there. (low)
 - m. Provide a way to have program remember where the last information was stored or accessed and default directory to that location (low)
 - n. Provide a way to print information about one crash with relevant information (not the numbers and having the user look up what they mean) (medium)
 - o. Provide information as to how each data refinement parameter limited the selection set? (low)
 - p. Provide information or automated way to query with respect to other theme attributes. For example, injury crashes between the hours of 1800 and 2200 that occurred on streets with an ADT less than 500. (medium)
 - q. Provide precise information about the date analysis capability (low)

- r. Search for data directories automatically (completed)
- s. Use summarize to produce symbols indicating # of accidents at a point (medium)
- t. Work on regional selection: RPA, ITC, statewide, multiple counties, cities/rural (medium)

B. Query

1. Fix the following: (high)
 - a. Collision type
 - b. Injured severity
 - c. Type of accident
 - d. Driver/Vehicle Contributing Circumstances
 - e. Time of Day
 - f. Special Use Vehicle
 - g. Fixed object type accidents
 - h. Driver condition
 - i. Position of injured
 - j. Traffic controls
 - k. License restrictions
 - l. Combination requests
 - m. User defined queries
 - n. Sex of driver (Gender?)
 - o. Driver charged
2. Create these special queries (and others?): (low)
 - a. Weather-related accident causes/dates
 - b. Date vs. state parks
 - c. Proximity to signals or other interesting features (bars, large employment centers, downtowns, shopping centers, curves)
 - d. Crashes per type of road
 - e. Highlighting points with high numbers of certain types of crashes (e.g., animal in roadway crashes)
 - f. Create a query for X # of injured, etc.
3. Provide some kind of information about the status of a query (possible suggestions are a status bar or a message stating query in progress please wait) (low)
4. Put option for # of vehicles to include 1, 2, 3, 4, 5+ (completed)
5. Automated way to save query results before further dividing (refining) search (medium)
6. Once query is executed then zoom to selected to get a close up view (medium)
7. New search on 1st subset and not second parameter selection on original data (medium)
8. Is there any way to save a query? Or log what the previous X # of queries were? (medium)
9. Address issues of county-wide searches along with: (high)
 - a. selection of crashes occurring in just a city
 - b. selection of crashes occurring in the county but not in a city (Sheriff Department)

- c. selection of crashes in multiple cities
- d. using the linework to select crashes (automated for all crashes on a particular road)

C. Report

1. Reports do not have fixed object struck #s in them (completed)
2. Put the refinement parameters in the report (possibly under parameter sheet?) (high)
3. Crash rate is not filled in for the reports (high)
4. Surface conditions do not have values in the report form (completed)
5. If generating multiple reports there is no way to easily differentiate what report is for what (i.e., there are 2 fixed object struck, 2 parameter sheet)...delete old reports (ask if want to save)...have dialog which opens those portions of the reports desired...(high)
6. Non-fatality issues reporting (completed)
7. Dynamic, interactive vs. set reports (low)
8. add graphs and/or charts? Link to Excel to produce charts? (medium)
9. Finish reports: engineering report, enforcement report, ranking report, user-defined graphs, other reports that might be useful -- divide these by year, location, any other division? (medium)
10. Address current problems with the report tables and provide a master report that includes all information that is broken into smaller components. This would provide print out capability for all of the reports without wasting paper space, but also allows certain reports of interest to be printed out when needed (high)
11. Develop a script to select tables from the "av_alas.apr" window and print them directly, without opening them. (high)
12. Develop a script to print the A, B, and C records for each selected accident, preferably in the same format that the PC-ALAS report prints detailed accident data. (high)

D. Printing

1. Printing selected records only (low)
2. Printing of reports (high)
3. Print the time and date as a footer or header for any printouts of maps, tables, and reports (medium)

E. Data

1. Multiple years for each county in one file for Shape, A, B, and C (low)
2. Shape, A, B, C stuff (low)
3. Create mpt2shp process in AV (low)
4. Obtain city corporate limits coverage for use in AV-ALAS city searches (medium)
5. Explorer ALAS ? ABBBC for 10 years for entire state (completed, ongoing?)
6. Delete all fields that are unnecessary (low)
7. Getting all BR ROADS information into several tables for use in AV-ALAS (medium)

F. Documentation

1. Update Developer's Guide (medium)
2. Update User's Manual (medium)

3. Document code (medium)
4. Develop training and documentation materials to help users better understand the relationship among the different records. (medium)
5. Develop an online reference to help users interpret descriptions such as VEHICLE_TY=1. Or find ways to provide fuller descriptions. (medium)
6. Develop thorough online help. Users often have trouble finding and coordinating resources (such as training manuals) with online programs. If the online help included a thorough reference, users may have less trouble and may become more self-sufficient, and additionally, may find new uses for GIS-ALAS. (high)
7. Develop a training program that includes online help use. Users often haven't encountered online help that is a useful reference, so they may need some time to get used to the idea and to understand how to use the help as a reference. (high)
8. In the training program, encourage users to share expertise. This may involve teaching users how to work on GIS-ALAS together. It also may involve encouraging them to join a support apparatus such as an electronic mailing list so they can ask questions and share answers. (high)

G. Miscellaneous

1. Shorten code by modularizing more (low)
2. Severity determinations (for crashes, sites, etc.) (low)
3. Further enforcement-related enhancements (medium)
4. Intersection magic interface (high)
5. Statewide operation (high)
6. Buffering roads to get nearest accidents (all accidents along line) and to calculate crash rate (low)
7. Buffering railroads to get all accidents along rail lines (low)
8. Interchange proximity (low)

APPENDIX B

Explorer ALAS Data File Creation (ABBBC) Procedure

The following is an outline of the ABBBC process. Following this process results in creation of the crash database in ArcExplorer format.

1. Create directories for code and place executables in correct directories

- **d:_abbbc**
 - **d:_abbbc\abbbc.bat**
 - This batch program runs the FORTRAN programs for combining and delimiting the A, B, and C record text files in **d:_abbbc\in** into ABBBC format outfiles in **d:_abbbc\out**. The batch program also deletes all files in **d:_abbbc\in** upon the answer of **Y** to the question asked.
 - **d:_abbbc\delzipYY.bat** (*YY stands for the last two digits of the year*).
 - These batch programs create **.zip** (WinZIP extractable) and **.exe** (self-extracting) files containing the **abbbc .shp, .shx, and .dbf** files for each county for the specified year. The batch programs then move the **.zip** and **.exe** to the proper **e:_abbbc\shp\YYYY** (*YYYY stands for the four digit year*) directory and also delete the files from the **d:_abbbc\out** and **e:_abbbc\shp** directories.
 - **d:_abbbc\FORTRAN FILES** (i.e., all the files for the fortran programs **abbbc, abbbc1, abbbc2, abbbc3, abbbc4, abbbc5, abbbc6, and abbbc7**. These include **.for, .ilk, .mak, .mdp, .obj, and .pdb** suffixes. The **abbbc** programs with numbers are actually used, the one without a number is the original and is no longer used.)
 - These FORTRAN programs read the A, B, and C records for each county for each year (from **d:_abbbc\in**) and combine them to create ABBBC format output files for each county for each year (to **d:_abbbc\out**). To do this, first the B and C records for a particular county and year are read into arrays. Then the A records are read in one at a time and compared to the B and C records searching for a match of the key field. Only a maximum 3 B record and 1 C record are matched to any one A record. The results are then written to an output file, one line at a time. Each FORTRAN program does a subset of the 99 Iowa counties as trying to do all 99 counties in one program caused the code to crash.
 - **d:_abbbc\in**
 - Directory contains the output of the **ABBBC.YYYY.Setup.File.Exporter** (*YYYY stands for the four digit year*) script within **abbbc.apr**. These output files are used as the inputs for the FORTRAN programs. They are deleted at the end of the **abbbc.bat** code (user must reply **Y** to whether they want all files deleted). Directory also contains the **abbbc.out** file (the result of a dir request of the **d:_abbbc\in** directory). It is also deleted at the end of the **abbbc.bat** code.

- **d:_abbbc\out**
 - Directory contains the output of the **abbbc.bat** process (i.e., the ABBBC format files created by the FORTRAN programs). These output files are used as inputs for the **ABBBC.YYYY.TextShp.Combiner** (YYYY stands for the four digit year) script within **abbbc.apr**. The files are deleted as part of the **delzipYY.bat** (YY stands for the last two digits of the year) code.
 - **e:_abbbc**
 - **e:_abbbc\abbbc.apr**
 - An ArcView project containing the **ABBBC.YYYY.Setup.File.Exporter** and the **ABBBC.YYYY.TextShp.Combiner** (YYYY stands for the four digit year in both cases) scripts. To run these, open the project, either by opening ArcView and then opening the project or by opening the project from Windows Explorer. Then select scripts within the project window, select the appropriate script, and press the **Run** button. The former script(s) read the data directory (i.e., **e:_gisalas_maps_county**), open each crash file (A, B, and C) for each county for the specified year, and output the data to a text file (into **d:_abbbc\in**). The latter script(s) read the data directory (i.e., **e:_gisalas_maps_county**), open each A record crash file for each county for the specified year, and output the data to another feature file (into **e:_abbbc\shp**) of another name (i.e., **aeCCYrYY**, where CC stands for the county number and YY stands for the two digit year). The script then reads in the ABBBC format files (from **d:_abbbc\out**) and exports them to **.dbf** format (into **e:_abbbc\shp**), overwriting the **aeCCYrYY.dbf** files already in the directory (effectively creating feature files with correct crash locations and ABBBC format information).
 - **e:_abbbc\shp**
 - Directory contains the output of the **ABBBC.YYYY.TextShp.Combiner** (YYYY stands for the four digit year) script within **abbbc.apr**. These files are compressed and deleted as part of the **delzipYY.bat** (YY stands for the last two digits of the year) code.
 - **e:_abbbc\shp\YYYY** (YYYY stands for the four digit year)
 - Directories contain the **.zip** (WinZIP extractable) and **.exe** (self-extracting) files containing the **abbbc.shp**, **.shx**, and **.dbf** files for each county for the specified year.
- 2. Place crash records in proper directories**
- **e:_gisalas_maps_county\counties** (counties stands for the 99 Iowa county names)
 - Each county directory should contain the A, B, and C crash record files for the year required.
- 3. Open ArcView**
- 4. Open e:_abbbc\abbbc.apr**
- Run Avenue script – **ABBBC.YYYY.Setup.File.Exporter** (YYYY stands for the four digit year)
 - This process takes approximately 17 minutes* per year.

5. Open DOS shell

- Change directories to **d:_abbbc\in**
- Type **dir loc*a*.* > abbbc.out**

6. Open a text editor (i.e., Notepad, Wordpad, TextPad, DeskEdit, or DOS Edit)

- Delete all information other than **File Names** (i.e., save instances of **locCCaYY.txt**) (*CC stands for the county number, YY stands for the last two digits of the year*)
- Save file as the same name (i.e., **abbbc.out** in **d:_abbbc\in**)

7. Open Windows Explorer

- Change directories to **d:_abbbc**
- Run **abbbc.bat**
- This process takes approximately 7 minutes* per year. The process runs a FORTRAN program which matches the A records with the first 3 instances of corresponding B records and the first instance of corresponding C records, creating an ABBBC delimited text output file in **d:_abbbc\out** with the name **aeCCyrYY.out** (*CC stands for the county number, YY stands for the last two digits of the year*) for each county for the specified year. Progress is shown within a DOS shell. The batch program runs 7 different FORTRAN executables which each do a subset of the 99 Iowa counties since doing all 99 counties in one FORTRAN program seemed to have problems.

8. Return to ArcView

- Run Avenue script – **ABBBC.YYYY.TextShp.Combiner** (*YYYY stands for the four digit year*)
- This process takes approximately 27 minutes* per year.

9. Back to Windows Explorer

- Run **delzipYY.bat** (*YY stands for the last two digits of the year*)
- This process takes approximately 2 minutes* per year.
- Zip file sizes are approximately 6.5MB, containing approximately 35MB of .shp, .shx, and .dbf files for each year.

10. Close ArcView

*All times determined on a Pentium 200 with 64MB RAM.

APPENDIX C

Independent Evaluation: Bob Schultz

Following is the list of suggestions submitted by Bob Schultz. They have been incorporated into the larger list in Appendix A.

1. Creating 3 stages of hierarchy for users (i.e., beginner, advanced, and power user) by doing this it would allow functionality to be deactivated for the low end users
2. Provide a user's guide to ArcView along with a thorough users guide to use AV-ALAS. This users guide should contain several examples that walk the user through the process of selecting crashes via node requests, link requests, crash locations, etc.
3. Provide some kind of information about the status of a query (possible suggestions are a status bar or a message stating "query in progress please wait")
4. Provide a visual check on what selection parameters have been chosen and also allow a back button to change parameters selected (i.e., if you select injury and accident information and you start to select the additional parameter and realize that you want to include driver information there is no way to go back and change your initial selection without starting completely over)
5. When querying multiple counties place the query results into 1 table for the A records, 1 table for the B records, and 1 table for the C records and not have a table for the A, B, and C records for each county
6. Provide a way to print information about one crash with relevant information (not the numbers and having the user look up what they mean)
7. Address issues of county-wide searches along with:
 - a. selection of crashes occurring in just a city
 - b. selection of crashes occurring in the county but not in a city (Sheriff Department)
 - c. selection of crashes in multiple cities
 - d. using the linework to select crashes (automated for all crashes on a particular road)
8. Provide precise information about the date analysis capability
 - a. What is the minimum time period for analysis?
 1. 1 day
 2. 1 week
 3. 1 month
9. Provide a way to have program remember where the last information was stored or accessed and default directory to that location
10. Print the time and date as a footer or header for any printouts of maps, tables, and reports
11. Address current problems with the report tables and provide a master report that includes all information that is broken into smaller components. This would provide print out capability for all of the reports without wasting paper space, but also allows certain reports of interest to be printed out when needed

12. Place the parameter selected on the tables as a header. There are several types of analysis that are done on the same locatoin; providing the parameters the output will not be mixed up or misunderstood
13. Provide information or automated way to query with respect to other theme attributes. For example, injury crashes between the hours of 1800 and 2200 that occurred on streets with an ADT less than 500.
14. When selecting crashes within a distance of a feature create a circle or other shape to reflect what that actual distance is from that point. In the example below the crashes 100 feet from the node (large blue dot) were selected, however providing a graphical reference allows a much clearer picture of where this 100-foot area is.

APPENDIX D

Results of GIS-ALAS Useability Research

Following is the memorandum sent by Clay Spinuzzi to CTRE. The suggested improvements have been incorporated into the complete list in Appendix A.

Memorandum

DATE: April 13, 1999
TO: Michael Pawlovich
FROM: Clay Spinuzzi
RE: Preliminary report and suggestions: GIS-ALAS

This report reviews my preliminary findings about PC-ALAS and Explorer-ALAS use as they relate to potential GIS-ALAS use. It also reviews preliminary test results of GIS-ALAS. Finally, it recommends changes to GIS-ALAS based on these findings.

PC-ALAS Use

I have extensively studied current PC-ALAS use because it's instructive to understand how users currently locate and analyze accidents, and how PC-ALAS helps and hurts those efforts. Twelve users were observed as they worked with PC-ALAS; eight others were interviewed and given questionnaires. Although I've found a wealth of information, a few things in particular seem closely related to GIS-ALAS use.

Data. An almost universal complaint was that of the age of data, particularly among those who write grants or answer the public's questions. This is the number one complaint.

Frequency. PC-ALAS use ranged from once a year to once a week. Most users, however, used PC-ALAS perhaps four times a year. Consequently, users generally don't use PC-ALAS enough to learn the program well. Since GIS-ALAS has a more complex interface, this problem might be even more severe.

Menu structure. PC-ALAS menu structure is sometimes confusing to the users. In particular, the Search menu is confusing: users expect vertical menu items to be discrete and independent actions, but the Search menu is really *a series of steps*. Consequently, a majority of the users either exhibited problems with performing searches or complained about such problems during interviews. The menu structure is changed with GIS-ALAS.

Reports. Reports were often printed for off-screen use. For instance, some users replied to questions from legislators or co-workers by printing reports for them. Other users scrolled through reports and formed hypotheses about the accidents by looking at summary information and information about individual accidents. These uses are

important because they present certain difficulties in GIS-ALAS, as I discuss in the GIS-ALAS Analysis section (below).

Coordination of resources. PC-ALAS users had trouble coordinating node maps, node tables, the PC-ALAS, and reports. Frequent problems included entering wrong node numbers, having difficulty unfolding and reading maps, dealing with out-of-date maps, and converting report findings back into map coordinates. These problems should mostly disappear with GIS-ALAS because the map representation is combined with the interface. (But the problem won't completely disappear -- see the recommendations below.)

Centralization of computing resources and expertise. This problem should not be underestimated. PC-ALAS use tends to be centralized within counties, cities or agencies. For instance, in most of the agencies I visited, one user is designated the "PC-ALAS user" and all requests are funneled through him or her. In some areas, one person (e.g. an engineer in the county engineer's office) gets requests from the other agencies in the area as well as from people within his own agency. Not only are the PC-ALAS duties centralized -- so is the expertise.

On the other hand, accident location and analysis *isn't* always centralized. Non PC-ALAS users could funnel requests to the local PC-ALAS user, but they often do things the old-fashioned way: compile accident statistics by hand or call the Iowa DOT.

These concerns are important in terms of GIS-ALAS use because it's likely that GIS-ALAS use will be even more centralized than PC-ALAS use. It has a more complex interface, requires more training, and takes more computing resources (i.e. faster computers with more memory, the purchase of ArcView). So, at least in the short term, GIS-ALAS should be optimized to support that model.

Explorer-ALAS Use

Only three users were observed using Explorer-ALAS. Yet I found a few things that relate to GIS-ALAS use.

Interpretation of accident information. Users had some trouble interpreting accident information. In particular, descriptions such as VEHICLE_TY=1 were hard to figure out and required users to make use of supplemental materials to decode the found information. It's important to note that this is another manifestation of the resource coordination problem that PC-ALAS users had: instead of referring to node maps and node tables, Explorer-ALAS users have to refer to manuals to find out what VEHICLE_TY means and what VEHICLE_TY=1 specifies.

Interpretation of accident records. Users also had trouble understanding the differences among A, B, and C records. These records have been in ALAS data since 1974, but this underlying structure has always been hidden; users tend to conceive of accidents as data points that encompass *all* accident data.

Errors in maps. This problem is a direct consequence of mapping the node-link system over the GIS map. The coordinates don't always match, so some accidents don't appear

directly on the roads. The users I observed used their knowledge of the local area to supplement the GIS data and interpret the maps.

These three concerns are important because they point to conceptual problems that users may have with GIS-ALAS in the future.

GIS-ALAS Testing

Eleven Community and Regional Planning students were observed using GIS-ALAS to perform prescribed tasks, and were interviewed immediately afterwards.

In the test, students were filmed as they followed instructions for operating GIS-ALAS. Using these instructions, they made a mile pointed request, looked at data for individual accidents, then printed a report on the data and interpreted a report table. Afterwards, they were interviewed about their experiences, with special attention to the problems that they had during the test.

Users had little trouble interpreting report tables or the specific information I asked them to interpret from individual accident data. In fact, overall they did well on the test. However, they did have certain common problems to which I give special attention below:

Data overload. Users seemed to be overwhelmed by the amount of data presented on the map. (The map initially displayed all 1990-1995 A records; hydrology; primary, secondary, and municipal streets; and railroads for Story County.) Although users were instructed to turn off all accident themes before requesting accidents, the users did not always seem to understand that they could scroll down the themes pane, and consequently they did not always turn off all the themes.

Window sizing. Windows were not always optimally sized. For instance, the Themes pane was narrow, meaning that it was difficult to see the differences in theme names -- especially after running the request. Similarly, the Identify Results window had to be either resized or horizontally and vertically scrolled for users to read the theme names -- an especially important point because the driver's sex is printed at the end of these theme names. Finally, all report windows were the same size -- too small for the tables -- again meaning that they had to be scrolled.

Window obstruction. A related problem that the students had was that when the report table windows are brought up, they overlap the Identify Results window. Since the Identify Results window is always on top, some users were confused when they brought up a report table and it appeared behind the Identify Results window.

Theme names. Users tended to be confused about theme names, particularly the difference among A, B, and C records. Most conceived of the different records as representing different accidents and were puzzled when they realized this was not so.

GIS-ALAS Analysis

In addition to the issues I've noted above, other issues present themselves.

Complexity of interface. The PC-ALAS interface was rather simple: it had a limited menu system, a limited set of input features, and only one window on the screen at a time. But GIS-ALAS' interface is far more complex. This increased complexity will mean more work on the part of users to operate the system.

Limitations in data selection. I've already noted some of the problems that PC-ALAS had with data selection parameters, especially in the course of performing searches. But PC-ALAS did have stable dialog boxes in which users could enter parameters, check them, and correct them (if necessary), then accept or cancel the dialog box. On the other hand, GIS-ALAS tends to receive parameters through a series of one-input dialog boxes and it's impossible to correct a wrong number if you've already gone on to the next item.

Limitations in treating reports as coherent wholes. In PC-ALAS, report elements are added cafeteria-style to a report, and the result is printed out as a whole. However, in GIS-ALAS, report elements must be printed separately. Furthermore, data for individual accidents must be printed separately through the Identify Results window, accident by accident and record by record. This arrangement might make it difficult to sustain users' current patterns of using reports.

Preliminary Recommendations

I have not finished my data analyses. But based on the study so far, I suggest the following changes to GIS-ALAS. It might be that not all of these are practical due to limitations in ArcView's scripting language.

Find ways to help users understand the underlying record structure. The A, B, and C records must be represented as separate themes in GIS-ALAS. This is a very different sort of representation than was used with mainframe-ALAS and PC-ALAS. Users must be helped to understand how these records relate to each other and to the accident. I suggest the following:

- Rename A, B, and C themes with more descriptive names, e.g. "Co. 85 Yr. 94 - Injuries". If this isn't possible, explain the differences among them in the online help and provide help buttons in the dialog boxes.
- Widen the Themes pane and the Identify Records window so that users can read the theme names without resizing.
- Develop training and documentation materials to help users better understand the relationship among the different records.
- Develop an online reference to help users interpret descriptions such as VEHICLE_TY=1. Or find ways to provide fuller descriptions.

Find more flexible ways to select data. As I mention in the analysis, GIS-ALAS receives parameters through a series of windows. This arrangement is good in that it walks users through necessary steps without skipping any. On the other hand, users cannot review the parameters they've entered before accepting them, and cannot remedy mistakes after accepting them. I recognize that there are certain limitations to the scripting language, but if at all possible, I suggest the following:

- Take multiple parameters in the same window so that users will be able to review their choices in relation to each other. These windows might look similar to old PC-ALAS dialog boxes.
- Present series of windows as Wizards so that users can easily go back to review and correct previous steps.

Find ways to treat reports as coherent wholes. PC-ALAS users are used to dealing with reports that include subsections. Yet GIS-ALAS treats the subsections as separate reports that must be printed separately. Although this arrangement has advantages -- such as the ability to print the subsections in any desired order -- users would find it helpful to be able to print a set of tables at once. I suggest the following:

- Develop a script to select tables from the “av_alas.apr” window and print them directly, without opening them.
- Develop a script to print the A, B, and C records for each selected accident, preferably in the same format that the PC-ALAS report prints detailed accident data.

Plan positions and sizes of windows. The GIS-ALAS users had trouble with overlapping windows and having to resize windows. I suggest the following:

- Plan window placement so that windows do not obscure each other. For instance, the Identify Records window could always appear in the upper left quadrant of the screen, while the report tables could always appear in the lower right quadrant.
- Plan window size so that windows won't have to be resized. Widen the Identify Records window so that each theme's name is completely displayed. Resize the report table windows so that they can be read without scrolling. If possible, resize *each* report window to fit the specific report table (e.g. the Fixed Objects Struck window would be a different size from the Parameter Sheet window).

Support locally centralized ALAS queries; support infrequent GIS-ALAS use. As I noted earlier, PC-ALAS is often operated in local centers: by one person in the agency, city, or county. GIS-ALAS will probably further this trend because it will have a steeper learning curve (so fewer people may feel comfortable using it) and will require more up-to-date computer equipment and software (so fewer copies of GIS-ALAS will be able to be used per county). Therefore, measures must be taken either (a) to ensure that users can be self-sufficient within their local centers or (b) to encourage users at local areas to share expertise. I suggest the following:

- Develop thorough online help. Users often have trouble finding and coordinating resources (such as training manuals) with online programs. If the online help included a thorough reference, users may have less trouble and may become more self-sufficient, and additionally, may find new uses for GIS-ALAS.
- Develop a training program that includes online help use. Users often haven't encountered online help that is a useful reference, so they may need some time to get used to the idea and to understand how to use the help as a reference.

- In the training program, encourage users to share expertise. This may involve teaching users how to work on GIS-ALAS together. It also may involve encouraging them to join a support apparatus such as an electronic mailing list so they can ask questions and share answers.
- Set up a website with online information that can be periodically updated. The information could include the user's manual in online form, examples of how GIS-ALAS is being used, and answers to commonly asked questions, as well as the most recent map/data downloads. It might also include case studies in which users relate how they have dealt with certain situations.

Michael, I hope these preliminary recommendations are helpful to you. I am continuing to analyze the data I've collected. Please feel free to call me at 296-1992 or email me at spinuzzi@iastate.edu if you have any questions. I hope things are going well with GIS-ALAS!

APPENDIX E

CTRE Link-Node to Latitude-Longitude (MPT2SHP) Process

The following is an outline of the MPT2SHP process. Following this process results in creation of the crash database in ArcView format.

1. **Copying program files for this process to the d:\ directory [or root directory of your choice (substitute whatever that root directory is for d:\ in the following)]:**
 - All program files should be in the d:_mpt2shp directory, these being:
 - 1996.bat
 - Alas_pts.mbx and Alas_pts.mb
 - Chname##.bat (## is the two year digit number)
 - Creatbat.exe, Creatbat.for, Creatbat.ilc, Creatbat.mak, Creatbat.obj, and Creatbat.pdb
 - Create##.mbx and Create##.mb (## is the two digit year number)
 - Deleter.bat
 - Exabc_##.mbx and Exabc_##.mb (## is the two digit year number)
 - Gentble2.mbx and Gentble2.mb
 - Impabc80.mbx and Impabc80.mb (for crash records in the 1980's)
 - Impabc90.mbx and Impabc90.mb (for crash records in the 1990's)
 - Impabc00.mbx and Impabc80.mb (for crash records in the 2000)
 - Node3.Tab, Node3.dat, Node3.id, Node3.ind, and Node3.map (these files are created earlier, it is a statewide ALAS node coverage)
 - Sample.tab, Sample.dat, Sample.id, Sample.ind, and Sample.map
 - Talas.wor
 - Subdirectories within d:_mpt2shp should be:
 - arc\
 - mif\
 - flags\
 - temp\
 - flagtables\
 - dat\
 - Note: All the program files have references to the d:\ directory and would need to have the root directory under which the program is actually run be substituted for the d:\ directory if another root directory is utilized.
2. **Copy .mpt files for the appropriate year(s) to the d:_mpt2shp\ directory**
3. **Rename .mpt files for the counties which are less than 10 (counties 1 thru 9) to include a zero prior to the county number (e.g., f1y96.mpt renamed to f01y96.mpt).**
 - A DOS batch program called chname##.bat will rename the files to the needed format. This file is located in the d:_mpt2shp folder. The ## reflects the year covered by the .mpt files (e.g., chname94.bat corresponds to 1994 mpt files).

- Go to the file manager and simply click on the file with the correct name and the program will begin to run and rename all files.
 - This program takes very little time to run. In some instances the DOS screen will appear and disappear. In other cases the DOS screen may remain in view with *finished* in the upper left corner; if this occurs simply close the screen by hitting the X in the upper right corner.
- 4. Create directory listing file of .mpt files**
- Open a DOS shell (e.g. at the bottom left corner of the screen click on the “start” button, then proceed to the “programs” selection, there should be an option of “Command Prompt”. This will display a new screen.)
 - Type d: (this will change the directory to the D drive, the location of the data)
 - Type cd _mpt2shp (this will change the current directory to the _mpt2shp directory)
 - If you are using Windows 95, type dir *.mpt > mpt.txt (this will create a listing of the mpt files)
 - If you are using Windows NT, type dir*.mpt>mpt.txt /-N (this will create a listing of the mpt files)
 - Close the DOS window (click on the X in the upper right corner)
- 5. Create a table called Tabname.tab from the Mpt.txt file**
- Open MapInfo
 - Within MapInfo:
 - A box will appear on the screen when MapInfo is opened
 - Choose the option Open A Table (Click Open)
 - Another box will appear on the screen
 - Change the Files of type from MapInfo to Delimited ASCII
 - Change the directory to the D:_mpt2shp
 - Choose the mpt file (this will open the file mpt.txt that was created in step 4)
 - Click Open
 - A box will appear on the screen that asks for information about ASCII format
 - Change the delimiter from tab to other (click on the circle to the left of the OTHER box)
 - In the box next to the other delete the comma (,) and replace it with a space (you need to press the space bar to do this)
 - Click on the OK button (this will open the table in MapInfo)
 - Next save the table as tabname
 - To do this choose File
 - Save copy as
 - Change name to tabname
 - Click Save
 - Now close the table named mpt.txt
 - File
 - Close All
 - Open the table tabname
 - File

- Open Table
 - Choose table Tabname from the d:_mpt2shp
 - Delete Rows containing nonfilename information
 - Choose the Table option at the top of the screen
 - Choose the Maintenance option
 - Choose the Table Structure
 - Remove the columns named _COL3, _COL4, _COL5, _COL6, _COL7 by highlighting each column and clicking on the Remove Field button.
 - Toward the bottom of the box is a place for the column name. Make sure that the “COL_1” is selected. Delete the current “_COL1” name and replace it with “Name”
 - Click OK
 - You will be told that “One or more of the fields have been shortened or removed. The resulting loss of data cannot be undone.” Click the OK button
 - The table may have disappeared, if so choose the window option from the top of the screen. From the drop down menu choose New Browser Window. This should show the new table on the screen (notice that there are only two columns now).
 - Some other material still needs to be removed from the table. To eliminate the unwanted rows click on the square to the left of anything that does not have MPT in the second column (be sure to scroll all the way to the bottom of the table). If there are multiple rows that need to be deleted hold the shift key down while selecting each row. When a row has been selected the box will become completely black.
 - Press the delete key on the keyboard. This should remove the unwanted rows.
 - Next you want to Pack the table to remove the now blank rows from the table.
 - Choose Table from the toolbar at the top of the screen.
 - Next choose Maintenance
 - And finally choose Pack Table...
 - Choose the table Tabname
 - You will be asked if you want to save changes before packing, choose Yes
 - Save all changes that we have made to the table.
 - Choose “File” from the top of the screen
 - Choose Close All
 - If asked to save changes respond with yes. If you are not asked then the changes have already been saved.
 - Proceed to next step
- 6. Be sure that Tabname, Sample, and Node3 are open in MapInfo**
- Choose the File option from the top of the screen
 - Next choose Open Table...
 - Open the files named: Tabname, Sample, and Node3
 - To open multiple tables at once use the “Ctrl” key and click on each file desired.
 - Click OK. The tables should appear on the screen (Note: Sample and Node3 should open as map files, while tabname should open as a browser window)
- 7. Divide .mpt files into A, B, C files**

- Within MapInfo, we will now run a program that will divide the MPT files into A, B, and C files.
 - Choose File
 - Next Choose Run MapBasic Program...
 - This should create a box on the screen with various MapBasic programs.
 - Change directories to the D:_mpt2shp directory.
 - Choose the Gentble2.mb MapBasic program.
 - Choose OK
 - This creates a files called fnm1.txt and several files with the endings a1.out, b1.out, and c1.out (e.g., f01y96a1.out, f01y96b1.out, f01y96c1.out)
 - This process takes under 5 minutes
- 8. Create files for use by later program**
- Now run a new program named Create##.mbx. This will create files needed later in the process
 - To run the program choose the File option at the top of the screen
 - Then choose the Run MapBasic Program option
 - This will open a window and allow you to choose which program to run.
 - If needed, change to the d:_mpt2shp directory
 - Choose the Create##.mbx program (note the ## reflects the 2 digit representation of the year, Create94 is for year 1994, Create00 is for year 2000).
 - This creates the d:_mpt2shp\dat\ directory files for use by a subsequent program.
 - It also creates the divider2.txt files.
 - **Outside MapInfo, run Creatbat.exe**
 - This creates the d:_mpt2shp\divider2.bat program
 - Go to the file manager and click on the createbat icon in the mpt2shp directory
 - This program takes essentially no time to run
- 9. Create the comma delimited A, B, C files**
- Run divider2.bat
 - To run this program go to the D:_mpt2shp directory and click on the divider2.bat icon on the screen. This should start the program.
 - This file runs the div2.for FORTRAN program on each counties A, B, C .out files (e.g., f01y96a1.out, f01y96b1.out, f01y96c1.out) and creates comma delimited files (e.g., f01y96a2.out, f01y96b2.out, f01y96c2.out)
 - It also creates a file called fnm2.txt repeatedly
 - This process takes under 10 minutes
- 10. Create MapInfo tables from comma delimited A, B, C files**
- Next we return to MapInfo to run another MapBasic program.
 - Within MapInfo, Choose the File option from the top of the window.
 - Next choose the Run MapBasic Program...
 - This will create a window that will show several files and folders.
 - If needed, go to the D:_mpt2shp direcotry
 - Then choose the Impabc90.mbx file (If you are doing the 1980's you will need the Impabc80.mbx, if you are doing the 2000 files you will need the Impabc00.mbx)

- Click OK
- This should start the file.
- This creates the MapInfo A, B, C files (e.g., f01y96a.tab, f01y96a.dat, f01y96b.tab, f01y96b.dat, f01y96c.tab, f01y96c.dat) from the comma delimited A, B, C files
- This process takes under 100 minutes

11. Create feature tables from A, B, C MapInfo tables

- Within MapInfo, choose File
- Then choose Run MapBasic Program
- Choose the Alas_pts.mb
- This creates the A, B, C MapInfo tables with features within the d:_mpt2shp\flags\ directory
- This creates other files (within the d:_mpt2shp\flagtables\ and d:_mpt2shp\temp\ directories) which are subsequently deleted
- This process takes under 3 hours.

12. Create ArcView files from MapInfo files

- Within MapBasic, File Run MapBasic Program Expabcpt.mb
- This creates the ArcView format A, B, C point coverages from the MapInfo format A, B, C point coverages by first exporting the MapInfo format files to MapInfo Interchange Format (MIF) and then running ESRI's mif2shp program on the MIFs to create ArcView format files. It then deletes the MIFs.
- This process takes under 40 minutes

Note the times listed are for a Pentium 200 computer with 32 Megs of RAM. Using a larger computer substantially decreased the time.