GIS TOOL TO MEASURE PERFORMANCE OF WINTER HIGHWAY OPERATIONS

Project 04-01
August 2006

Midwest Regional University Transportation Center
College of Engineering
Department of Civil and Environmental Engineering
University of Wisconsin, Madison

Authors: Alan Vonderohe, Carola Blazquez, Nick Brezovar, Sungchul Hong, Jason Lauters, Teresa Adams;
University of Wisconsin, Madison

Principal Investigator: Alan Vonderohe;
Professor, Department of Civil and Environmental Engineering, University of Wisconsin, Madison

DISCLAIMER

Acknowledgements

The authors wish to thank the Highway Departments and Land Information Offices of Columbia, Kenosha, and Portage Counties, all of whom provided data and much assistance throughout this project. The Marathon County Land Information Office provided data. The Bureau of Highway Operations at the Wisconsin Department of Transportation is thanked for its continuing enthusiastic support of the work reported herein. The Midwest Regional University Transportation Center at the University of Wisconsin – Madison provided funding and administrative support.

EXHIBIT B

Technical Report Documentation Page

1. Report No. **04-01**	2. Government Accession No.	3. Recipient's Catalog No. **CFDA 20.701**
4. Title and Subtitle **GIS Tool to Measure Performance of Winter Highway Operations**		5. Report Date **August, 2006** 6. Performing Organization Code
7. Author/s **Alan Vonderohe, Carola Blazquez, Nick Brezovar, Sungchul Hong, Jason Lauters, Teresa Adams**		8. Performing Organization Report No. **MRUTC 04-01**
9. Performing Organization Name and Address **Midwest Regional University Transportation Center** **University of Wisconsin-Madison** **1415 Engineering Drive, Madison, WI 53706**		10. Work Unit No. (TRAIS) 11. Contract or Grant No. **0092-05-22**
12. Sponsoring Organization Name and Address Wisconsin Department of Transportation Hill Farms State Transportation Building 4802 Sheboygan Avenue Madison, WI 53707		13. Type of Report and Period Covered **Final Report [01/01/05 – 08/31/06]** 14. Sponsoring Agency Code

15. Supplementary Notes
Project completed for the Midwest Regional University Transportation Center with support from the Wisconsin Department of Transportation.

16. Abstract

A five-year research effort (fifth-year funding through MRUTC) culminated in development, implementation, and installation of a GIS application for assessing performance of winter highway applications. The software accepts data recorded on board winter maintenance vehicles during operations and combines it with spatial data representing roadways and vehicle patrol sections. Analysts can then select among a number of performance measures and decision management tools for outputs from the system. Outputs are categorized according to labor, equipment, materials, and map displays that indicate vehicle routes and data collected along the way. The software, full user documentation, and necessary spatial databases were installed in two Wisconsin county highway department offices and at Wisconsin DOT headquarters. Training was provided to staff.

The spatial databases were developed, and scrutinized for quality, by the research team from data provided by the counties. FGDC-standard metadata were included with the spatial databases. Documentation of the full system included internal and external technical documentation for the software.

Final development of the application required refinement of performance measures, decision management tools, and the user interface. A number of previously unsolved technical problems also needed to be addressed. These included the "map-matching" problem in which moving vehicles must be tracked by roadway and patrol section by registering strings of two-dimensional vehicle coordinates to digital maps (spatial databases). The problem is exacerbated by errors in the coordinates and in the maps. A decision-rule algorithm was developed and tested against a number of available data sets. The algorithm resolves nearly all ambiguities encountered in the data. This algorithm is embedded in the installed version of the software. Testing revealed the limit (1:24,000) on source-scale of the spatial databases, needed to support the application.

Future maintenance of both the software and the data raise technical and institutional issues that were identified and described by the research team. Recommendations concerning these issues are included in the final sections of this report.

17. Key Words **Winter Operations, Concept Vehicle, AVL, GPS Application, GIS Application, Performance Measures, Map Matching**	18. Distribution Statement **No restrictions. This report is available through the Transportation Research Information Services of the National Transportation Library.**		
19. Security Classification (of this report) **Unclassified**	20. Security Classification (of this page) **Unclassified**	21. No. Of Pages	22. Price **-0-**

Form DOT F 1700.7 (8-72) **Reproduction of form and completed page is authorized.**

Table of Contents

Executive Summary

During the winter of 1999-2000, the Wisconsin DOT (WisDOT), in conjunction with Highway Commissioners from eight Wisconsin counties, embarked upon a 5-year research and implementation program for Winter Maintenance Concept Vehicles. In the first two years, concept vehicles were deployed in the Counties of Columbia, Florence, Manitowoc, Barron, Kenosha, Trempealeau, Portage, and Taylor. These vehicles contain numerous on-board sensing devices, including AVL/GPS, material spreader sensors, blade status sensors, vehicle speed sensors, and air and pavement temperature sensors. Data from these sensors are not only recorded in message sets on-board the vehicles, they are also telemetered in real time to base stations in County Highway Offices for monitoring and operational purposes.

Data of this nature, in kind and in quantity, have never before been available to transportation agencies in Wisconsin. All data coming from these vehicles are spatially and temporally referenced and are, therefore, ideal for management and analysis with geographic information systems (GIS) technology.

During years 1-4, of the Winter Maintenance Concept Vehicle Program, WisDOT contracted with the University of Wisconsin-Madison to 1) conduct research on the nature of the data, its quality, and its potential applications and 2) develop and test technology that exploits the data to support decision making at local and state levels. The research reported herein was funded, during year 5, by WisDOT through the Midwest Regional University Transportation Center. Year 5 was the concluding year of the overall project.

Year 5 work included refinement and comprehensive documentation of a GIS application dubbed “Wiscplow”. The application computes performance measures and produces decision management tools by post-processing data recorded on-board the vehicles and integrating it with spatial databases depicted roadways, patrol sections, and their attributes. The necessary spatial databases did not exist at the county level, so year 5 work included development and testing of them. Ultimately, Wiscplow, its documentation (including tutorials), and all necessary spatial databases were installed at the Columbia and Portage County Highway Departments and at WisDOT headquarters. On-site training was provided to staff.

Successful development of Wiscplow required solutions for a number of technical problems, not the least of which was the “map-matching” problem arising from spatial ambiguities in the data caused by positional and representational errors. Vehicle locations are captured as two-dimensional coordinates that must be associated with roadways and patrol sections by overlaying them with digital maps. Because there are errors in both the coordinates and the maps, vehicles can be computed to be on incorrect roadways. This results in incorrect values for performance measures that are often computed by patrol section and based upon traveled distance. The map matching problem was successfully addressed, during year 5, through development, testing, and implementation of a new decision-rule algorithm.

Wiscplow is intended for use, by analysts and other technical staff, in a post-processing environment. It is a standalone application, not an enterprise solution. It has no real-time monitoring capability and is not portable, as it requires access to databases that are typically only available on client-server networks. Its reporting functions are categorized by labor, equipment, materials, and map display. Internal and external technical documentation were developed for Wiscplow. FGDC-standard metadata were developed for all spatial databases. Wiscplow’s source code, all documentation, and all spatial databases are included on a CD

attached to the master copy of this report. A separate installation CD is also included with the master copy of this report.

On-going maintenance of both Wiscplow and the necessary spatial databases raises a number of institutional issues and options. Wiscplow requires roadway network shapefiles with source scales of 1:24,000 or larger. These shapefiles must include patrol sections. They must also be attributed with roadway functional class. "Custodianship" of these databases is a difficult issue. Options include:

1. WisDOT's local roads databases are maintained at headquarters in Madison. The unit that does this work clearly has personnel with appropriate skills and technology for maintaining Wiscplow spatial databases. However, such activities are not within their mission.

2. WisDOT's district offices use GIS as a matter of routine and could develop the capability to maintain Wiscplow's spatial databases. However, WisDOT district offices typically do not maintain or manage data for county highway departments.

3. It is technically feasible for UW-Madison to maintain the necessary spatial databases. However, the university is not in the practice of doing such things and some mechanism would have to be developed for continuity beyond the tenure of current Research Team members. Some level of continued funding would have to be developed.

4. Consultants could be contracted to maintain the spatial databases. Such an arrangement would, perhaps, be more institutionally stable than having UW-Madison do the maintenance, but a continuing funding source would have to be identified.

5. The most attractive option is to have either the county highway departments or the county land information offices take custodianship of the spatial databases. This keeps both use and maintenance at the local level. Those who need the data become responsible for it. Impediments include lack of skills in spatial database maintenance at county highway departments and lack of resources at county land information offices.

Options for maintenance of Wiscplow's source code include:

1. WisDOT's Bureau of Automation Services (BAS) has the skills necessary for maintenance of Wiscplow. However, BAS has a long-standing policy of not maintaining software developed by other parties.

2. The source code could be maintained at the county level, by land information offices or IT units. However, they would have to agree to do this. Furthermore, each installation of Wiscplow would evolve differently over time, as each would be maintained separately. Eventually, data models and, therefore, databases would become incompatible across installations.

3. UW-Madison could potentially maintain the source code. Some mechanism would have to be developed for continuity beyond the tenure of current Research Team members. Some level of continued funding would have to be developed.

4. Consultants could be contracted to maintain the source code. A continuing funding source would have to be identified.

5. The source code could be maintained, on a retainer basis, by a member or members of the Research Team. A continuing, low-level funding source would have to be identified.

1. Background

During the winter of 1999-2000, the Wisconsin DOT (WisDOT), in conjunction with Highway Commissioners from eight Wisconsin counties, embarked upon a 5-year research and implementation program for Winter Maintenance Concept Vehicles. In the first two years, concept vehicles were deployed in the Counties of Columbia, Florence, Manitowoc, Barron, Kenosha, Trempealeau, Portage, and Taylor. These vehicles contain numerous on-board sensing devices, including AVL/GPS, material spreader sensors, blade status sensors, vehicle speed sensors, and air and pavement temperature sensors. Data from these sensors are not only recorded in message sets on-board the vehicles, they are also telemetered in real time to base stations in County Highway Offices for monitoring and operational purposes.

Data of this nature, in kind and in quantity, have never before been available to transportation agencies in Wisconsin. All data coming from these vehicles are spatially and temporally referenced and are, therefore, ideal for management and analysis with geographic information systems (GIS) technology.

During years 1-4, of the Winter Maintenance Concept Vehicle Program, WisDOT contracted with the University of Wisconsin-Madison to 1) conduct research on the nature of the data, its quality, and its potential applications and 2) develop and test technology that exploits the data to support decision making at local and state levels.

2. Summary of Years 1-4 Research

Research during Year 1 included (Vonderohe, et. al., 2000):

1. Interviews of personnel within all eight participating counties and key personnel at WisDOT to identify needs.

2. Development of a draft model for the data, constituting a component of an underlying framework for applications.

3. Development of core GIS functionality for conversion of point data, as collected by differential Global Positioning System (DGPS) units aboard the vehicles, to data associated with roadway centerlines and, therefore, with roadway names and patrol sections.

4. Testing of the positional accuracy of the DGPS units.

Research during Year 2 included (Vonderohe, et. al., 2001):

1. Establishment of a permanent DGPS test facility on the UW campus.

2. Extension of a Microsoft Excel® application, initially written by Raven, Inc., for winter storm reporting. This application was documented and distributed to participating County Highway Departments.

3. Refinement and implementation of the model for detailed databases comprised of decomposed message sets and other data.

4. Identification of goals, objectives, and performance measures for Machinery Management and Winter Storm Report applications.

5. Development and implementation of a data model for summary databases founded upon the identified performance measures.

6. Initial-stage development of a GIS-based prototype for Machinery Management and Winter Storm Report applications.

7. Investigation of the feasibility of Real-Time Monitoring application development.

8. Examination of Local Roads Databases, being developed by WisDOT, with an eye towards their ability to support GIS-based winter maintenance concept vehicle applications.

Research during Year 3 included (Vonderohe, et. al., 2002):

1. Development of a methodology for building patrol section route systems in roadway spatial databases and application of that methodology to state and US highway representations in the Columbia County database.

2. Extension and refinement of the software to include a set of GIS-based applications for winter maintenance operations, based upon the computation of performance measures from raw data coming from the vehicles. The software system was dubbed "Wiscplow".

3. Field testing of Wiscplow in two Wisconsin counties.

4. Development of management decision tools for interpretation of computed performance measures and their interrelationships.

5. Analysis of the sensitivity of selected performance measures for Winter Storm Reports to temporal resolution and spatial noise in raw data coming from the vehicles.

6. Description of editing functions necessary for management of spatial and aspatial data supporting Wiscplow.

Research during Year 4 included (Vonderohe, et. al., 2003):

1. Identification and classification of spatial ambiguities between reported vehicle coordinates and digital roadway centerline representations. Such ambiguities can lead to incorrect determination of the roadway upon which a vehicle is traveling. They arise from positional errors in the reported coordinates and in the digital cartography. Without resolution, they result in incorrectly computed performance measures.

2. Conduct of two workshops for project participants. The first workshop introduced participants to Wiscplow functionality and developed detailed concepts for refined performance measures and decision management tools. The second workshop provided hands-on use of Wiscplow, after it had been revised according to needs identified at the first workshop. The second workshop also stressed implementation issues such as the

need for accurate roadway spatial databases and institutional support for Wiscplow in business operations.

3. Evaluation of WisDOT videolog GPS coordinate strings, collected in 2001 and 2002, as potential roadway centerline representations for use with Wiscplow. Analysis revealed gaps, overlaps, errors, and missing segments (e.g., ramps), that rendered these data unusable for roadway centerline representations. It was clear that the data would require extensive editing before they could be made sufficient for use with Wiscplow.

4. Description of data requirements for the Wiscplow system. Roadway spatial databases with 1:24, 000 or larger source scale are required. These spatial databases must be maintained and kept current. They must include patrol sections and functional class attributes. None of the participating counties possessed such data at the end of Year 4. Therefore, it was proposed that the Research Team, in conjunction with selected counties, develop the data during Year 5.

5. Description of options for short- and long-term institutional arrangements for management of winter maintenance vehicle data, roadway spatial databases, and the Wiscplow system. These options are re-articulated near the conclusion of this report.

3. Year 5 Research Objectives

Year 5 was envisioned as the concluding effort in 1) development of concepts, metrics, data, and technologies for management and exploitation of the rich data streams arising from winter maintenance concept vehicles and 2) planning for extended implementation, use, and management of the developed data and technologies into the future. Overall objectives of the Year 5 research were to:

1. Develop and implement a solution to the spatial ambiguity problem.

2. Develop and test the necessary roadway spatial databases for selected participating counties.

3. Complete development and testing of Wiscplow functionality, including performance measures, decision management tools, and spatial database management tools.

4. Develop complete technical and user documentation for the Wiscplow system.

5. Install and provide training on Wiscplow and the spatial databases at each selected participating County Highway Department.

6. Re-articulate the options for short- and long-term institutional arrangements for management of winter maintenance vehicle data, roadway spatial databases, and the Wiscplow system.

4. The Spatial Ambiguity Problem and Its Solution

Automated vehicle tracking involves determination of the roadway upon which a vehicle is traveling by registration of two-dimensional vehicle coordinate locations to cartographic representations of roadway centerlines (i.e., digital roadway maps). Problems sometimes arise due to errors in 1) the vehicle coordinates, collected by differential GPS and 2) the cartographic representations, typically having a source scale that limits their positional quality. Because of these errors, vehicle coordinates do not exactly register with roadway centerlines (see Figure 1). This phenomenon can lead to ambiguities in that, by examination of single pairs of coordinates, the vehicle can appear to be on the incorrect roadway.

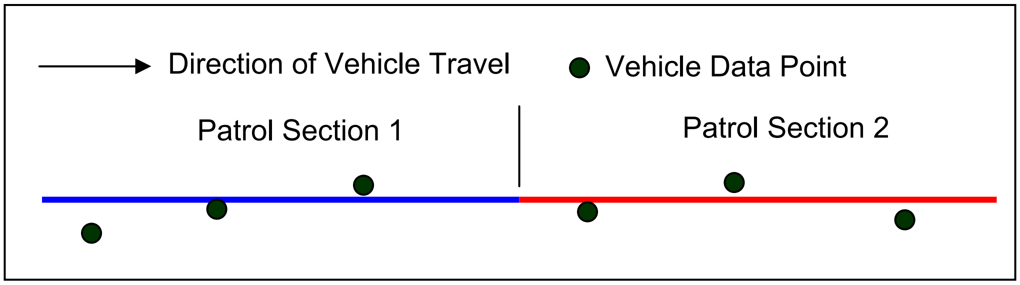


Figure 1.
Vehicle Data Points Do Not Register with
Cartographic Centerlines

Perhaps the simplest example occurs at at-grade intersections (see Figure 2). Spatial analysis software typically associates a two-dimensional data point with the centerline that is nearest by perpendicular measure. The mathematical process is referred to as "snapping". In Figure 2, the vehicle appears to be on the wrong roadway for the data point nearest the intersection. The problem is exacerbated at diverging or converging roadways such as ramps (see Figure 3). At complex interchanges, such as in Figure 4, multiple ambiguities can arise. The problem of resolving these ambiguities is known as "map matching" and has been the subject of research by a number of investigators for more than a decade (Blazquez and Vonderohe, 2005).

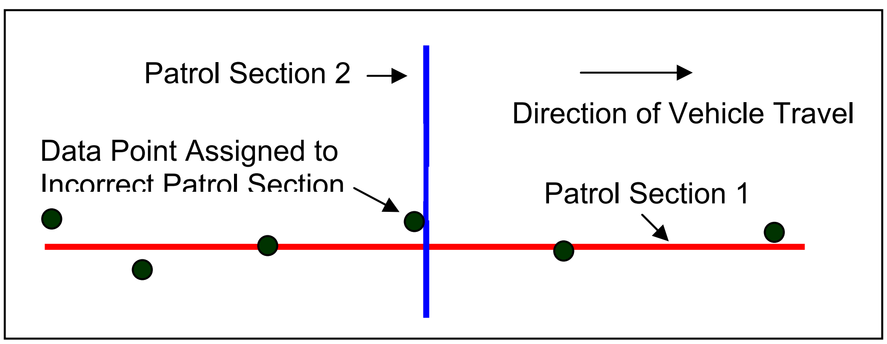


Figure 2.
Incorrectly Assigned Data Point at Intersection

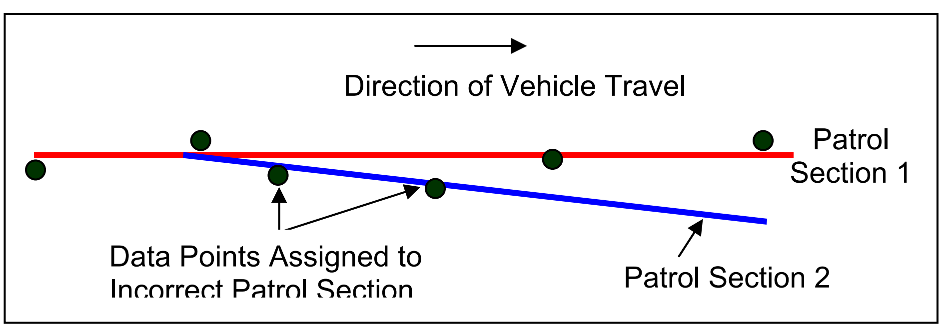


Figure 3.
Incorrectly Assigned Data Points at Ramp

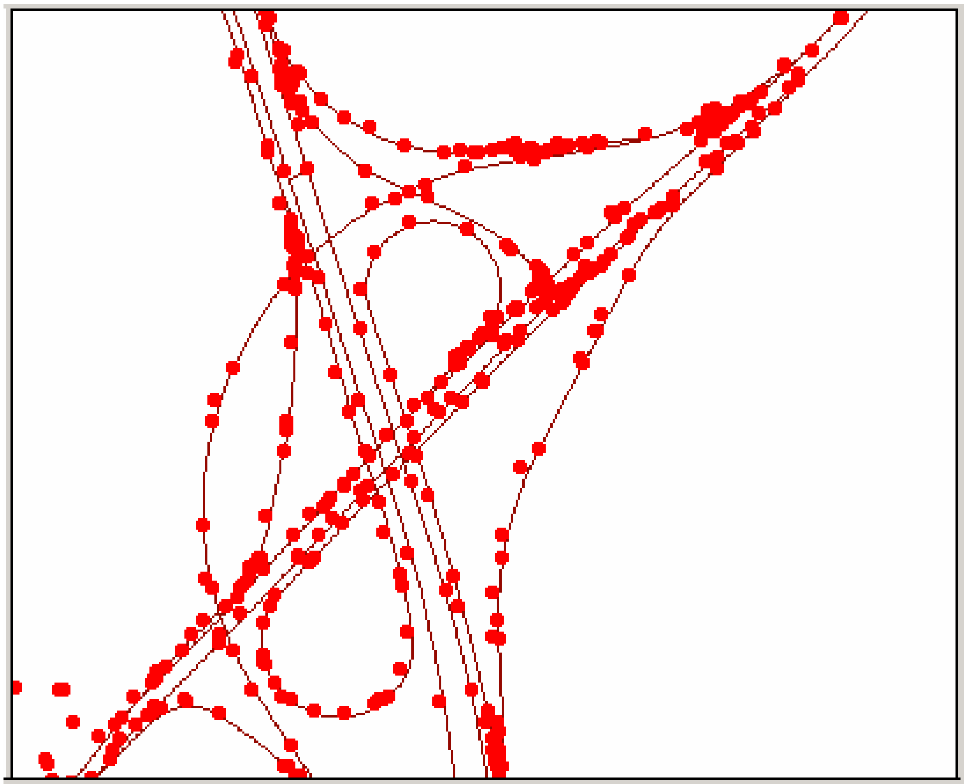


Figure 4.
Multiple Ambiguities at a Complex Interchange

Many performance measures, computed by Wiscplow, are associated with patrol sections. Clearly, if a vehicle data point is snapped to an incorrect patrol section, associated performance measures will be erroneous. In addition, calculation of some performance measures requires determination of traveled distances. A calculated traveled distance is grossly in error if one of a pair of consecutive vehicle data points is snapped to an incorrect roadway.

4.1. Map-Matching Algorithm

The map-matching problem was addressed by one of the Research Team members (Blazquez). Her algorithm examines strings of consecutive vehicle data points and uses recorded vehicle speeds, computed traveled distances, and allowable turns and directions of travel for the roadways (Blazquez, 2005). The algorithm selects all roadways within a buffer of specified distance around a vehicle data point and snaps the point to the closest roadway by determining the minimum perpendicular distance from the data point to each roadway. Consider a set of five consecutive vehicle data points (0 – 4). As shown in Figure 5, points 1 and 2 are snapped to ramp 2 because it is the closest roadway contained within the buffers around the points. Subsequently, the shortest path, indicated with a bold arrow, is computed between the two snapped data points (S1 and S2) using network topology and turn restrictions. The speed of travel between S1 and S2 is computed from the length of the shortest path and the difference in time stamps for the points. The computed speed is then compared to the average of the vehicle speeds recorded at the data points. If the computed speed is within a specified tolerance of the average recorded speed, then the obtained shortest path is viable and the snapped locations, S1 and S2, are accepted as correct.

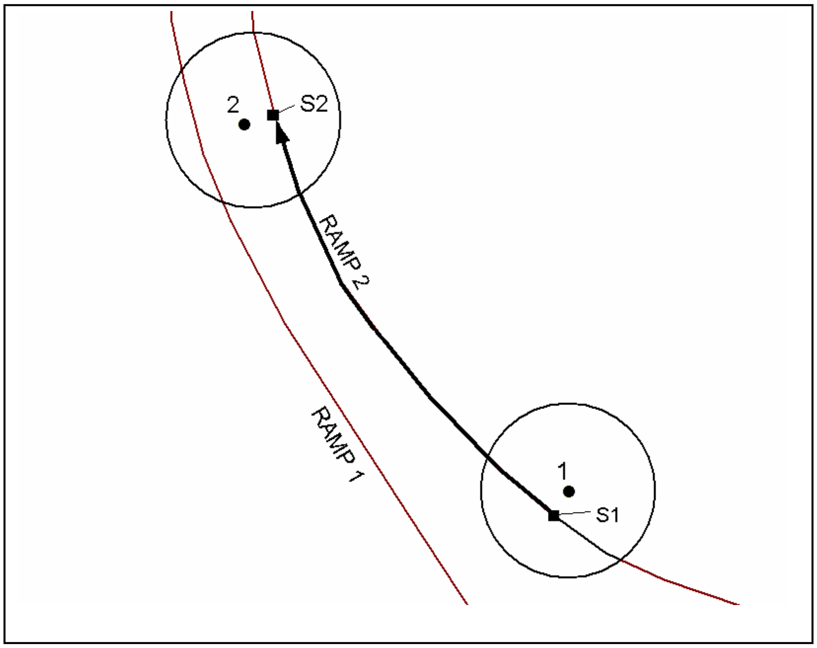


Figure 5.
Two Vehicle Data Points (1 and 2) and their Correctly
Snapped Locations (S1 and S2)

The algorithm advances to the next data point, 3, snaps it to the nearest roadway centerline within its buffer, and calculates the shortest path between S2 and the newly-snapped data point, S3. If the path between S2 and S3 is not feasible because the speed comparison yields a large disparity, then the algorithm follows the flow chart illustrated in Figure 6. The algorithm determines if feasible routes exist between the preceding and subsequent points bounding the data points of concern. If a feasible route is found using either of the bounding points, the intermediate data points not used to obtain the shortest path are snapped to the roadways along the determined feasible path.

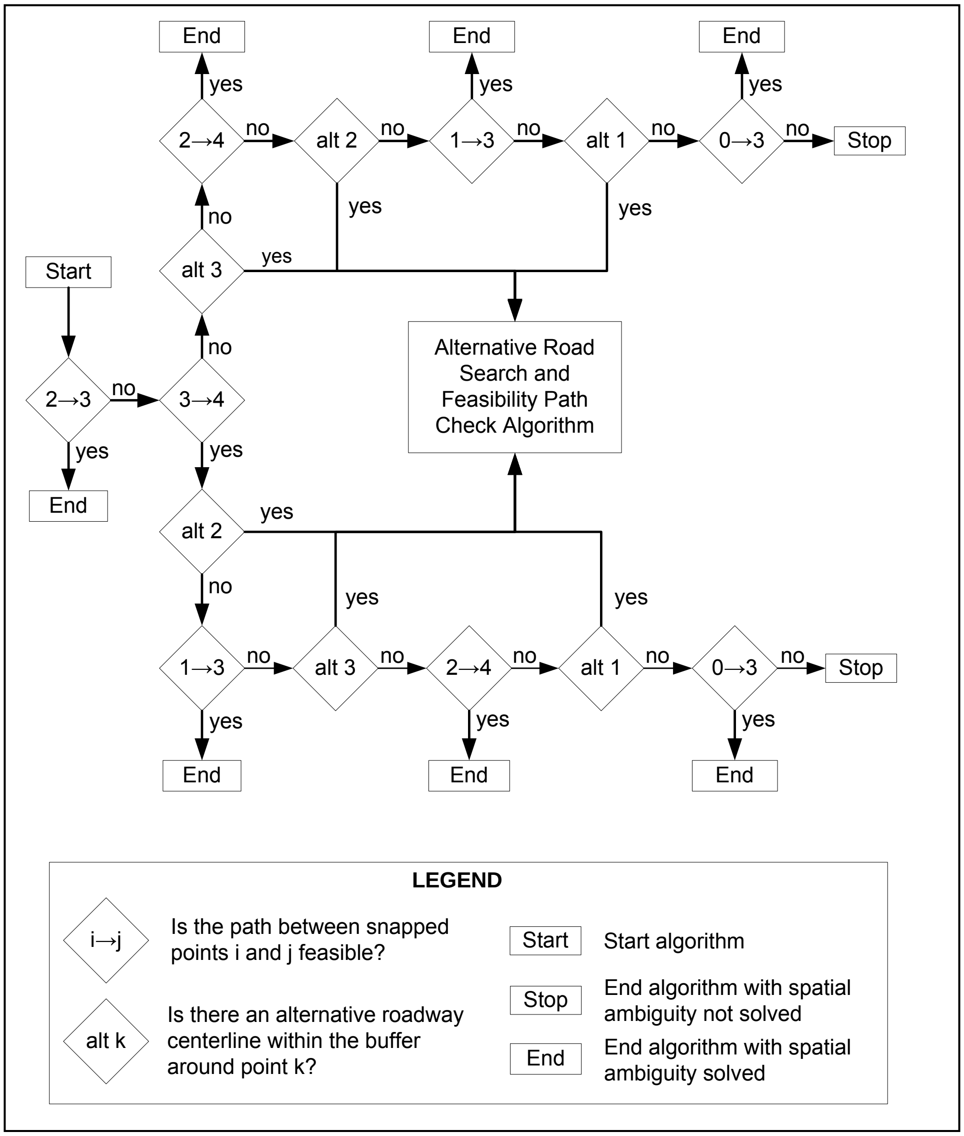


Figure 6.
Map-Matching Algorithm

For example, if there is no feasible path between S2 and S3, then the algorithm looks ahead by snapping point 4 to the nearest roadway centerline within its buffer, and determines if the shortest path between snapped points S3 and S4 is feasible. If the tested path is not feasible, the algorithm snaps point 3 to the next nearest roadway centerline within its buffer, obtaining point alt3. The algorithm illustrated in Figure 7 is initiated if an alternative roadway centerline exists within the buffer. This algorithm verifies if a path is feasible between the alternative snapped location for point 3 (where k = 3) and former and succeeding neighboring snapped points 2 and 4 (where k - 1 = 2 and k + 1 = 4). If the shortest paths between these three points are not feasible because the speed comparison fails, the algorithm searches for other roadway centerlines, within the buffer around point 3, that have not already been used in a feasibility path check. Finding a new candidate, point 3 is then snapped to it and the feasibility of shortest paths between snapped points 2, 3, and 4 (k - 1, k, k + 1) is checked again. If these paths are feasible, then the spatial ambiguity is resolved, and the algorithm in Figure 7 terminates.

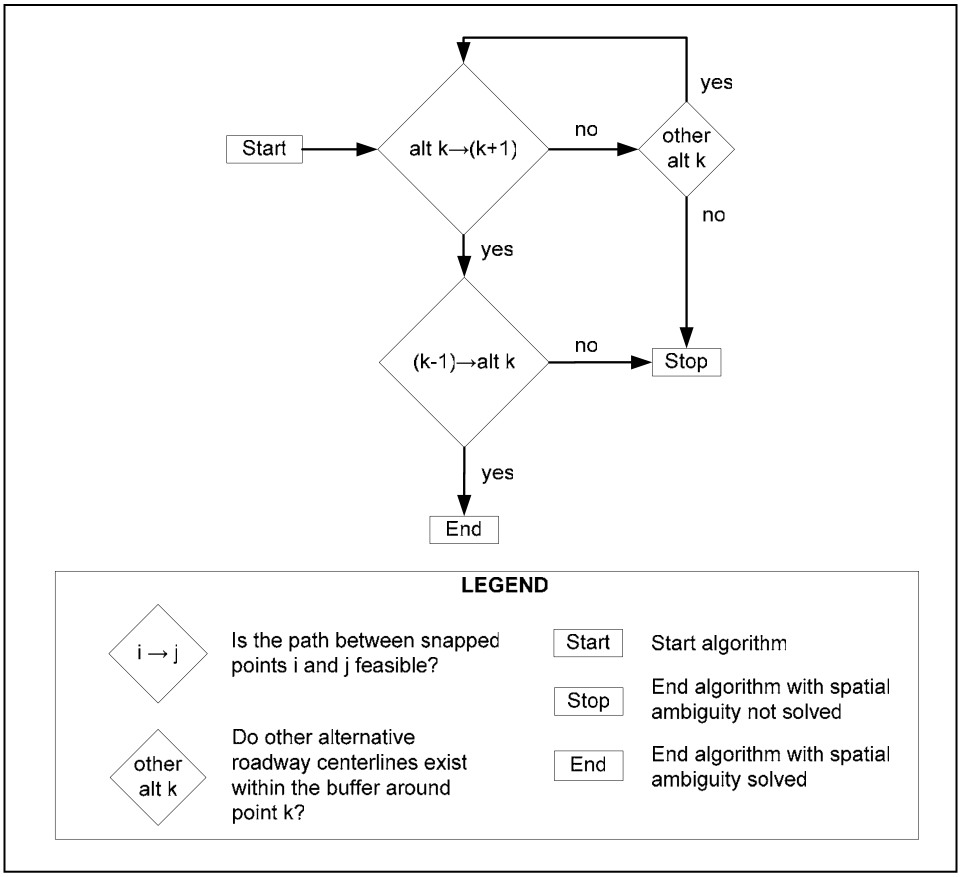


Figure 7.
Alternative Road Search and Feasibility Path Check Algorithm

If no alternative roadway centerline exists within the buffer for point 3, the algorithm in Figure 7 stops without resolving the ambiguity, using the alternative snapped location for point 3, and exits back to the algorithm presented in Figure 6. The feasibility of the shortest path between snapped points 2 and 4 is then tested. If this path is not feasible, the algorithm continues by snapping preceding points 2, 1, and 0, one at a time, to alternative roadways and determining if feasible paths exist between these newly-snapped points and snapped point 3. If none of the five consecutive points (0 through 4) aid in solving the spatial mismatch between the snapped points for points 2 and 3, then it is likely that no roadway centerlines within their buffers yield a feasible path and larger buffers and / or more consecutive data points need to be utilized by the algorithm. Once a feasible path is obtained, intermediate points not used during the map-matching process are snapped to the roadway along that feasible path.

On the other hand, if a path is feasible between snapped points 3 and 4, then, following the flow chart in Figure 6, alternative roadway centerlines are sought within the buffer for point 2. The algorithm in Figure 7 checks for feasibility of paths between the newly-snapped point 2 and the previous and subsequent snapped points. If there are no remaining alternative roadways within the buffer for point 2, or if there is no feasible path between point alt2 and the neighboring points, then the algorithm in Figure 6 checks the feasibility of the path between snapped points 1 and 3. If this path is feasible, point 2 is snapped to the roadway along that path and the spatial ambiguity is resolved. Otherwise, the algorithm loops back to point 3 and begins repetitions by examining the next nearest roadway within the buffer for point 3. The combination of both algorithms illustrated in Figures 6 and 7 calculates shortest paths between neighboring data points and alternative snapped locations for data points until a feasible path is obtained or until a specified number of consecutive points have been examined.

4.2. The Algorithm Applied to Actual Data

The example illustrated in Figure 8 includes a set of vehicle data points collected every five seconds by a winter maintenance concept vehicle during the 2002-2003 winter season in Columbia County, Wisconsin. Columbia County's roadway centerline spatial database has a nominal scale of 1:4,800. The spatial mismatch, occurring at the diverging roadways in Figure 8, is resolved by the map-matching algorithm. In this example, points 0, 2, 3, and 4 are snapped to the nearest roadway within their buffers, resulting in points S0, S2, S3, and S4, shown as rectangles. Points S0, S3, and S4 are on the Interstate 39 centerline, while point S2 is on the ramp centerline. No roadways are contained within the buffer for point 1, thus this point is not used in determining the feasible path. However, point 1 is snapped to the correct roadway once the shortest path between the adjacent data points is determined to be feasible.

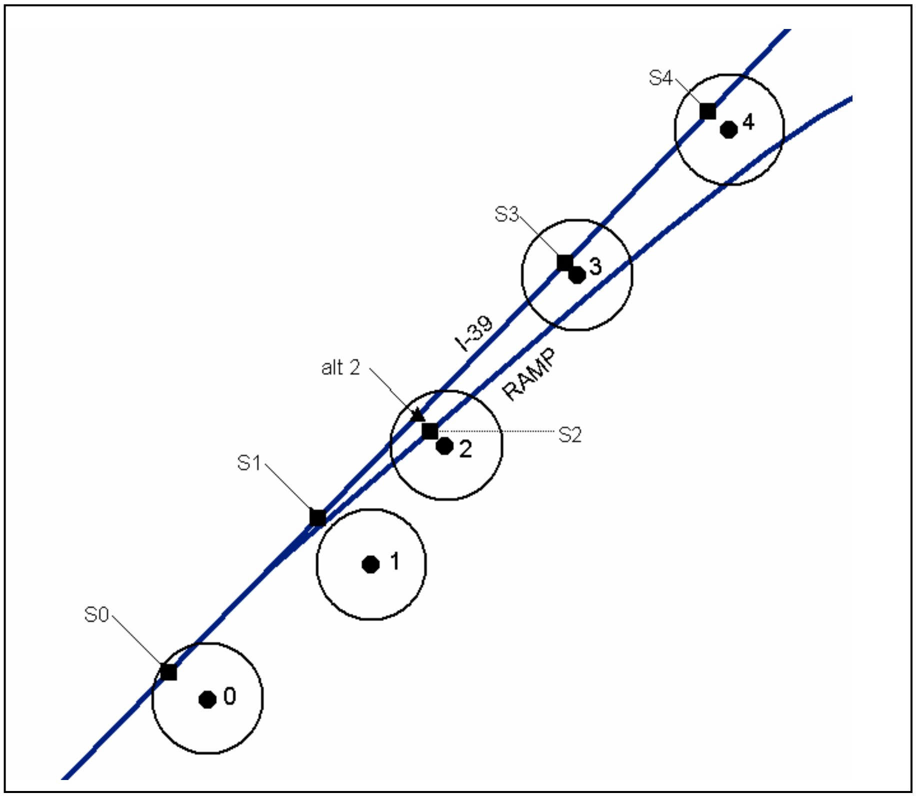


Figure 8.
Resolution of Spatial Ambiguities at a Ramp

The shortest path between points S0 and S2 is computed. Subsequently, the speed comparison shown in Table 1 is performed to determine if this path is feasible. In this case, the obtained path is feasible since the difference between the average calculated and recorded speeds (26 and 31 mph, respectively) is within tolerance (20 mph). Therefore, the current snapped positions for points 0 and 2 are initially assumed to be correct. The algorithm continues by finding the shortest path between the next pair of snapped points, S2 and S3. This path is not feasible because if the vehicle was at S2, it would have to exit down the ramp and travel approximately 5,126 feet in 5 seconds at an average speed of 699 mph to reach S3.

Thus, S2 or S3 or both were snapped to an incorrect roadway centerline. The algorithm now looks ahead to S3 and S4 and determines that the difference between calculated (29 mph) and average recorded speeds (35 mph) is within tolerance. Therefore, an alternative roadway centerline is sought within the buffer around point 2. Interstate 39 is found to be the next nearest roadway, resulting in point alt2, shown as a triangle in Figure 8. Feasibility is now checked for paths between the preceding S0 and alt2, and between alt2 and its successor, S3. As indicated in Table 1, both computed shortest paths are feasible. Calculated speeds along these paths are within 20 mph of their respective average recorded speeds for the vehicle. Therefore, the spatial ambiguity at the diverging roadway is resolved and the correct roadway for point 2 is Interstate 39. Point S1 is then obtained by snapping point 1 to the Interstate 39 centerline.

Table 1.
Speed Comparison for Determining Feasibility of Shortest Paths

Data Points	Shortest Path Distance (feet)	Calculated Speed (mi/hr)	Average Recorded Speed (mi/hr)	Feasible Path?
S0 → S2	392	26	31	YES
S2 → S3	5125	699	33	NO
S3 → S4	213	29	35	YES
S0 → alt2	392	26	31	YES
alt2 → S3	215	29	33	YES

The algorithm resolves spatial ambiguities at converging roads, divided highways, and intersections in a similar manner. Figure 9 presents results for vehicle data points at an interchange in Columbia County, Wisconsin. Circles indicate original measured vehicle data points. Incorrectly snapped locations for these points are depicted as asterisks, and correctly snapped locations, determined after the algorithm was executed, are shown as rectangles. Bold arrows represent shortest feasible paths between correctly snapped points. Eleven of 28 vehicle data points were initially snapped to incorrect roadways. All of these spatial mismatches were resolved by the map-matching algorithm.

The appropriate buffer size for the algorithm depends on the quality and geometry of the spatial data (i.e., vehicle data points and roadway centerline spatial databases). Testing of Columbia County data indicated that buffer sizes from 30 to 45 feet correctly snapped all vehicle data points. Separate testing of Portage County, Wisconsin data indicated an appropriate buffer size of 45 to 50 ft. The source scale of the Portage County roadway centerline spatial database is 1:12,000, while that of Columbia County is 1:4,800.

The map-matching algorithm is very effective. However, it is based upon certain assumptions which, if violated, will cause it to fail. Spatial ambiguities due to the vehicle moving against allowable directions of travel or making disallowed turns cannot be resolved. It is also assumed that all actual roadways are represented in the roadway spatial database and that vehicles must travel on roadways (not, for example, parking lots). If the roadway spatial database is incomplete or incorrect, or if the vehicle travels other than on roadways, ambiguities cannot be resolved and the associated collected vehicle data are ignored by Wiscplow. See Blazquez (2005) for an in-depth analysis of the algorithm's controlling parameters (i.e., buffer size, speed comparison tolerance, and number of consecutive data points to be examined).

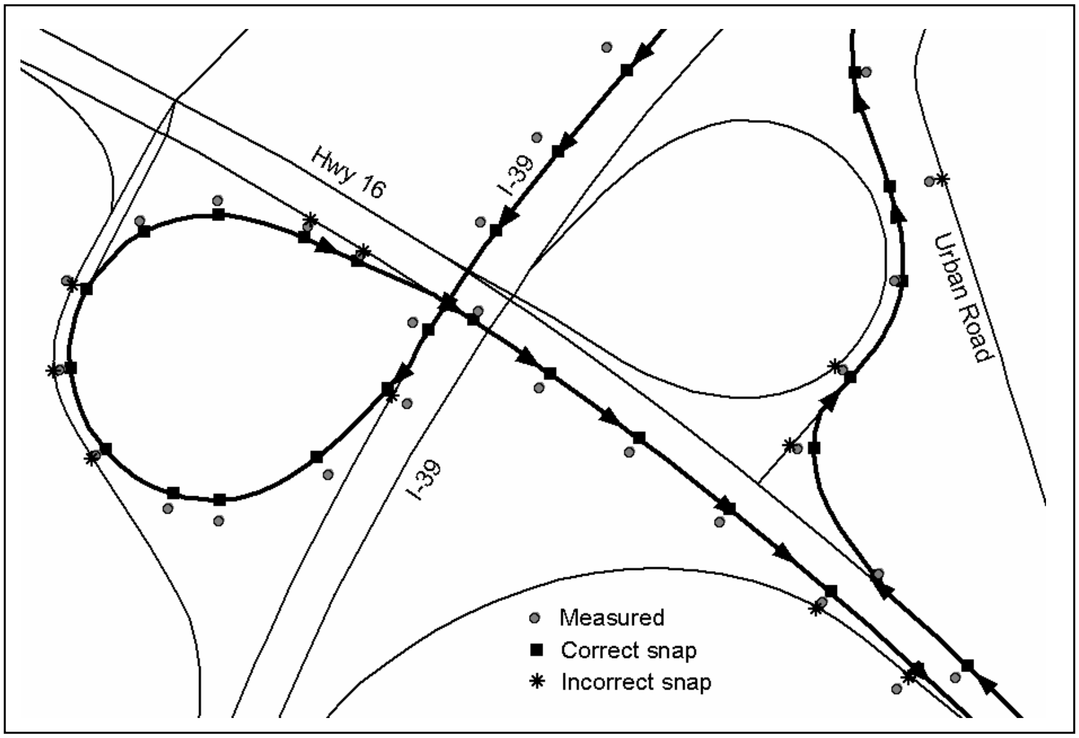


Figure 9.
All Eleven Spatial Ambiguities at This Interchange Were Resolved

5. Development and Testing of Roadway Spatial Databases

The map-matching algorithm requires roadway spatial databases to be structured as "network shapefiles" that contain not only cartographic lines but also have connected topological structures with no gaps or overlaps. Such an underlying network structure, consisting of nodes and directed links, is necessary for pathfinding between vehicle data points, calculation of distances along paths, and representation of allowable turns and directions of travel. Network shapefiles must be attributed with roadway functional class because US and state highways must be distinguished from county and local roads. They must also be attributed with roadway names to support map-based queries. Network shapefiles must also include patrol sections with appropriate identifiers. In addition, earlier analysis had indicated that the positional accuracy requirements of Wiscplow necessitated a source scale for the cartography of 1:24,000 or larger. No participating county maintained a topologically-structured network shapefile of its roadways, although all of them had some digital roadway representation. The counties' roadway spatial databases were of varying accuracy and currency. Subsequently, the Research Team undertook development of network shapefiles, building upon data available from the counties, including existing roadway spatial databases and digital orthophotos.

Initially, this work was expected to entail development of network shapefiles for interstate, US, and state highways only. These are the roadways upon which the counties perform reimbursable winter maintenance for WisDOT. Five counties were initially selected for development: Columbia, Kenosha, Manitowoc, Portage, and Trempealeau. Shortly after work began, Manitowoc and Trempealeau Counties asked to be excused from the project, citing concerns over potential future costs for vehicle and spatial database maintenance. Given a reduction in the number of participating counties, it was decided to expand the scope of network shapefile development to include not only interstate, US, and state highways but also all local roads within each remaining county (Columbia, Kenosha, and Portage). Work for each county entailed 1) data capture and editing, 2) data structuring, 3) quality control, 4) testing with Wiscplow, and 5) development of metadata.

5.1. Columbia County

The original Columbia County roadway centerline spatial database had been compiled photogrammetrically at a source scale of 1:4,800. During 2004 and early 2005, the Columbia County Land Information Office made major updates to the database, resulting from on-the-ground additions and changes to roadways. They also inserted patrol sections with identifiers in the updated spatial database. An example of updates made by the Columbia County Land Information Office appears in Figure 10.

Figure 10.
Old Roadway Centerlines (Left) and Updated Roadway Centerlines (Right) in Columbia County Spatial Database

The Research Team had been working with the original roadway spatial database, having appropriate attributes for use with Wiscplow. Therefore, it was necessary to develop a new network shapefile that had the updated cartography, the attributes of the original spatial database, and the topology necessary to support Wiscplow. A "spatial join" function was used to initiate this. Spatial joins merge attributes from two spatial databases, but only for features that coincide. Since there were non-coincident features in the two databases, due to the updates, attributes of the updated features were added manually by interpretation of a hard-copy roadway map and overlay of the network shapefile with a county-wide digital orthophoto. During this process, the entire network shapefile was visually examined for registration with the orthophoto. Testing of the topology, or network integrity, of the network shapefile, with a pathfinding engine and with Wiscplow, revealed a few topological inconsistencies that were subsequently corrected.

The cartography of the final Columbia County network shapefile is shown in Figure 11. There are two layers: 1) Roadways and 2) Patrol Sections. In Figure 11, patrol sections are highlighted in bold red. Metadata for the network shapefile, including descriptions of all attributes, were prepared according to Federal Geographic Data Committee Content Standards for Geospatial Metadata. These metadata are embedded within the network shapefile database (see Figure 12). The complete Columbia County network shapefile was installed, along with Wiscplow, at

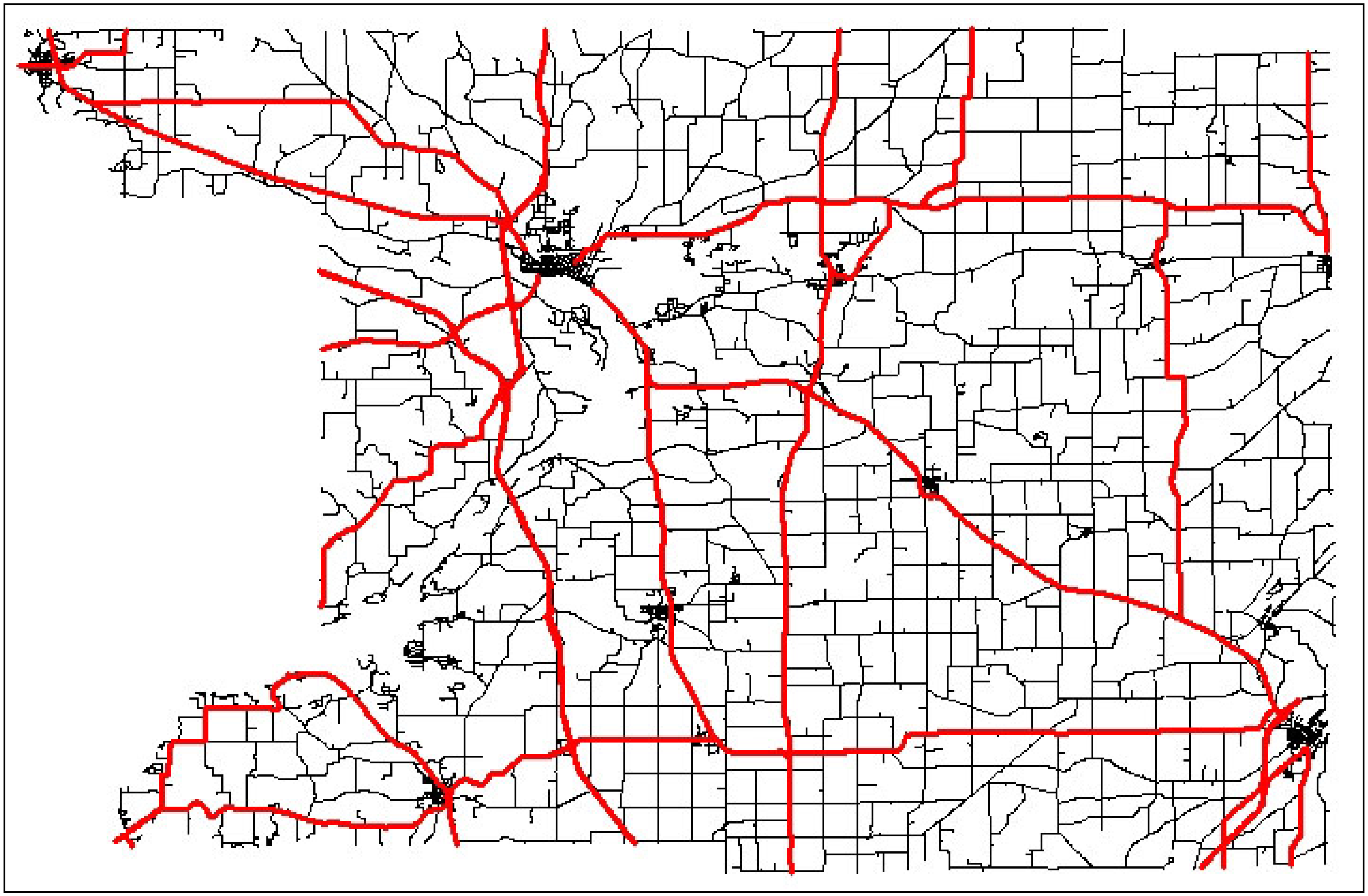
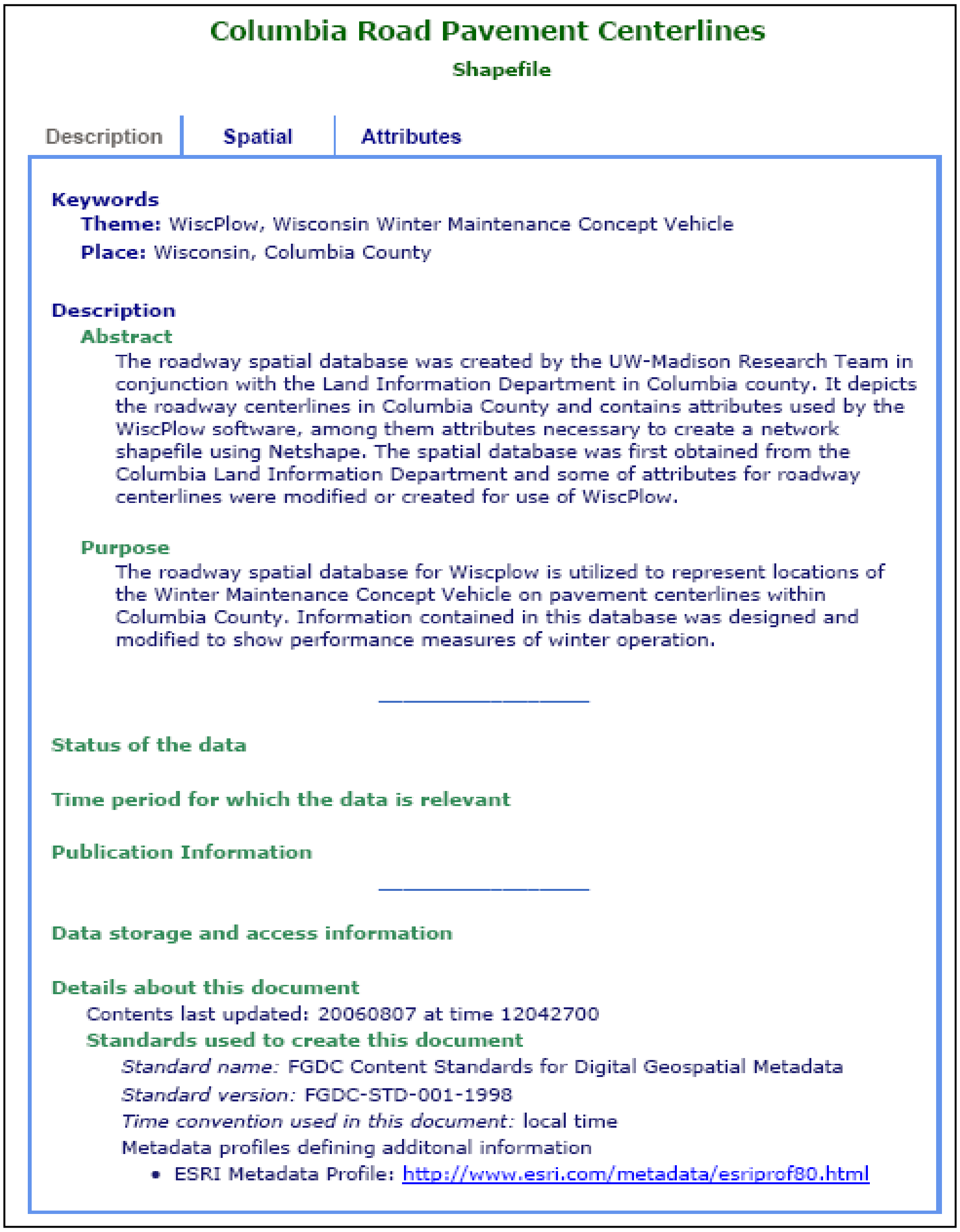


Figure 11.
Columbia County Network Shapefile (Patrol Sections in Bold Red)



Columbia Road Pavement Centerlines

Shapefile

Description | Spatial | Attributes

Keywords
Theme: WiscPlow, Wisconsin Winter Maintenance Concept Vehicle
Place: Wisconsin, Columbia County

Description
Abstract
The roadway spatial database was created by the UW-Madison Research Team in conjunction with the Land Information Department in Columbia county. It depicts the roadway centerlines in Columbia County and contains attributes used by the WiscPlow software, among them attributes necessary to create a network shapefile using Netshape. The spatial database was first obtained from the Columbia Land Information Department and some of attributes for roadway centerlines were modified or created for use of WiscPlow.

Purpose
The roadway spatial database for Wiscplow is utilized to represent locations of the Winter Maintenance Concept Vehicle on pavement centerlines within Columbia County. Information contained in this database was designed and modified to show performance measures of winter operation.

Status of the data

Time period for which the data is relevant

Publication Information

Data storage and access information

Details about this document
Contents last updated: 20060807 at time 12042700
Standards used to create this document
Standard name: FGDC Content Standards for Digital Geospatial Metadata
Standard version: FGDC-STD-001-1998
Time convention used in this document: local time
Metadata profiles defining additonal information
- ESRI Metadata Profile: http://www.esri.com/metadata/esriprof80.html

Figure 12.
A Page of Metadata for the Columbia County Network Shapefile

the Columbia County Highway Department Office (see Section 8). The network shapefile, along with its metadata, are included on a CD attached to the master copy of this report.

5.2. Kenosha County

The original Kenosha County roadway centerline spatial databases had been compiled from large-scale cadastral maps by digitizing centerlines. For a number of years the Kenosha County Land Information Office had been updating the spatial database using survey plans and coordinate geometry (COGO) software. That office estimated the accuracy of the roadway centerlines to be 5-10 feet.

The roadway centerlines were provided in five shapefiles, one each for interstate highways, other US highways, state trunk highways, county trunk highways, and minor roads. Roadway names were provided as annotation, not as attributes. The network shapefile was generated by merging and editing the five separate shapefiles, manually entering roadway names (from the annotation) and other necessary attributes for Wiscplow, and manually entering topological structures for allowable turns and travel directions. Editing consisted of 1) adjusting or re-digitizing centerlines that did not register with the countywide high-resolution orthophoto and 2) making small corrections for topological consistency. Figure 13 shows three examples requiring adjustment or re-digitizing of roadway centerlines. Figure 14 shows two examples requiring editing for topological consistency.

Figure 13.
Roadway Centerlines (Yellow) Requiring Adjustment or Re-Digitizing

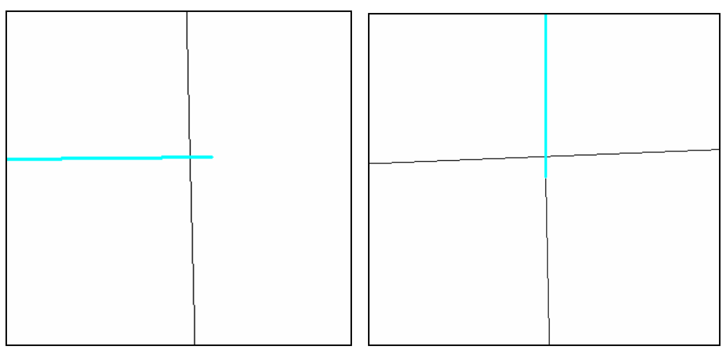


Figure 14.
Topological Problems: An Overshoot (Left) and An Undershoot (Right)

The spatial reference system used by the Kenosha County Land Information Office is the State Plane Coordinate System referenced to the North American Datum of 1927 (NAD (1927)). Concept vehicle coordinates, captured by differential GPS, are latitude and longitude, referenced to the North American Datum of 1983 (NAD (1983)). The vehicle coordinates are transformed, by Wiscplow, from latitude and longitude into state plane map projection coordinates, but Wiscplow cannot perform datum transformations. Thus, the network shapefile was transformed into NAD (1983) HARN State Plane Wisconsin South, placing it on the same datum as the vehicle coordinates.

Patrol sections were embedded in the network shapefile by manually inserting their boundaries based upon a hardcopy map provided by the County Highway Department. Attributes for patrol section identifiers were manually entered.

After editing of the cartography by one of the Research Team members, a second Team member visually inspected the entire data set for registration with the countywide orthophoto. The network integrity, or topological structure, of the network shapefile was checked with a pathfinding engine and with Wiscplow.

The cartography of the final Kenosha County network shapefile is shown in Figure 15. There are two layers: 1) Roadways and 2) Patrol Sections. In Figure 15, patrol sections are highlighted in bold red. Metadata for the network shapefile, including descriptions of all attributes, were prepared according to Federal Geographic Data Committee Content Standards for Geospatial Metadata. These metadata are embedded within the network shapefile database.

The complete Kenosha County network shapefile was delivered to the Kenosha County Land Information Office. The network shapefile, along with its metadata, is included on a CD attached to the master copy of this report.

5.3. Portage County

The original Portage County roadway spatial database was digitized from United States Geological Survey Digital Orthophoto Quarter Quads at a source scale of 1:12,000. Since initial data capture, alignment changes had taken place on one of the major highways and the County Land Information Office had recently completed the corresponding updates of the spatial database. The Research Team was provided with a copy of the updated database and with a countywide, high-resolution digital orthophoto. The County Highway Department provided a hardcopy map of patrol sections.

Portage County performs winter maintenance on a stretch of a state highway extending north into Marathon County. The Marathon County Land Information Office provided a digital cartographic representation of this roadway segment, which was then transformed into the Portage County coordinate system and merged with the Portage County roadway spatial database.

Patrol sections were embedded in the network shapefile by manually inserting their boundaries based upon the hardcopy map. Attributes for patrol section identifiers were manually entered. Other attributes and the topological structure required by Wiscplow were manually entered.

Topological testing with a pathfinding engine and with Wiscplow revealed numerous problems that had to be addressed by editing to ensure network integrity. One type of problem existed at a number of highway interchanges. Crossing highway segments might represent overpasses, underpasses, or at-grade intersections. Nodes must be inserted at at-grade intersections to allow turns. Nodes must be removed at overpasses and underpasses to disallow turns. The orthophoto served as a visual reference in resolving these problems.

In some places, roadway segments were missing from the database and were manually digitized from the orthophoto (see Figure 16). In other places, all topologically necessary roadway segments existed, but did not allow for all possible ways of travel through the network. These segments had to be modified or re-digitized to correct the problems (see Figure 17).

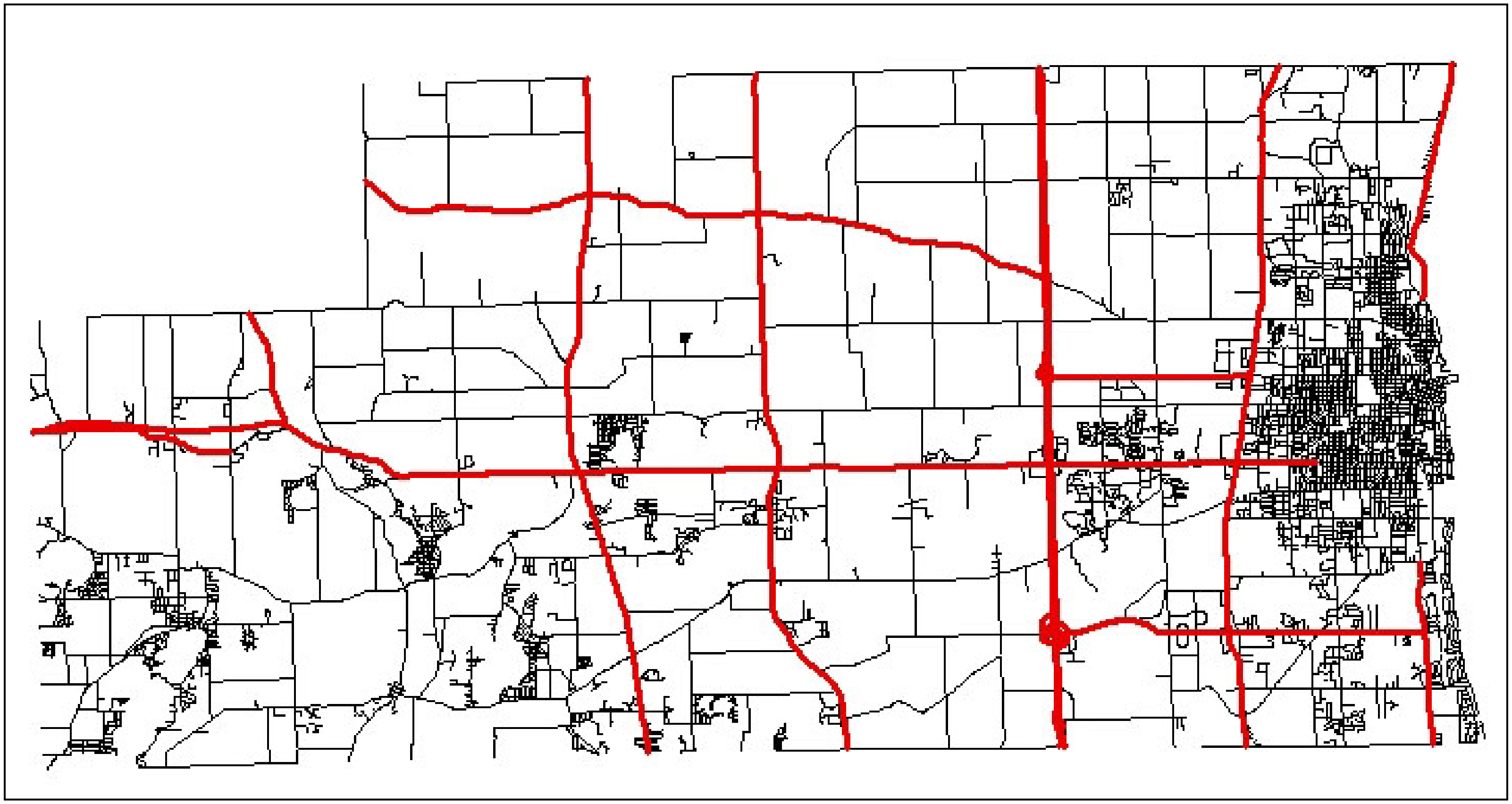
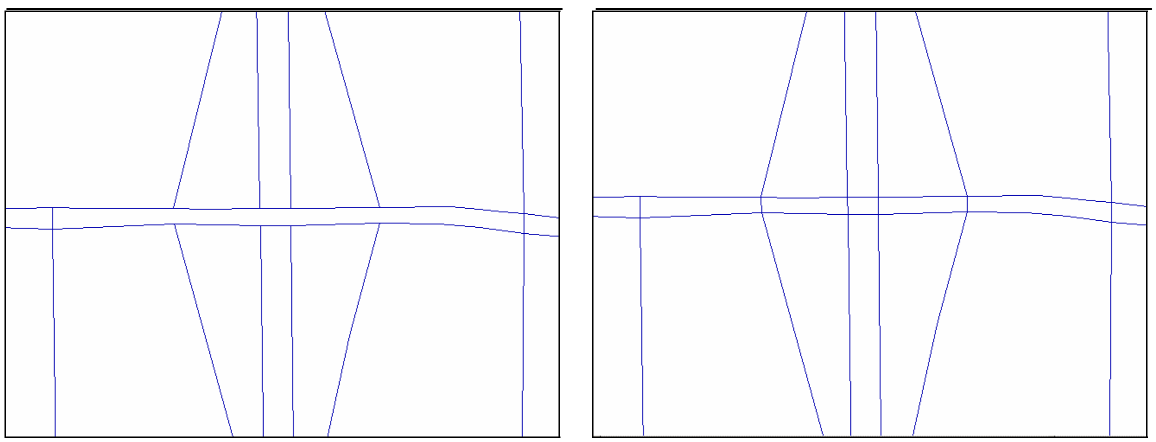


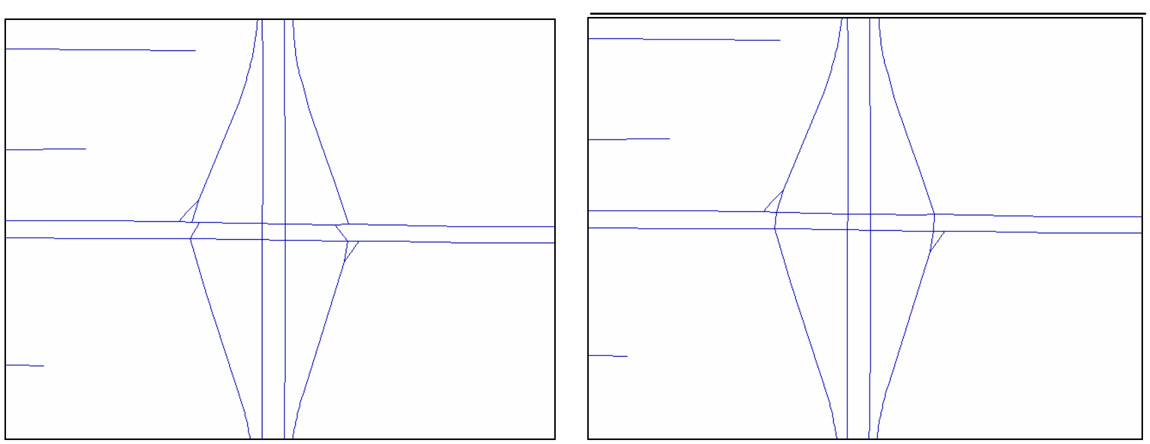
Figure 15.
Kenosha County Network Shapefile (Patrol Sections in Bold Red)



BEFORE: Roadway centerlines necessary to allow turns on and off of ramps are omitted from dataset. Two small segments of the freeway are omitted.

AFTER: Roadway centerlines are digitized from orthophotos, allowing travel on the highway and all possible turns on and off the highway.

Figure 16.
Missing Roadway Segments

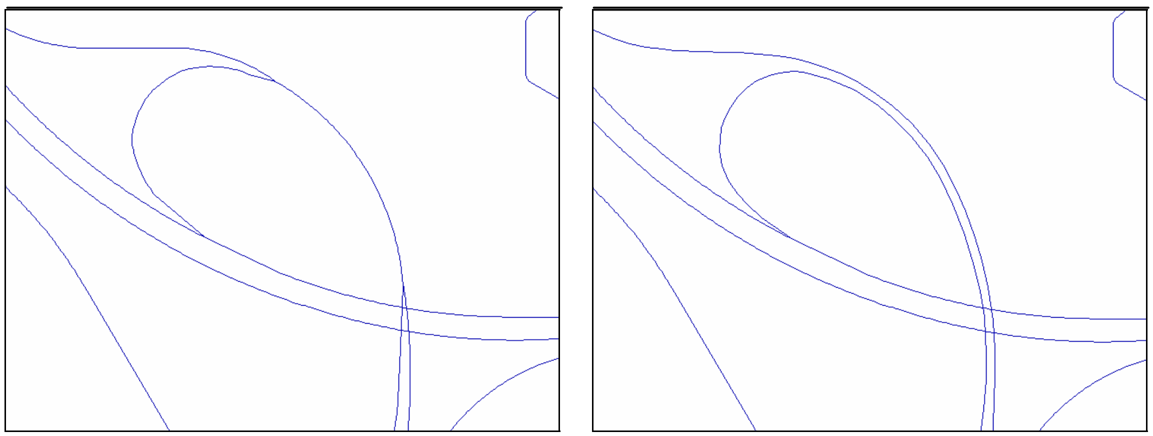


BEFORE: Travel from off-ramps directly back to on-ramps is not permitted.

AFTER: Travel from off-ramps directly back to on-ramps is permitted, and all other necessary topology is preserved.

Figure 17.
Roadway Segments Require Modification

In other places, the manner in which roadways were represented could not be integrated into an accurate roadway network. The most common problem of this nature was encountered when a single roadway centerline was used to represent two adjacent, unconnected lanes of traffic flowing in opposite directions. This often occurred on ramps (see Figure 18). There were also a number of undershoots and overshoots, as with the Kenosha County data.



BEFORE: Two lanes of on-ramp/off-ramp structure, separated by a physical barrier, are improperly represented as one linear feature with bidirectional traffic flow.

AFTER: The single linear feature is re-digitized as two features (using orthophotos), each allowing only unidirectional flow.

Figure 18.
Ramp Requires Re-Digitizing to be Topologically Correct

After editing of the cartography by one of the Research Team members, a second Team member visually inspected the entire data set for registration with the countywide orthophoto. The network integrity of the final network shapefile was confirmed with a pathfinding engine and with Wiscplow.

The cartography of the final Portage County network shapefile is shown in Figure 19. There are two layers: 1) Roadways and 2) Patrol Sections. In Figure 19, patrol sections are highlighted in bold red. As with Columbia and Kenosha Counties, metadata for the network shapefile, including descriptions of all attributes, were prepared according to Federal Geographic Data Committee Content Standards for Geospatial Metadata. These metadata are embedded within the network shapefile database.

The complete Portage County network shapefile was installed, along with Wiscplow, at the Portage County Highway Department Office (see Section 8). The network shapefile, along with its metadata, is included on a CD attached to the master copy of this report.

6. Completion and Testing of Wiscplow Functionality

6.1. Wiscplow Functionality

Wiscplow is intended for use, by analysts and other technical staff, in a post-processing environment. It is a standalone application, not an enterprise solution. It has no real-time monitoring capability and is not portable, as it requires access to databases that are typically only available on client-server networks. Wiscplow presents users with a ten-tab interface (see Figure 20). The tabs control data entry, data analysis and reporting, and display functions.

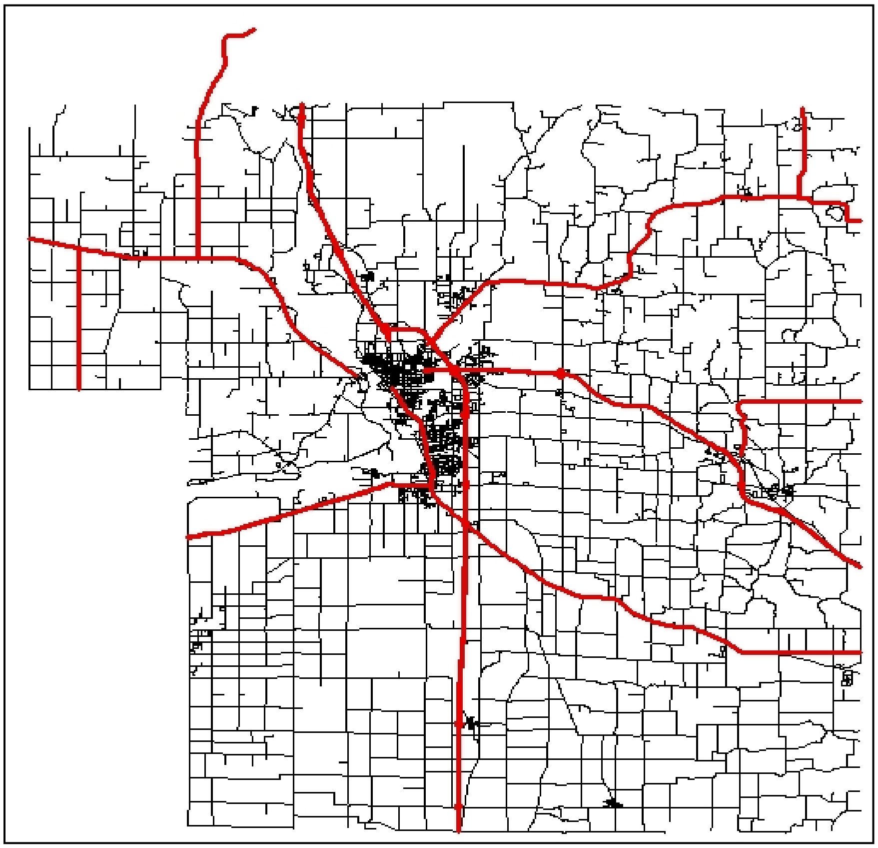


Figure 19.
Portage County Network Shapefile (Patrol Sections in Bold Red)

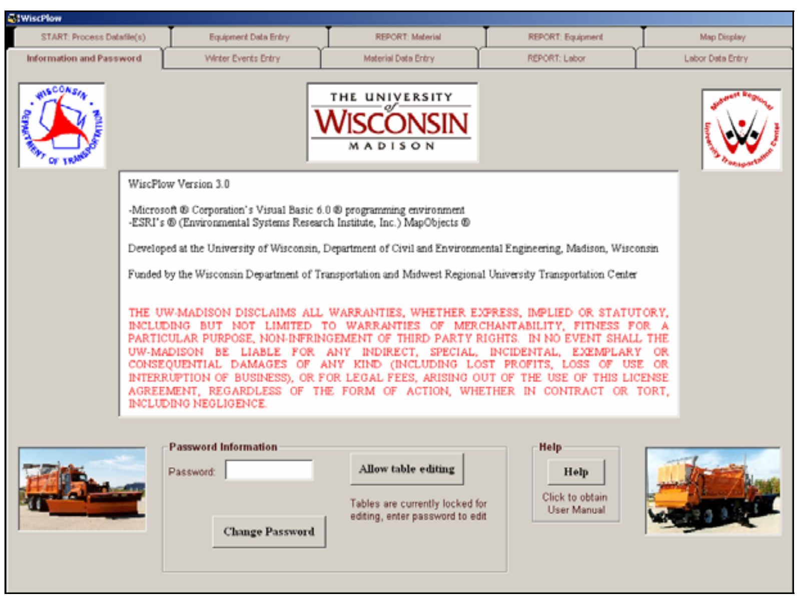


Figure 20.
Wiscplow User Interface with Ten Tabs

6.1.1. Information and Password Tab

Figure 20 shows the Information and Password Tab. The tab displays acknowledgements and disclaimers, presents users with a “Help” button, and has functions for entering and changing passwords. The “Help” button opens a 109-page Microsoft Word® file of user documentation (see Section 7.2). There is no searchable index or “help” wizard.

Entry of a password allows entry and modification of material, equipment, and labor data. After entry or modification of these data, users may lock the database for security by returning to the Information and Password Tab and pushing the appropriate button.

6.1.2. Winter Events Entry Tab

The Winter Events Entry Tab appears in Figure 21. Here, users enter data on storm events, incidents, and anti-ice events. Wiscplow automatically assigns identifiers to all events. For each type of event, displayed data tables contain start dates and times, end dates and times, and codes describing events. Modifications to existing data require deletion of appropriate records and entry of new data. Data in the tables can be selected for deletion, but they cannot be edited by typing on a keyboard.

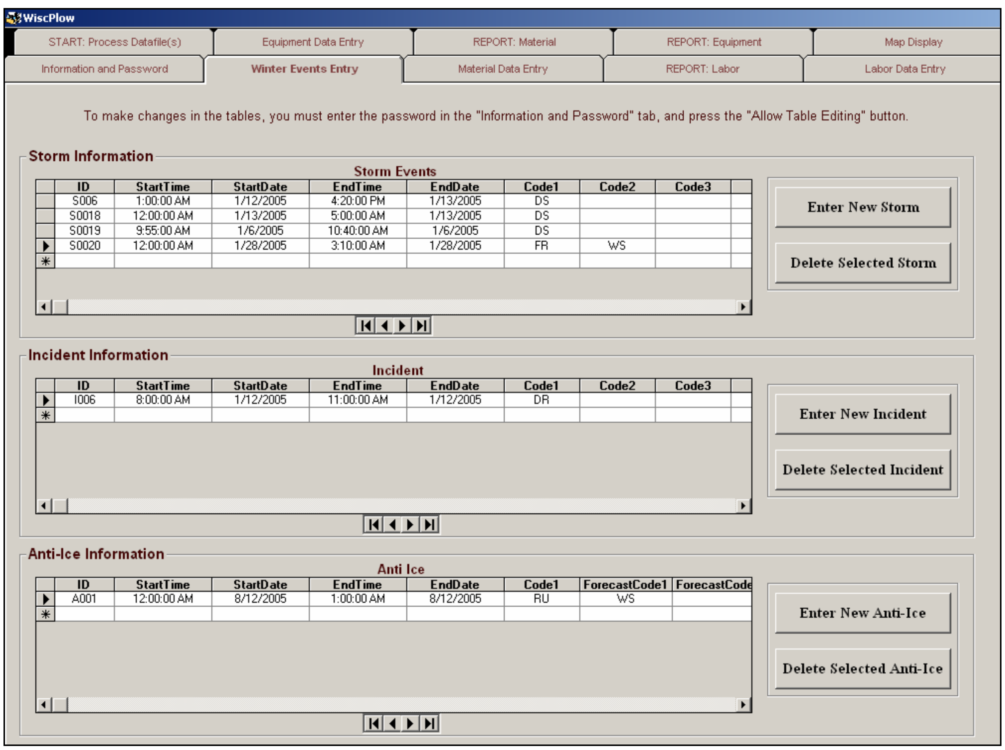


Figure 21.
Winter Events Entry Tab

When new data are entered by pushing, for example, the “Enter New Storm” button, a data entry screen appears (see Figure 22). Here, dates are picked from calendars, times are picked from scrolled lists, and codes are chosen from descriptive “pick lists”. Distinctions are made between routine and forecasted anti-ice events.

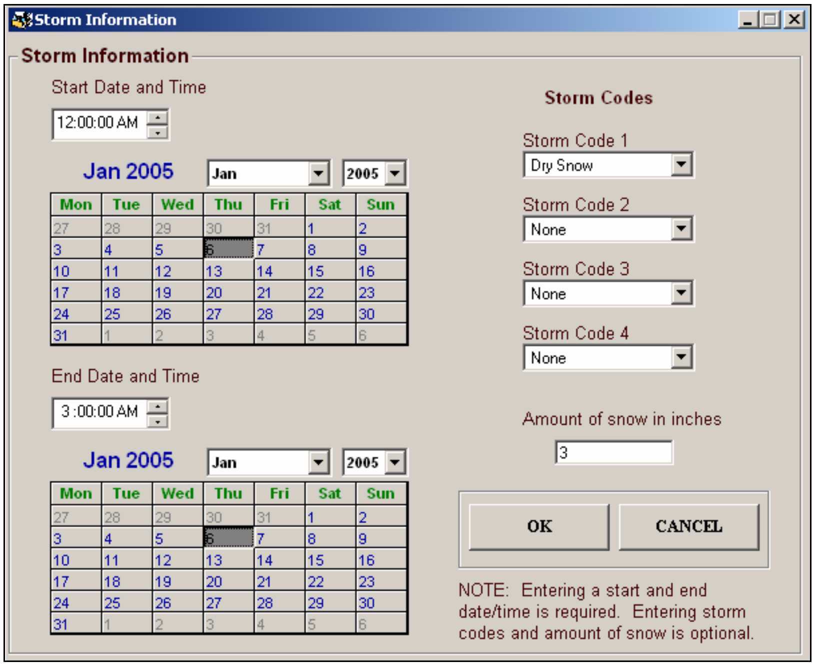


Figure 22.
Storm Data Entry Screen

6.1.3. Material Data Entry Tab

The Material Data Entry Tab appears in Figure 23. The tab displays data for dry agents and liquid agents. Existing data can be modified by deletion of existing, and entry of new, full data records. Pushing the “Enter New Product” button causes a data entry screen to appear, where new agents and associated codes can be chosen from pick lists.

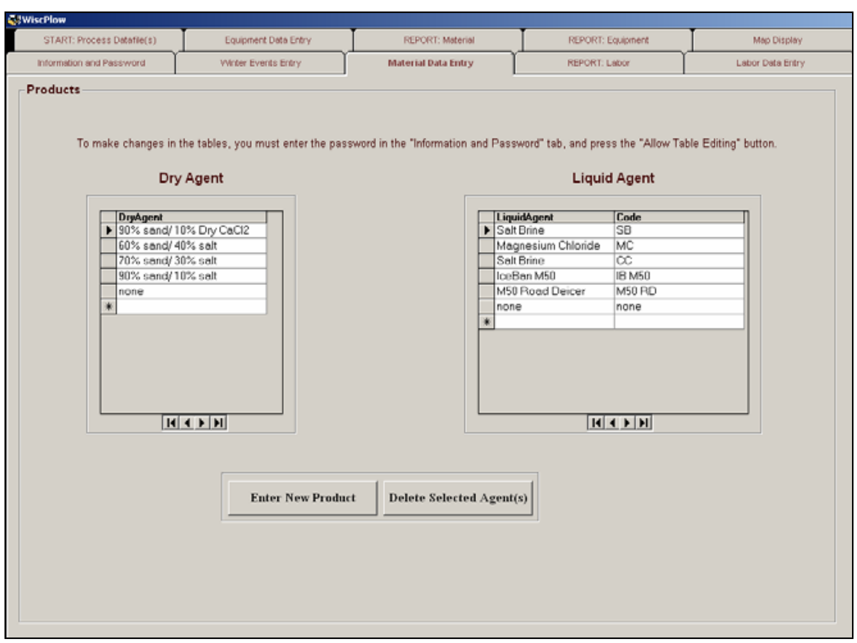


Figure 23.
Material Data Entry Tab

6.1.4. Labor Data Entry Tab

The Labor Data Entry Tab appears in Figure 24. Data on operators and operating days are displayed, along with a “View Labor Rates Information” button. Data in the tables can be modified by deletion of existing, and entry of new, full records.

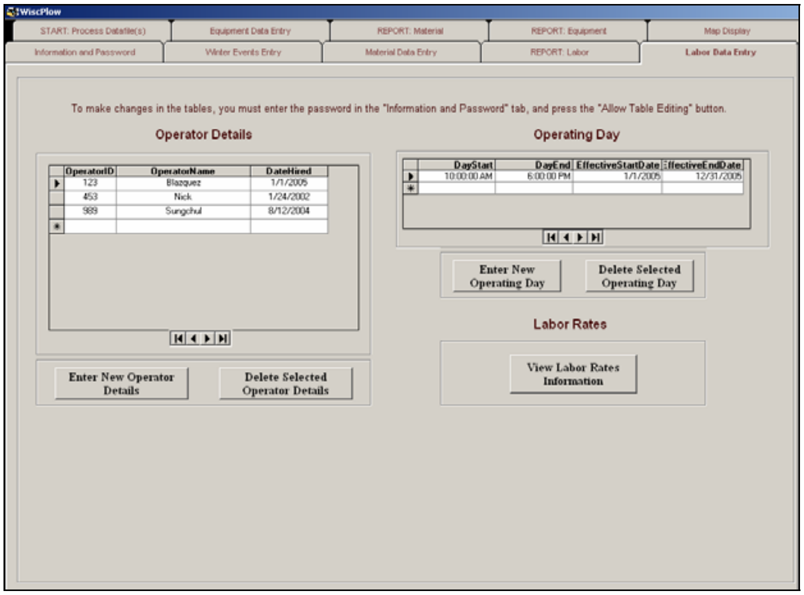


Figure 24.
Labor Data Entry Tab

Pushing the “Enter New Operator Details” button cause the screen in Figure 25 to appear. Here, users identify operators by name and ID and pick the associated hiring date from a calendar. Hiring dates are necessary because some labor rates depend upon seniority.

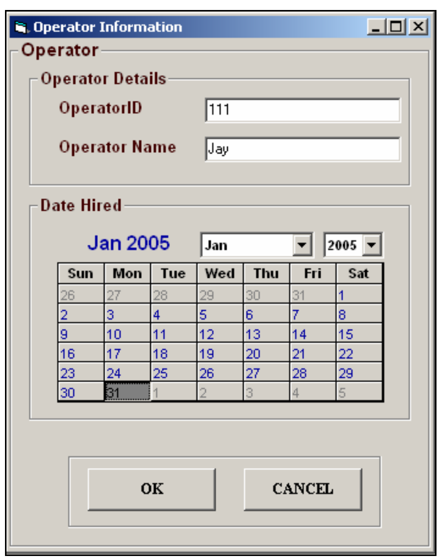


Figure 25.
Operator Information Screen

Pushing the “Enter New Operating Day” button on the Labor Data Entry Tab causes the screen in Figure 26 to appear. Here, the start and end times for an operating day, and the effective start and end dates for that operating day are picked from scrolled lists and calendars, respectively. This information is necessary to assign regular and overtime wages to work times reported by the vehicles.

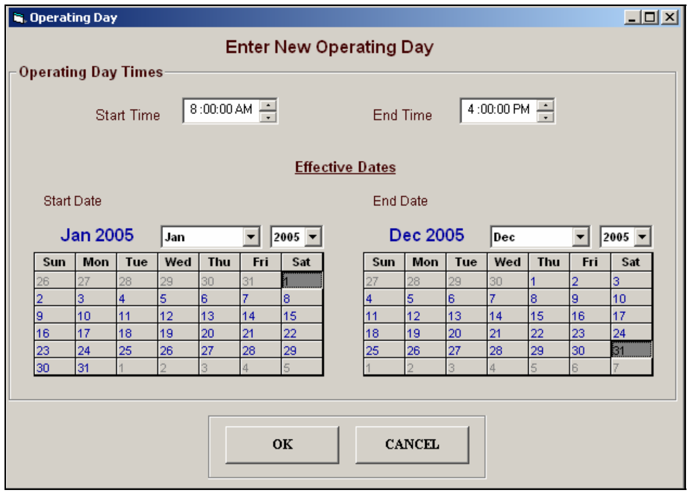


Figure 26.
Enter New Operating Day Screen

Pushing the “View Labor Rates Information” button on the Labor Data Entry Tab causes a county-dependent labor rates entry screen to appear. The screen for Columbia County is shown in Figure 27. Data in the table can be modified by deletion of existing, and insertion of new, full records. Effective start and end dates for labor rates are chosen from calendars, and hourly wages, (in Columbia County’s case) by seniority, are typed into appropriate boxes. The Labor Rate Setup screen is tailored to each county’s labor contract.

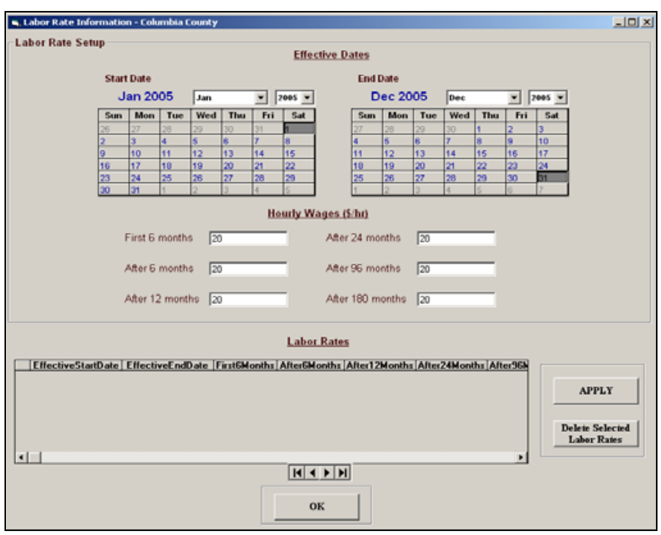


Figure 27.
Labor Rate Setup Screen for Columbia County

6.1.5. Equipment Data Entry Tab

The Equipment Data Entry Tab appears in Figure 28. Tables of data on equipment configurations and costs are displayed. These tables contain data on how vehicles are configured, by equipment class, and costs per hour for equipment classes, respectively. The costs-per-hour table includes effective start and end dates because these costs vary over time. Existing data can be modified by deleting full records and entering new data. Pushing the “Enter New Data” button, for either configuration or cost, causes an appropriate data entry screen to appear.

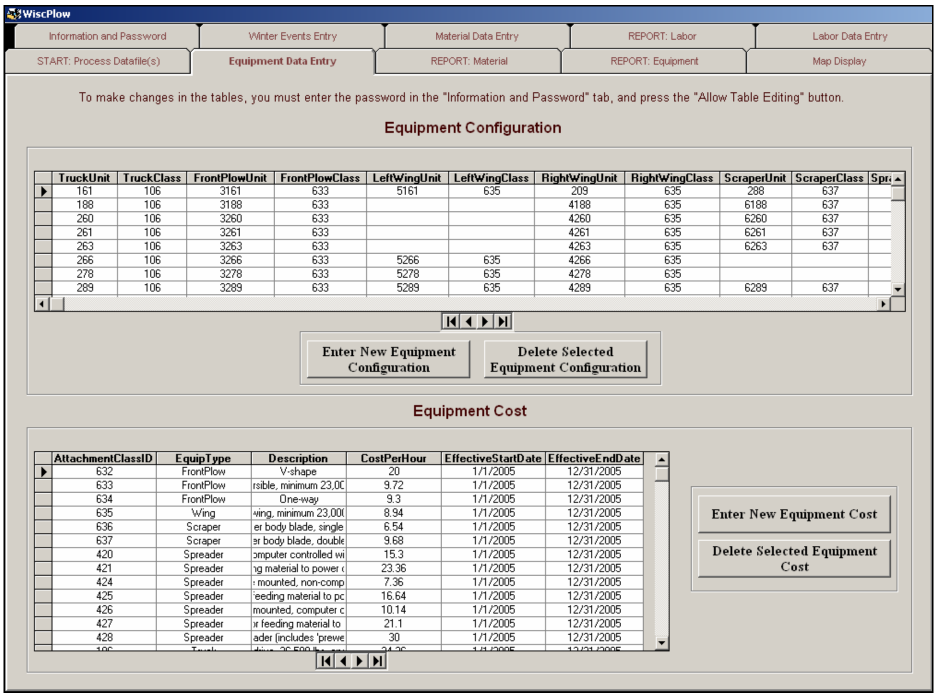


Figure 28.
Equipment Data Entry Tab

6.1.6. START - Process Datafiles Tab

The START – Process Datafiles Tab appears in Figure 29. At this tab, users select vehicle data files for processing and associate vehicle data with vehicles, materials, operators, and winter events. Vehicle data files are recorded on-board the vehicles on PCMCIA cards that are later downloaded for processing by Wiscplow. The first few records of a typical vehicle data file appear in Figure 30. The records contain information on time, date, latitude, longitude, vehicle speed, material spreading rates, equipment status, and air and pavement temperature. These records are formatted according to Raven’s DCS710 or DCS710A console standard.

When the “Add a file to process list” button is pushed, a browsing screen appears and the user navigates to the appropriate folder and selects the data file to add to the list. This activates the “Details” box on the START – Process Datafiles Tab (see Figure 31). The user then associates information with the selected vehicle data file by choosing from pick lists in the “Details” box. When all details have been picked, the user presses the “Apply Details” button, invoking indicated relationships among the data and displaying them in the updated list of data files to process. Any number of data files can be added to the process list in this manner. Wiscplow is structured in this way because large data files require significant processing time. After a long storm event, analysts might have many large data files to process. These can be placed in a long list that can then be processed in batch mode overnight. Data files can be deleted from the process list. It is also possible to remove, from the existing database, other data files that have been previously processed.

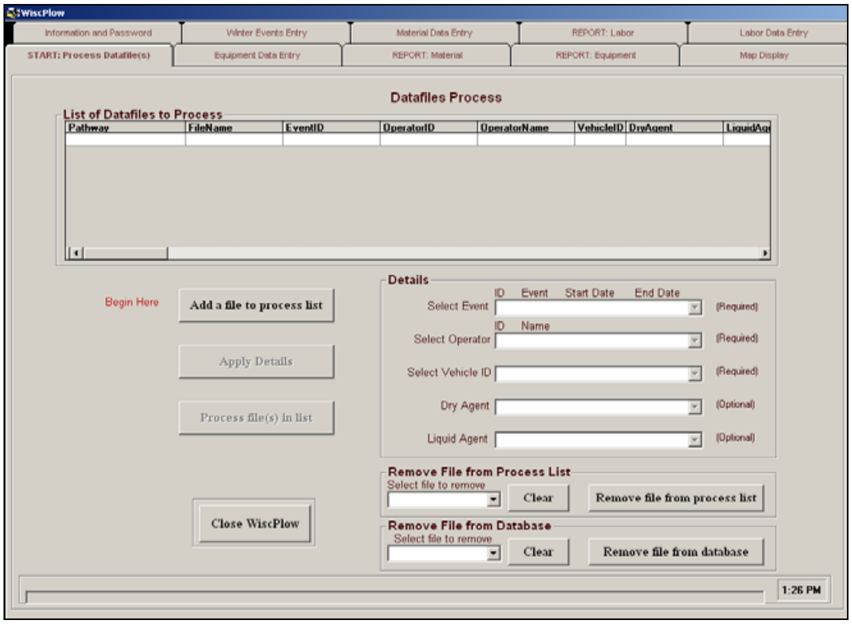


Figure 29.
START – Process Datafiles Tab

```
6/5/0,-,104A

DCS710A
M10,ResDrvC,ResDrv,AgDrvC,AgDrv,CommDrvC,CommDrv,Bridge,Culvert,StopSgn,OtrSgn,NoPass,RdRpr,ShldRpr,Asph,Seal,Estimate

M12,89
M13,8:17:41,0.00000,0.00000,0.0,A,N
M1,8:17:41, , ,0, , , , , , ,0,0.00000,0.00000,0.0
M19,8:17:41,mt,52,52,0,off,off,off,0,off,off,off,off,off,off
M22,8:17:41,s1,off,off,off,off,off,0,off,off,off,off,off,5,5,0,2
M23,8:17:41,ps,down,no,no,down,down,down,down,down,down,down,no,no,no,no,0.00000,0.00000,0.0
M13,8:17:41,0.00000,0.00000,0.0,A,N
M6,8:17:41,c1,2500,1200,20,260,20,20,3350,143,136
M7,8:17:41,c2,200,200,600,800,1000,1200,1400,1600,1800,2000,12
M8,8:17:41,c3,100,200,300,400,500,600,700,800,900,1000,12,0
M17,8:17:45,c4,1500,0,200,200,600,800,1000,1200,1400,1600,1800,2000,12
M18,8:17:45,c5,1500,0,200,200,600,800,1000,1200,1400,1600,1800,2000,12
M20,8:17:45,c6,60,0,143,20,25,30,35,40,45,50,55,60,65
M21,8:17:46,c7,10,0.0,0.0,12,144,144,144
```

Figure 30.
The First Few Records of a Typical Vehicle Data File

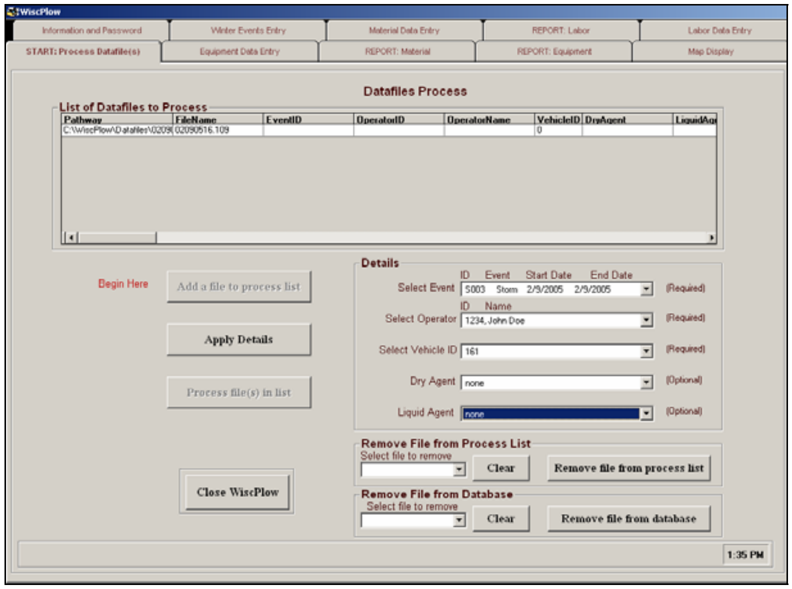


Figure 31.
Associating a Vehicle Data File with a Vehicle, an Operator, a Storm Event, and Materials

Pressing the "Process file(s) in list" button initiates a sequence of operations on each data file in the list. The records are parsed and placed in a database that includes the associations with vehicle, operator, storm event, and materials indicated earlier by the user. Latitudes and longitudes of vehicle data points are transformed into the map projection coordinates of the roadway network shapefile. The map-matching algorithm then associates the vehicle data points with roadways and patrol sections. These associations are stored in the database. A point shapefile of the vehicle data points is also produced. The vehicle data points can then be viewed and queried at the Map Display Tab.

6.1.7. Map Display Tab

At the Map Display Tab, users can display the roadway network shapefile, the patrol section shapefile, and any number of vehicle data shapefiles. Each shapefile is considered to be a "layer" that can be added, turned on and off, or deleted from the display. Figure 32 shows the Map Display Tab with Columbia County's roadway network and patrol sections. There are a set of function buttons just above and to the left of the map. These include pan and zoom, add and remove layers, print, measure distance, and identify functions. The "identify" function can be used to query any feature in the map display for its underlying attributes. For example, in Figure 33 the user has added a vehicle data shapefile, zoomed in, "identified" a particular vehicle data point, and obtained the names of the roadway and patrol section upon which the vehicle was traveling, vehicle speed, operator name, environmental data, date and time, equipment status, and material spreading rates. Multiple vehicle data files can be viewed simultaneously, showing relationships between vehicle tracks and activities.

6.1.8. REPORT: Material Tab

The REPORT: Material Tab appears in Figure 34. At this tab users can request reports on up to 19 performance measures: ten for application rates, seven for inventory, and one each for

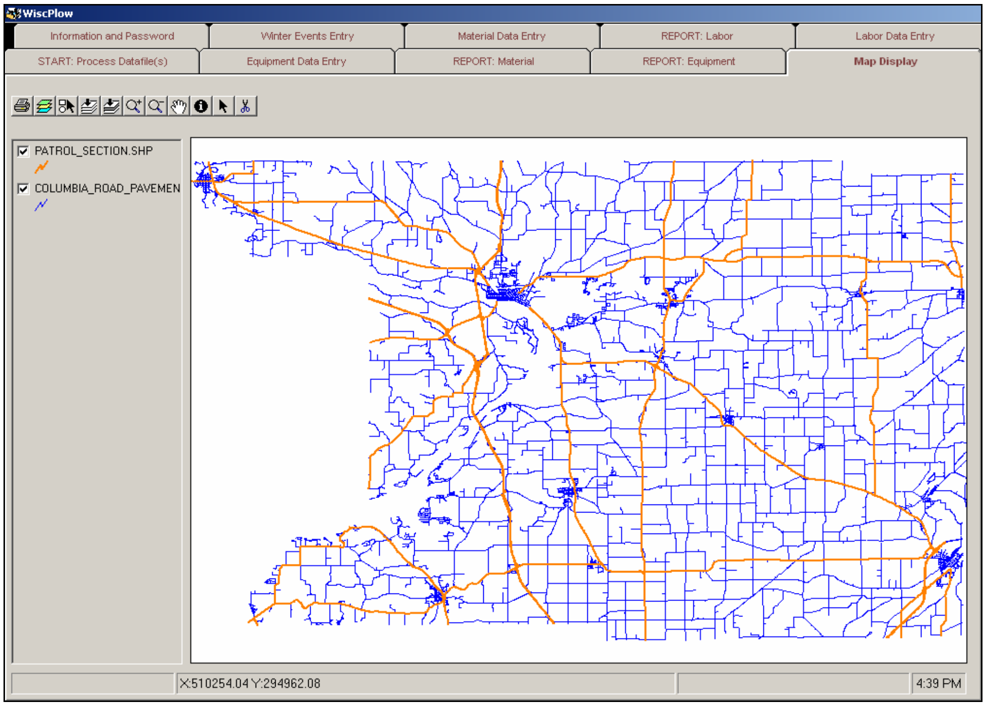


Figure 32.
Map Display Tab with Function Buttons and Columbia County Roadway Network and Patrol Sections

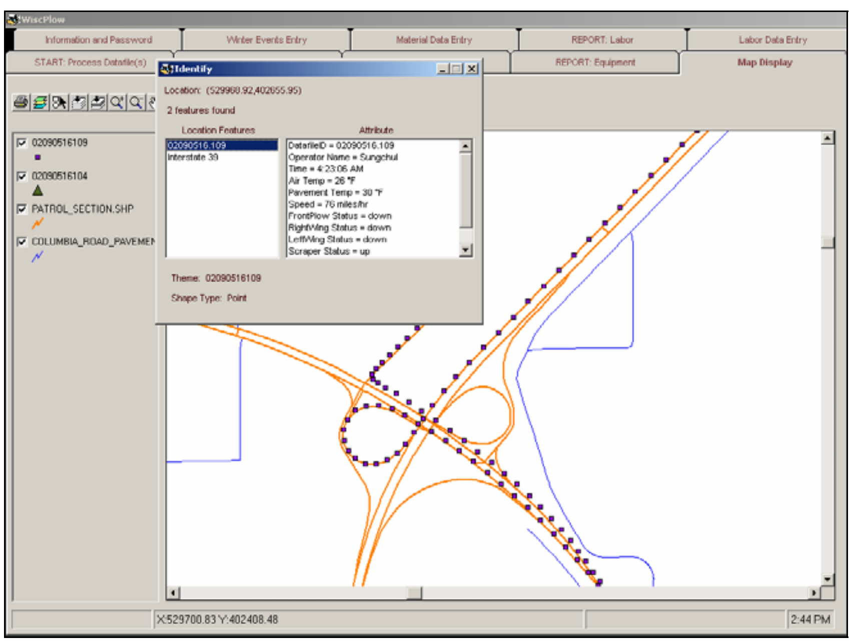


Figure 33.
Zoomed Map Display with Vehicle Data Points and Result of a Query Using the “Identify” Function

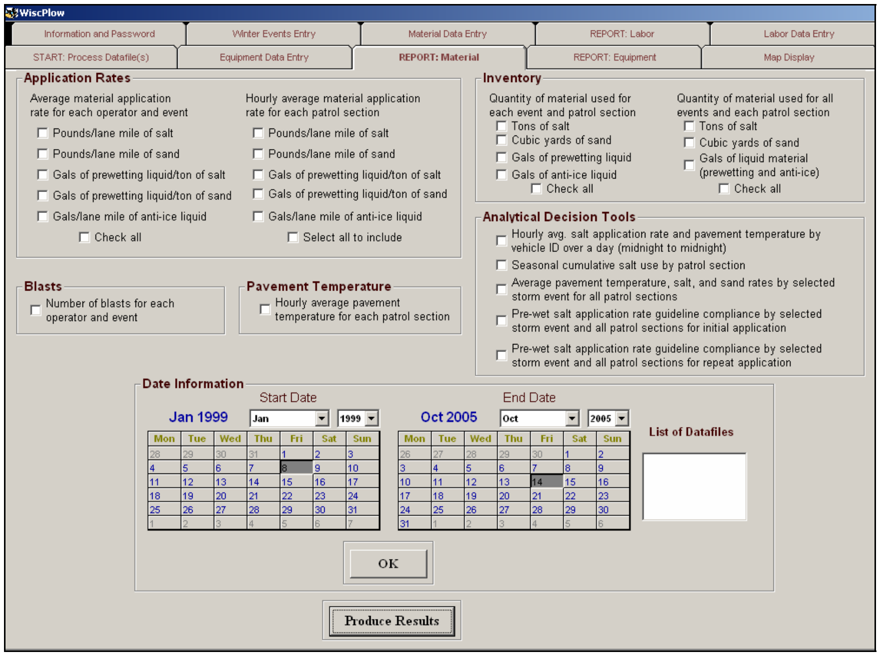


Figure 34.
REPORT: Material Tab

blasts and pavement temperature. Users may also request up to five analytical decision tools (i.e., charts) that show relationships among performance measures over time and space. Start date and end date calendars are included for selection of time periods to be covered by the reports and analytical decision tools. An example report is shown in Figure 35 and an example analytical decision tool is shown in Figure 36. Reports can be printed or stored as rich text format (.rtf) files.

Columbia
Automated Winter Storm Reports: Total Salt used for each Patrol Section for each Storm For the Period of 1/9/2001 through 2/19/2001

StormID	PtrlSctnID	Total Salt used in Tons
S0012	106	16.79
S0012	107	0.32
S0012	110	0.82
S0023	106	8.02
S0023	107	0.14
S0023	110	0.23
S0031	107	0.00
S006	106	4.01
S006	107	0.07
S006	110	0.11

Figure 35.
Example Material Report

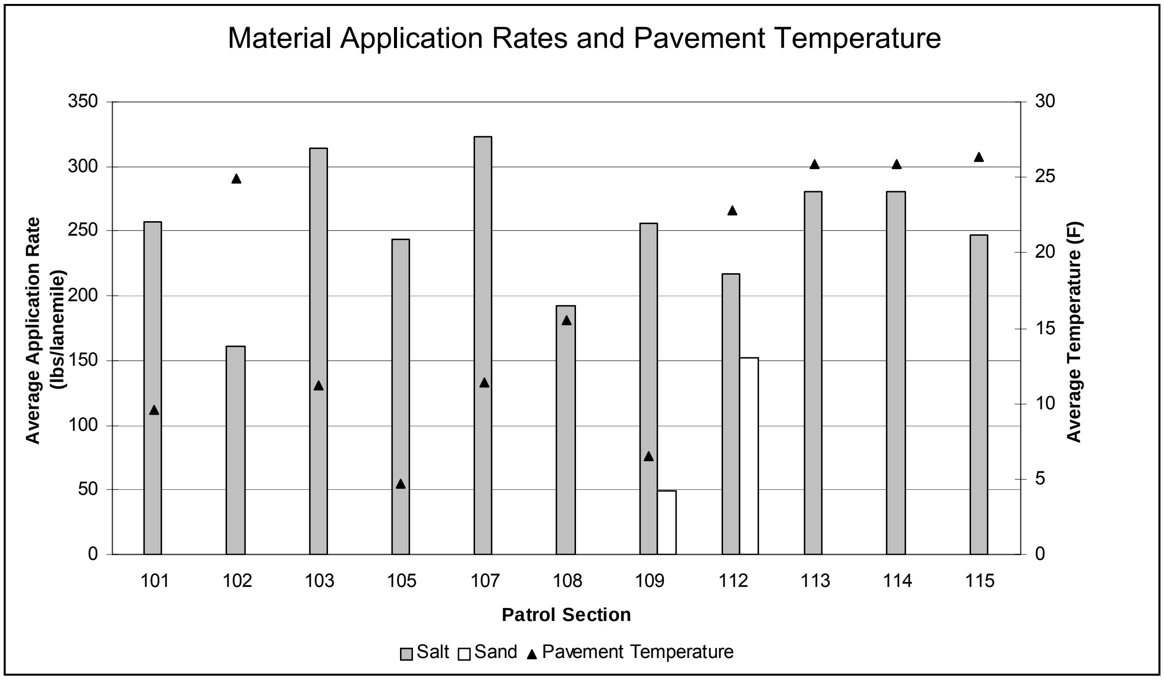


Figure 35.
Example Material Analytical Decision Tool

Analytical decision tools can be printed or stored as Microsoft Excel® spreadsheets.

6.1.9. REPORT: Equipment Tab

The REPORT: Equipment Tab appears in Figure 36. Users can request up to ten reports and

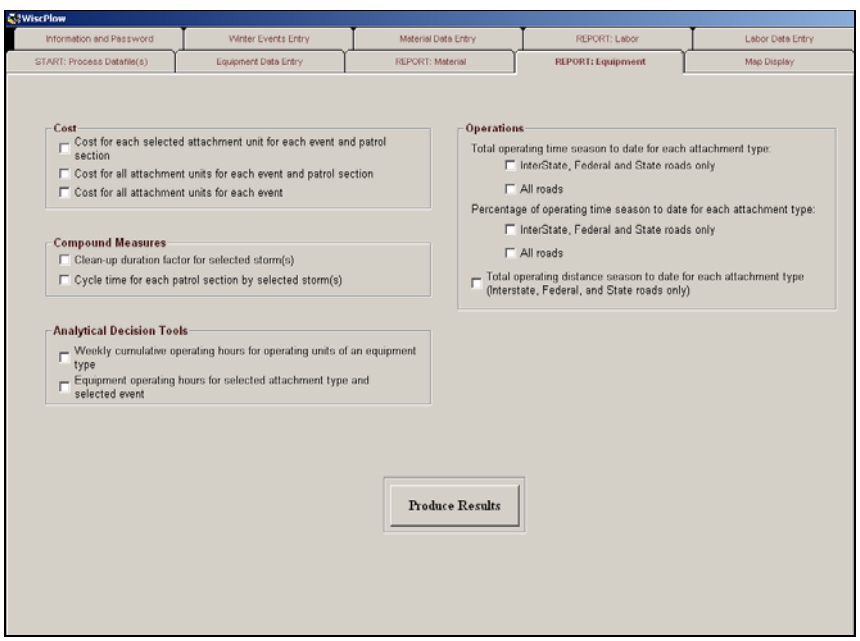
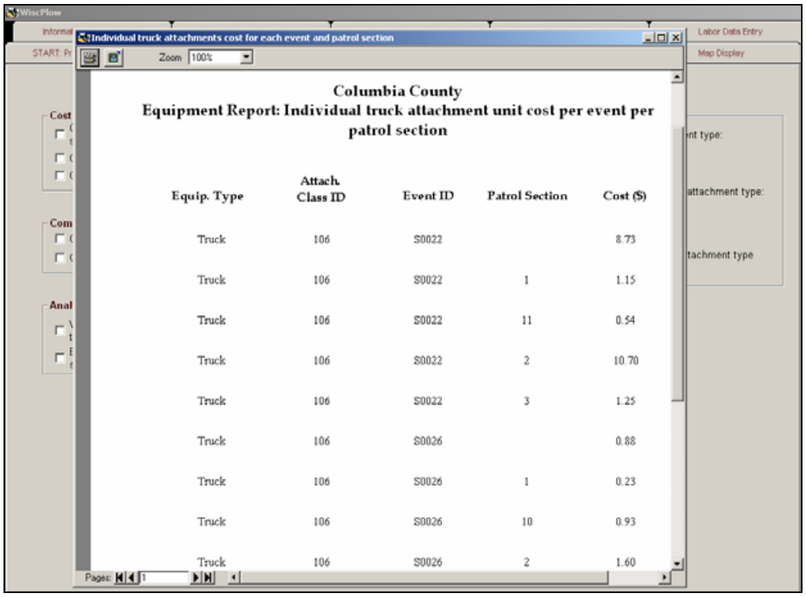


Figure 36.
REPORT Equipment Tab

Up to two analytical decision tools. Figure 37 shows an example report and Figure 38 shows and example analytical decision tool.



Columbia County
Equipment Report: Individual truck attachment unit cost per event per patrol section

Equip. Type	Attach. Class ID	Event ID	Patrol Section	Cost ($)
Truck	106	S0022		8.73
Truck	106	S0022	1	1.15
Truck	106	S0022	11	0.54
Truck	106	S0022	2	10.70
Truck	106	S0022	3	1.25
Truck	106	S0026		0.88
Truck	106	S0026	1	0.23
Truck	106	S0026	10	0.93
Truck	106	S0026	2	1.60

Figure 37.
Example Equipment Report

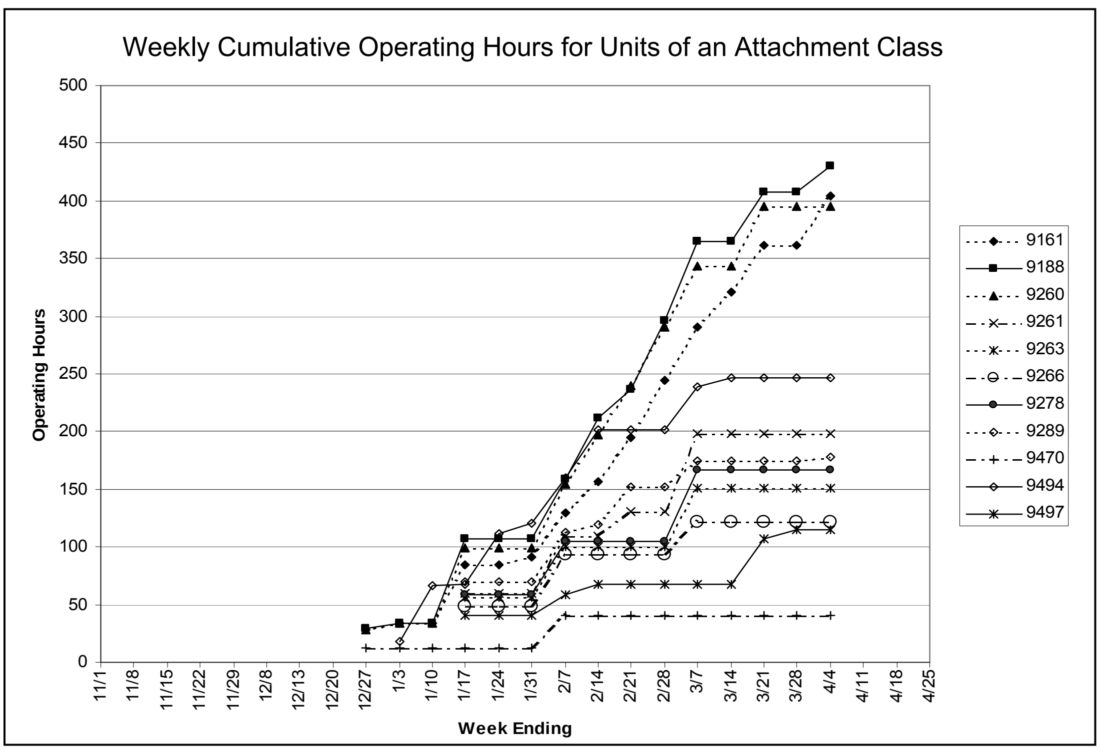


Figure 38.
Example Equipment Analytical Decision Tool

6.1.10. REPORT: Labor Tab

Figure 39 shows the REPORT: Labor Tab. Users can request reports on up to five performance measures. Figure 40 shows an example labor report.

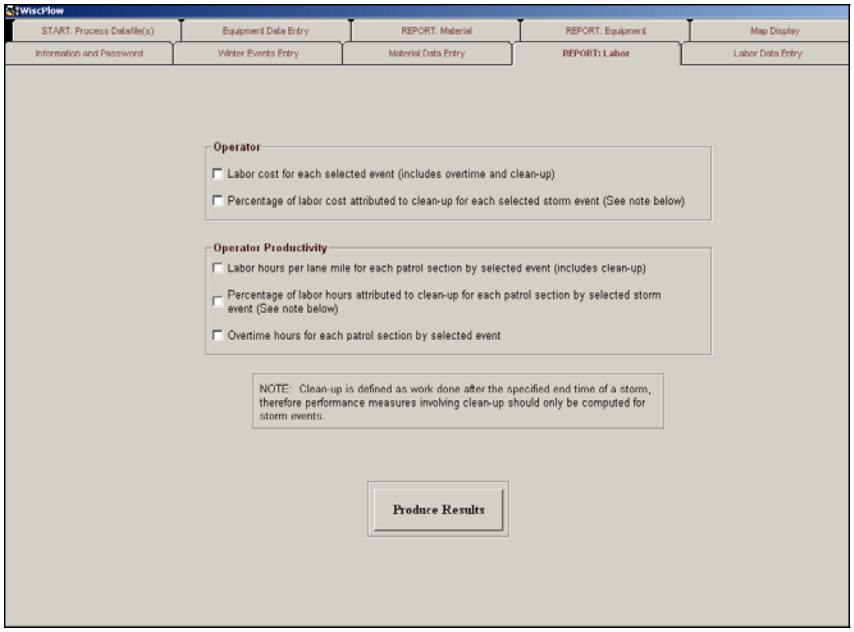
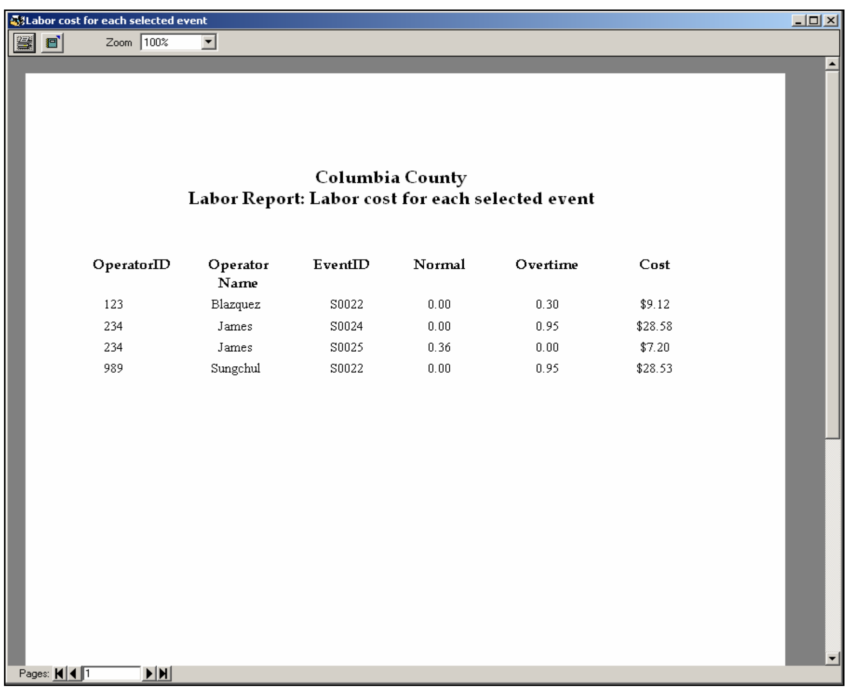


Figure 39.
REPORT: Labor Tab



Columbia County
Labor Report: Labor cost for each selected event

OperatorID	Operator Name	EventID	Normal	Overtime	Cost
123	Blazquez	S0022	0.00	0.30	$9.12
234	James	S0024	0.00	0.95	$28.58
234	James	S0025	0.36	0.00	$7.20
989	Sungchul	S0022	0.00	0.95	$28.53

Figure 40.
Example Labor Report

6.2. Testing and Validation of Wiscplow

Two series of tests were applied to Wiscplow's functionality. The first series was used to validate the system's computational integrity, to ensure that performance measures were being computed correctly, and that decision management tools were producing correct results. This was done by selecting a few small vehicle data files for Columbia and Portage Counties. Wiscplow was used to parse the data files into database tables. Then, the data in the tables were used in hand computations and spreadsheets to produce check values for performance measures and inputs to decision management tools. These check values were then compared to corresponding values produced automatically by Wiscplow. Due to hardware difficulties, Kenosha County had not collected any vehicle data files. Therefore, the Research Team generated fictitious, but representative, vehicle data files for Kenosha, developed check values from them, and compared the check values to Wisplow's results when using the Kenosha County Roadway network shapefile and patrol sections.

The second series of tests was used to ensure the integrity of the user interface, to prevent or trap user errors, and to ensure that execution-time errors, due to invalid data values, would not occur. This was done by having a member of the Research Team, who was not involved in development of Wiscplow source code, attempt to make the program fail by simulating as many user mistakes as could be identified. When error-trapping or user interface flaws were detected, the source code was modified to correct them.

7. Technical and User Documentation

Detailed technical documentation is necessary not only to describe the internal workings of software to interested readers but also to ensure that the system can be maintained in the future by other than the original developers. Detailed user documentation is necessary to ensure utility of software in the business place.

7.1. Technical Documentation

Wiscplow's technical documentation has an internal component and an external component.

7.1.1. Internal Technical Documentation

Wiscplow's internal technical documentation consists of extensive comments inserted within the Visual Basic® source code. Figure 41 shows a small portion of Wiscplow's source code with comments highlighted in bold type face. Figure 42 shows how comments and source code relate to Wiscplow's user interface. The source code consists of 23 rich-text-formatted files that are included on a CD along with the master copy of this report.

Wiscplow's spatial analytical functionality is based upon Environmental Systems Research Institute's (ESRI) Map Objects®. The installed version of Wiscplow consists of a large executable module and a number of dynamic linked libraries, text files, and bitmaps. A full installation CD for Wiscplow is included with the master copy of this report.

Wiscplow requires access to the following Microsoft products:

1. Excel® (for decision management tools).

2. Access® (for database management).

Command1: OK

```
Private Sub Command1_Click() 'OK button
On Error Resume Next
Dim addedInformation As Boolean
addedInformation = False
Set newdbinfo = DBEngine.Workspaces(0).OpenDatabase(DirPathInfoDbase)
Set newtblinfo1 = newdbinfo.OpenRecordset("AgentLiquid")
Set newtblinfo2 = newdbinfo.OpenRecordset("AgentDry")
'Check to make sure that liquid agent and code both exist, if one exists
If (DataCombo2.Text = "") And (DataCombo3.Text <> "") Then
  MsgBox "You must select a liquid agent.", vbExclamation, "WiscPlow"
  Exit Sub
End If
If (DataCombo3.Text = "") And (DataCombo2.Text <> "") Then
  MsgBox "You must select a code for the liquid agent.", vbExclamation, "WiscPlow"
  Exit Sub
End If
'If dry agent has been selected, add it to database
If DataCombo1.Text <> "" Then
    With newtblinfo2
        .AddNew
        .Fields("DryAgent") = DataCombo1.Text
```

Figure 41.
Example Internal Source Code Documentation for “OK” Button
(Comments are in Bold Type)

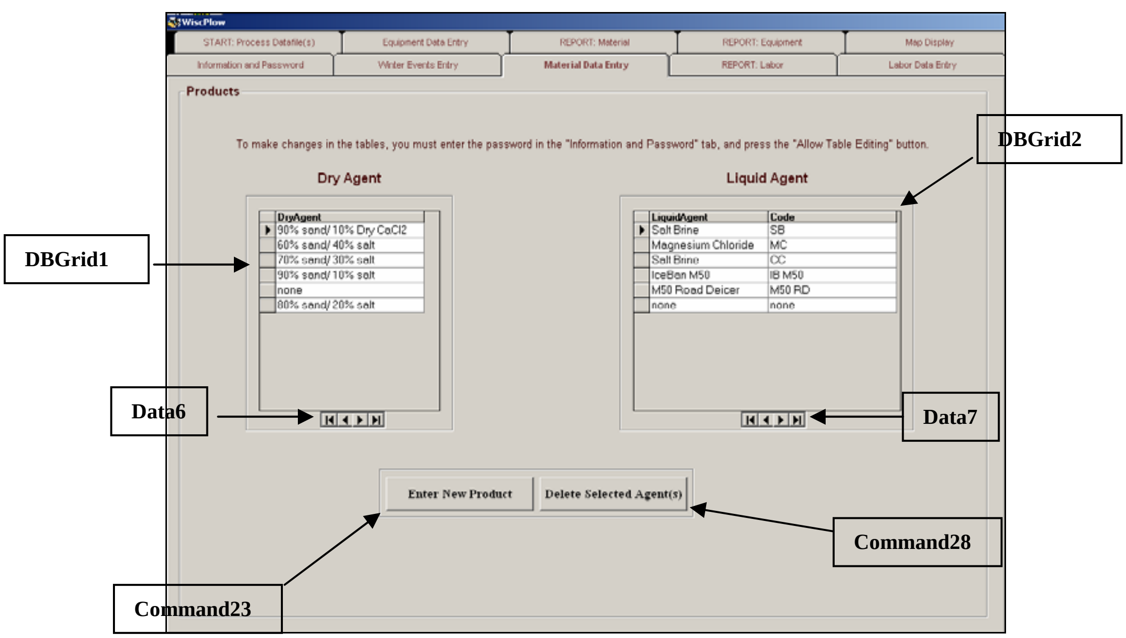


Figure 42.
Source Code Related to Material Data Entry Tab

Command23: Enter New Product

```
'Ok button that takes into effect changes to the dry and liquid agent, Products tab
Private Sub Command23_Click()
'Check the Password by calling function
If CheckPassword = False Then
   Exit Sub
End If
'Bring up material entry form
Form5.Show
'Refresh Grids after done entering records
DBGrid1.Refresh
DBGrid2.Refresh
End Sub
```

Command28: Delete Selected Agent(s):

```
'Delete selected agent in Products tab
Private Sub Command28_Click()
'Check the Password by calling function
If CheckPassword = False Then
   Exit Sub
End If
Dim response As Integer
Dim count As Integer
count = DBGrid1.SelBookmarks.count + DBGrid2.SelBookmarks.count
'Show message if no record is selected
If count = 0 Then
   response = MsgBox("You must first select a record to delete.", vbExclamation, "WiscPlow")
   Exit Sub
End If
'Ask the user if they are sure they want to delete record(s)
If count > 0 Then
   response = MsgBox("Are you sure you would like to delete the selected record(s)?", vbExclamation +
vbYesNo, "WiscPlow")
End If
'If they answer no, then exit procedure
If response = 7 Then
   Exit Sub
End If

'If they answer yes, delete records and display message that record(s) were deleted
If response = 6 Then

   'If none value is not selected, delete the selected value
   If Data7.Recordset.Fields("LiquidAgent").Value <> "None" And
Data7.Recordset.Fields("LiquidAgent").Value <> "none" And Data6.Recordset.Fields("DryAgent").Value <>
"None" And Data6.Recordset.Fields("DryAgent").Value <> "none" Then
      'Delete record(s) from DBGrid1
        If DBGrid1.SelBookmarks.count <> 0 Then
          For n = 0 To DBGrid1.SelBookmarks.count - 1
             Data6.Recordset.Bookmark = DBGrid1.SelBookmarks(n)
             Data6.Recordset.Delete
          Next n
```

```
            DBGrid1.Refresh
            Data6.Refresh
        End If
        'Delete record(s) from DBGrid2
        If DBGrid2.SelBookmarks.count <> 0 Then
            For n = 0 To DBGrid2.SelBookmarks.count - 1
                Data7.Recordset.Bookmark = DBGrid2.SelBookmarks(n)
                Data7.Recordset.Delete
            Next n
            DBGrid2.Refresh
            Data7.Refresh
        End If
        'Display message that record(s) were deleted
        MsgBox "The selected record(s) have been deleted", vbInformation, "WiscPlow"
    End If

    'Message to notify none value in Dryagent and Liquid agent is default value
    If Data7.Recordset.Fields("LiquidAgent").Value = "None" Or
Data7.Recordset.Fields("LiquidAgent").Value = "none" Or Data6.Recordset.Fields("DryAgent").Value = "none" Or
Data6.Recordset.Fields("DryAgent").Value = "None" Then

            MsgBox "None is default value", vbExclamation, "Wiscplow"
            Data7.Refresh
            Data6.Refresh

    End If

End If
End Sub
```

Figure 42 (continued).
Source Code Related to Material Data Entry Tab (Comments in Bold Type Face)

3. Word® (for access to the users' manual).

7.1.2. External Technical Documentation

Wiscplow's external technical documentation consists of:

1. A procedure map, a graphical representation of the procedure map, and the Wiscplow folder structure. These appear in Appendix A of this report.

2. A data dictionary that describes each item in each table in each Wiscplow database. The data dictionary runs to 42 pages and is included on a CD along with the master copy of this report.

7.2. User Documentation

The users' manual for Wiscplow runs to 109 pages and is not included in the text of this report. It exists as a Microsoft Word® document, embedded within Wisplow and accessible through the system's "Help" button. The users' manual is also included on a CD that accompanies the master copy of this report.

The table of contents for the Wiscplow users' manual appears in Figure 43. A full Wiscplow installation includes a tutorial with associated vehicle data files, databases, and network and patrol section shapefiles. A walk-through of the tutorial is included in Appendix A of the users' manual.

Wiscplow Users' Manual Table of Contents

Figure 43.
Table of Contents for Wiscplow Users' Manual

8. Installation and Training at County Highway Departments

The Wiscplow system, including user documentation, tutorials, test data, and roadway centerline network shapefiles, were installed at the Columbia and Portage County Highway Department Offices. County Information Technology staff were on-hand for the installation, as were County Highway Department personnel. Training was provided at each installation site. Wiscplow is now in the hands of local-level analysts for use in business operations. The system was also installed at WisDOT Headquarters, in the Bureau of Highway Operations Office.

Due to hardware problems, Kenosha County has never collected any vehicle data files. Moreover, near the conclusion of this project, all data collection equipment had been removed from Kenosha's concept vehicle and there were no immediate prospects for having it re-installed. Consequently, it was decided to postpone installation of Wiscplow at Kenosha County until a later date. The network shapefile, developed by the Research Team, was delivered to the Kenosha County Land Information Office.

The installation CD for Wiscplow includes a wizard that guides installers through the process. That CD is included along with the master copy of this report.

9. Options for Maintenance of Wiscplow and Its Data

There are a number of options for future maintenance of Wiscplow and the required databases. There are technical requirements and institutional needs. Full-scale implementation of Wiscplow across participating counties is expected to be incremental as more instrumented vehicles become available over time.

9.1. Data Requirements and Options

9.1.1. Roadway Spatial Databases

Wiscplow requires roadway network shapefiles with source scales of 1:24,000 or larger. These shapefiles must include patrol sections. They must also be attributed with roadway functional class. Such spatial databases were developed for Columbia, Kenosha, and Portage counties, but to remain useful, they must be maintained over time to keep them current. For example, modifications will be necessary whenever alignments change, new roadways are built, or patrol section boundaries are changed. "Custodianship" of these databases is a difficult issue. Options include:

6. WisDOT's local roads databases are maintained at headquarters in Madison. The unit that does this work clearly has personnel with appropriate skills and technology for maintaining Wiscplow spatial databases. However, such activities are not within their mission.

7. WisDOT's district offices use GIS as a matter of routine and could develop the capability to maintain Wiscplow's spatial databases. However, WisDOT district offices typically do not maintain or manage data for county highway departments.

8. It is technically feasible for UW-Madison to maintain the necessary spatial databases. However, the university is not in the practice of doing such things and some mechanism would have to be developed for continuity beyond the tenure of

current Research Team members. Some level of continued funding would have to be developed.

9. Consultants could be contracted to maintain the spatial databases. Such an arrangement would, perhaps, be more institutionally stable than having UW-Madison do the maintenance, but a continuing funding source would have to be identified.

10. The most attractive option is to have either the county highway departments or the county land information offices take custodianship of the spatial databases. This keeps both use and maintenance at the local level. Those who need the data become responsible for it. Impediments include lack of skills in spatial database maintenance at county highway departments and lack of resources at county land information offices.

9.1.2. Vehicle Data

Management of large volumes of vehicle data over time might become problematic. Current file naming conventions include acquisition date and vehicle identifier. This convention should be adequate for identification of individual files. However, as the number of instrumented vehicles grows and the stream of in-coming data increases, the shear volume of information might become unmanageable without careful procedures. It is recommended that vehicle data be archived on a seasonal basis so that data at hand for Wiscplow operations is for the current season only. If there is need to analyze data for multiple seasons, for example, to compare performance measures from year to year, the necessary data can be removed from its archive on an as-needed basis.

9.2. Institutional Options for Maintenance of Wiscplow Source Code

On-going maintenance of Wiscplow's source code, beyond Year 5, is problematic. Options include:

6. WisDOT's Bureau of Automation Services has the skills necessary for maintenance of Wiscplow. However, BAS has a long-standing policy of not maintaining software developed by other parties.

7. The source code could be maintained at the county level, by land information offices or IT units. However, they would have to agree to do this. Furthermore, each installation of Wiscplow would evolve differently over time, as each would be maintained separately. Eventually, data models and, therefore, databases would become incompatible across installations.

8. UW-Madison could potentially maintain the source code. Some mechanism would have to be developed for continuity beyond the tenure of current Research Team members. Some level of continued funding would have to be developed.

9. Consultants could be contracted to maintain the source code. A continuing funding source would have to be identified.

10. The source code could be maintained, on a retainer basis, by a member or members of the Research Team. A continuing, low-level funding source would have to be identified.

10. Summary and Conclusions

A GIS-based decision-support system for winter highway maintenance has been developed, tested, documented, and installed at two county highway department offices and WisDOT headquarters. Wiscplow computes performance measures and produces reports and decision-management tools based upon them. Testing included both computational validation and integrity of the user interface.

Wiscplow requires accurate network shapefiles of roadway centerlines and patrol sections. These were developed, along with associated metadata, tested, and delivered to three Wisconsin counties. Control procedures were used to ensure quality in the network shapefiles.

The “map-matching” problem, arising from errors in DGPS vehicle coordinates and in roadway centerline representations, was addressed through development, testing, and implementation of a decision-rule algorithm. Wiscplow now associates nearly all vehicle data points with correct roadway centerlines.

Detailed internal and external technical documentation was developed for Wiscplow. Such documentation is necessary for on-going maintenance of Wiscplow beyond the tenure of the Research Team.

Detailed user documentation, including tutorials and test data, was developed and delivered to participating parties. Such documentation is necessary to ensure Wiscplow’s utility in the workplace. Training was provided to county highway department personnel at the two installation sites.

Wiscplow was developed as a single-user “standalone” system because its use at the local level was envisioned this way. Wiscplow is not an enterprise solution. Under its current configuration, data from more than one county cannot be combined.

Five options for maintenance of roadway network shapefiles were described. The most attractive of these options is to have county highway departments or land information offices take custodianship of the data. However, there are impediments, including lack of resources and inexperience with GIS (and, therefore, lack of track record with land information offices) on the part of many county highway departments.

Institutional arrangements concerning maintenance of Wiscplow source code might be the most difficult problem to address while ensuring longevity of the products of this research. Five options were described. Perhaps the most feasible is a low-level retainer for a member or members of the Research Team to take custodianship of the source code.

This report describes activities and accomplishments during the concluding fifth year of a significant research and development relationship among WisDOT, a number of county highway departments and land information offices, and the Midwest Regional University Transportation Center and the Department of Civil and Environmental Engineering at the University of Wisconsin – Madison. It seems valid to assert that all parties have benefited. WisDOT and the counties received advanced technology and training in analysis and management of winter maintenance concept vehicle data. They participated in development of an understanding of these data and in development of tools for exploitation of the data in decision making. The University of Wisconsin – Madison saw this research form the basis for seven M.S. degrees and two PhD degrees. The research produced six published papers and nine formal presentations at national and statewide conferences and workshops.

List of References

Blazquez, C., (2005), "Decision-Rule Algorithm for Map Matching in Transportation", PhD Dissertation, University of Wisconsin – Madison.

Blazquez, C. and A. Vonderohe, (2005), "Simple Map-Matching Algorithm Applied to Intelligent Winter Maintenance Vehicle Data", Transportation Research Record, Journal of the Transportation Research Board, No. 1935, pages 68-76.

Vonderohe, A., Malhotra, A., Sheth, V., Mezera, D., and T. Adams, (2000), Wisconsin Winter Maintenance Concept Vehicle: Data Management Year 1, Final Project Report, Submitted to the Wisconsin Department of Transportation.

Vonderohe, A., Adams, T., Malhotra, A., Sheth, V., Manchikanti, K., and D. Mezera, (2001), Wisconsin Winter Maintenance Concept Vehicle: Data Management Year 2, Final Project Report, Submitted to the Wisconsin Department of Transportation.

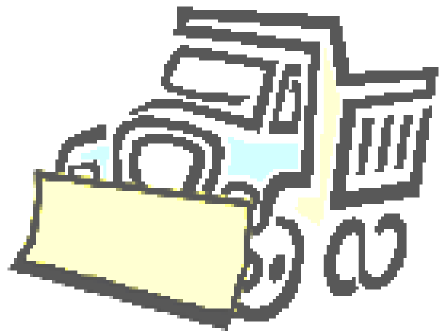
Vonderohe, A., Adams, T., Malhotra, A., Stanuch, G., and C. Blazquez, (2002), Wisconsin Winter Maintenance Concept Vehicle: Data Management Year 3, Final Project Report, Submitted to the Wisconsin Department of Transportation.

Vonderohe, A., Adams, T., Blazquez, C., and J. Maloney, (2003), Wisconsin Winter Maintenance Concept Vehicle: Data Management Year 4, Final Project Report, Submitted to the Wisconsin Department of Transportation.

Appendix A.
Wiscplow External Technical Documentation

WiscPlow Version 3.0

(Technical Documentation)



Developed at

The Department of Civil and Environmental Engineering

University of Wisconsin–Madison

Funded by the

Wisconsin Department of Transportation

&

Midwest Regional University Transportation Center

December, 2005

<Table of Contents>

1. PROCEDURE MAP

PURPOSE: This document is intended to aid in the understanding of WiscPlow's source code. It attempts to show the hierarchical structure of WiscPlow's procedures. Each section, beginning with a bold header, corresponds to a button pressed by the user in the WiscPlow interface. What follows is the hierarchical progression of procedures within WiscPlow that are called when that button is pushed. Only the most complicated procedures, which involve calls to other subsequent procedures, are mapped out in this document. Other commands and procedures in WiscPlow that are not discussed here can be easily understood by anyone with programming experience in Visual Basic, and therefore are not required in this document.

Loading the Main Form: The Form_Load() procedure runs automatically every time WiscPlow's main form is loaded.

Form_Load() – For the Main Form – This procedure reads in all the data required on the main form. It reads in global variables from the current.ini file in the pathways subfolder of WiscPlow. It also creates and configures the map objects used to create the Map Display in WiscPlow. It populates the Storm, Incident, and Antiicing tables in the current database and a few tables in the info database by making calls to the following procedures:

1. CombineAllEvents() – Populates the AllWinterEvents table in the current database using Storm, Incident, and Antiicing tables.
2. CombineAllIndexes() – Populates the AllEventIndexes table in the info database using StormIndex, AntiIcingIndex, and IncidentIndex tables.

START: Process Datafile(s) Tab: Most of WiscPlow's most complicated code is contained within procedures called from the user interface in the "START: Process Datafile(s)" tab, contained in the main form. Each bold header below contains the name of the initial procedure called in Visual Basic, as well as the caption contained on the button in the WiscPlow user interface.

AddNewFileButton_Click() – "Add New File To Process List" Button – This procedure runs when the user clicks the "Add New file to Process List" button. The procedure clears all the variables used to read raw datafiles before calling the following procedure.

1. Readfilename() – Goes through many checks of the data, then decodes the new datafile. It checks if a database already exists for the specified file and also checks to make sure that the file contains information meaningful to WiscPlow. It contains calls to the following procedures:
    a. DecodeMessages() – This procedure is called if all of the above checks of the data are passed. It extracts information from the raw datafile and stores it in arrays.
    b. bFileExists(sFile As String) – This procedure is called to determine if a file exists. It is used a few times throughout the Readfilename() procedure.

Command1_Click() – "Apply Details" Button – After the user has entered information in the appropriate combo boxes, they push the "Apply Details" button and this procedure runs. It calls the following procedure to make sure that all the information added by the user is valid.

1. verifyInformation() – This procedure checks to make sure that all the appropriate data has been entered by the user, and then checks to make sure that all the information necessary to process the datafile exists in the databases. For example, it checks to see that there is a labor rate for the specified operator and checks to make sure that costs exist in the database for all attachments on the specified truck.

Command11_Click() – "Remove file from database" Button – This procedure runs when you remove a file (selected from a drop-down menu by the user) from the database. It searches through the current database and deletes all records associated with the specified datafile. It also deletes the database for that specific file from the datafile_databases subfolder of WiscPlow. Lastly, it deletes the shapefile used in the map display for the specified file from the datafile_shapefiles subfolder of WiscPlow.

Command8_Click() – "Remove file from process list" Button – Removes the specified file (selected by the user from a drop-down menu) from the Process List by removing its record from the Batch Process table in the current database.

Command12_Click() – "Process File(s) in List" Button – When the user is ready to process datafiles, they press the "Process File(s) in List" button, and this procedure runs. Everything that follows is looped through for every datafile to be processed. Files to be processed are listed in BatchProcess table of current database.

2. erasemessage() – erases all arrays containing decoded messages (i.e. Message14)
3. decodemessagefor12() – Decodes raw data and stores the results in arrays.
4. BuildDatabase() – First, this procedure calls CreateDatabase() to create a database for each datafile and appends tables to the database. Then it calls PopulateDatabase() to populate the tables using values stored in arrays.
    a. CreateDatabase() – Calls all of the following procedures to create specified tables, then appends all of the tables to one database.
        i. CreateOperator
        ii. CreateOprtngDay
        iii. CreateTnkVol
        iv. CreateBrdCrmbs
        v. CreateGPSStatus
        vi. CreateApplZone
        vii. CreateBlstCal
        viii. CreateVehicle
        ix. CreateRectnTime
        x. CreatePrwtMtrCal
        xi. CreateSpdCal
        xii. CreateSndCal
        xiii. CreateSltCal
        xiv. CreateAntIceCal
        xv. CreatePrd1Cal

xvi. CreatePrd2Cal
xvii. CreateLnWdthCal
xviii. CreateSndRt
xix. CreatePrwtSndRt
xx. CreateSltRt
xxi. CreatePrwtSltRt
xxii. CreatePrd1Rt
xxiii. CreatePrwtPrd1Rt
xxiv. CreatePrd2Rt
xxv. CreatePrwtPrd2Rt
xxvi. CreateAntIceRt
xxvii. CreateSpnrRPM
xxviii. CreateTmprtr
xxix. CreateSpeed
xxx. CreateAlarm
xxxi. CreateWyPntID
xxxii. CreateAttFrontPlow
xxxiii. CreateDmpBdy
xxxiv. CreateAttLeftWing
xxxv. CreateAttRightWing
xxxvi. CreateAttScraper
xxxvii. CreateSprdr
xxxviii. CreateAttSpreader
xxxix. CreateAttSprayBar
xl. CreateAttTruck
xli. CreateBrdCrmbsData
xlii. CreateEquipUsage

b. PopulateDatabase() – Calls all of the following procedures to populate the tables in each datafile's database using the arrays containing the decoded messages from decodemessagesfor12().
i. FillBrdCrmbs
ii. FillOperator
iii. FillOprtngDay
iv. FillGPSStatus
v. FillTnkVol – Commented Out Right Now
vi. FillApplZone
vii. FillBlstCal
viii. FillVehicle
ix. FillRectnTime
x. FillPrwtMtrCal
xi. FillSpdCal
xii. FillSndCal
xiii. FillSltCal
xiv. FillAttFrontPlow
xv. FillDmpBdy
xvi. FillAttLeftWing

xvii. FillTmprtr
xviii. FillSprdr
xix. FillPrd1Cal
xx. FillPrd2Cal
xxi. FillWyPntID
xxii. FillAlarm
xxiii. FillSpeed
xxiv. FillLnWdthCal
xxv. FillAntIceCal
xxvi. FillSndRt
xxvii. FillSltRt
xxviii. FillPrwtSndRt
xxix. FillPrwtSltRt
xxx. FillPrd1Rt
xxxi. FillPrd2Rt
xxxii. FillPrwtPrd1Rt
xxxiii. FillPrwtPrd2Rt
xxxiv. FillAntIceRt
xxxv. FillSpnrRPM
xxxvi. FillAttScraper
xxxvii. FillEquipUsage

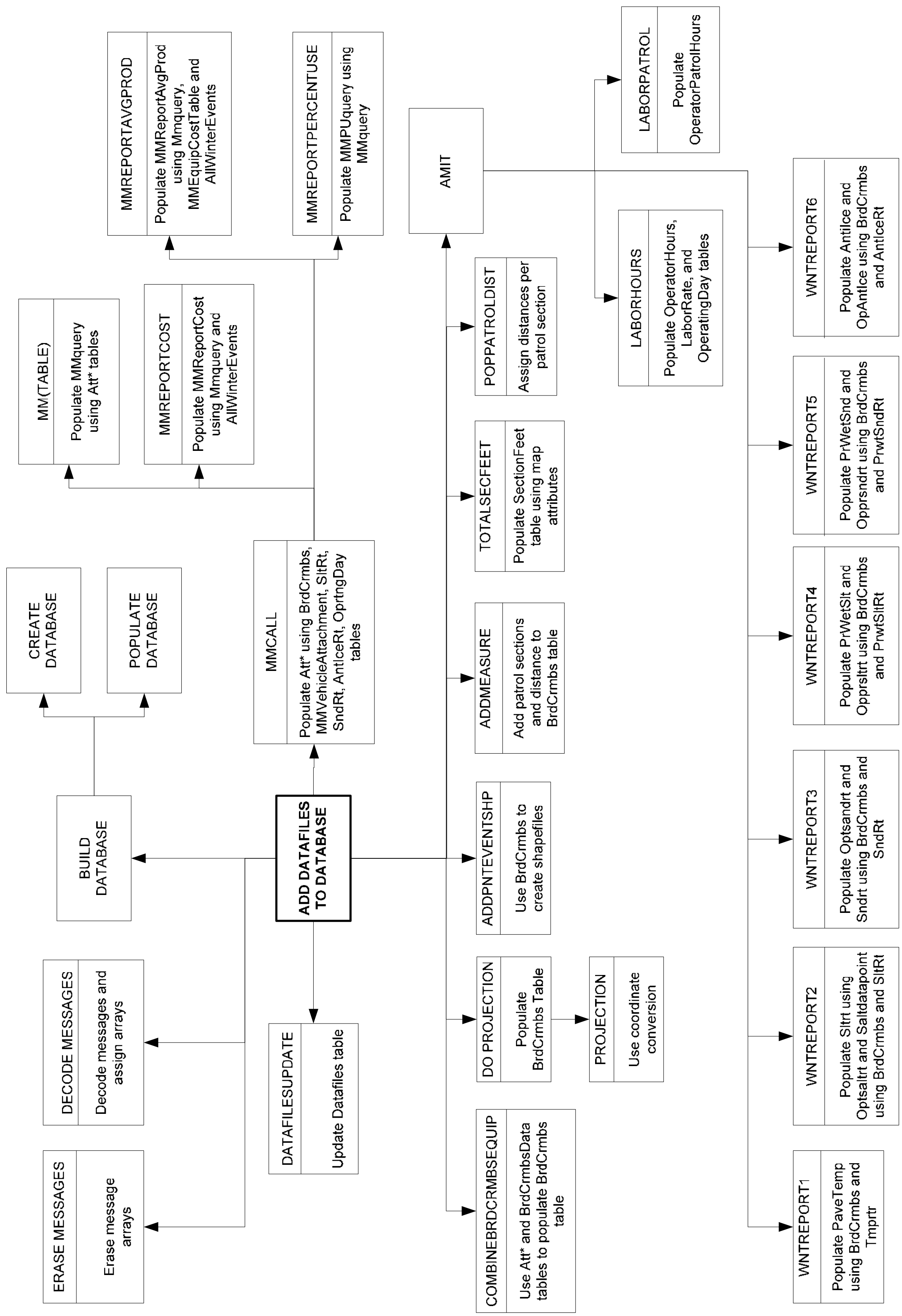
5. CombineBrdcrmbEquip() – Combines data from BrdCrmbsData table and EquipUsage table to create BrdCrmbs table in each datafile database.
6. DoProjection() – Converts GPS coordinates to county coordinates and then populates the X and Y columns in BrdCrmbs table in database. It calls the following procedure in order to do so.
    a. projection() – This procedure actually converts the GPS coordinates into the appropriate county map coordinates using the projection parameters stored in the CoordinateConversion table of the Info database.
7. addpnteventshp() – Adds the breadcrumbs from BrdCrmbs table into a new map layer.
8. totalsecfeet() – Populates the SectionFeet table in current database, showing total distance of each patrol section in county.
9. poppatroldist() – Populates the PatrolDistance table in current database, showing distances traversed in each patrol section for processed data.
10. addmeasure() – This procedure is the map-matching algorithm used by WiscPlow to determine what road a given GPS point is on. It contains calls to the following procedures, and sometimes these procedures call one another, to perform the map-matching algorithm. Ultimately, this procedure determines what road the given GPS points are on, and uses that data to populate the patrol section, roadway class, and distance fields in BrdCrmbs table.
    a. try1tolast
    b. correctWrongSnap
    c. passingArrays
    d. getpath
    e. tryaltprev

f. tryaltnext
g. try2to4
h. try2to5
i. try1to3
j. speedonepoint
k. speedzero
l. GetOther
m. CheckSpeed
n. GetMinDist
o. QueryID
p. PercentAlong
q. PatrolDistance
r. FindPath
s. OpenShapeFile

11. Amit() – This procedure populates tables in the current database using data from the datafile database, which was created earlier. The tables in the current database are used directly by WiscPlow to compute performance measures. The following procedures are called, and they populate different tables in the current database using information extracted from the datafile's database.
    a. Laborhours() – Calculates number of hours an operator worked and populates OperatorHours table of current database.
    b. Laborpatrol() – Calculates normal and overtime hours worked and populates OperatorPatrolHours table in current database.
    c. wntreport1() – Populates PaveTemp table in current database.
    d. wntreport2() – Populates tables related to salt rate in current database.
    e. wntreport3() – Populates tables related to sand rate in current database.
    f. wntreport4() – Populates tables related to pre-wet salt in current database.
    g. wntreport5() – Populates tables related to pre-wet sand in current database.
    h. wntreport6() - Populates tables related to anti-ice in current database.
12. MMcall() – Populates tables in current database related to machinery management.
    a. MM(table) – This procedure is called several times while populating the table called MMQuery. Each time it is called, it takes one of many tables beginning with Att (e.g. AttSpreader) as its argument.
    b. MMReportCost() – Populates MMReportCost table using MMQuery and AllWinterEvents tables.
    c. MMReportAvgProd() – Populates the MMReportAvgProd table using the MMQuery table.
    d. MMReportPercentUse() – Populates the MMPUQuery table using MMQuery.
        i. Pucontinue() – Populates MMPercentUse & MMPercentUseState using the other MM tables.

13. DatafilesUpdate() – Updates the Datafiles table in the current database to include the datafile just processed and the information associated with it.

2. GRAPHICAL REPRESENTATION OF PROCEDURE MAP



3. WISCPLOW FOLDER STRUCTURE

PURPOSE: This chart shows the general structure of WiscPlow Version 3, beginning with the main WiscPlow folder and moving down into the contents of the folders and files within.