

UPDATING AND EXPANDING A MULTI RESOLUTION  
LINEAR REFERENCING SYSTEM  
FUNCTIONAL MERGE

By

ANTHONY LANE MORRISON

DR. ANDREW J. GRAETTINGER, COMMITTEE CHAIR

DR. STEVEN L. JONES

DR. RANDY K. SMITH

A THESIS

Submitted in partial fulfillment of the requirements  
for the degree of Master of Science  
in the Department of Civil, Construction and Environmental Engineering  
in the Graduate School of  
The University of Alabama

TUSCALOOSA, ALABAMA

2012

Copyright Anthony Lane Morrison 2012

ALL RIGHTS RESERVED

## ABSTRACT

The Wisconsin Department of Transportation (WisDOT) uses two separate linear referencing systems (LRSs) for complete statewide crash mapping. The State Trunk Network (STN) represents only state routes while the Wisconsin Information System for Local Roads (WISLR) includes all roads, with additional detail given to local routes. A functional link between the two systems, the link-link table, has been developed that allows data to be translated from STN (a higher resolution representation) to WISLR (a lower resolution representation). This functional link between the two systems requires updating to reflect changes due to roadway construction, linear referencing system edits, and other changes. The focus of this research was to develop a technique to update the functional merge between the two systems.

Updating the link-link table is needed to allow for continued data translation between from STN to WISLR, allowing for regular formation of a statewide crash map. While data is easily translated from high to low resolution, ambiguities arise when data is moved from low resolution to high resolution. Research presented in this thesis identifies common problems associated with low to high resolution data translation and provides some rules and guidelines to accommodate these issues.

This thesis presents the link-link table update methodology, an approach to expanded data translation between dissimilar-resolution LRSs, and recommendations for future work related to each topic.

## DEDICATION

This thesis is dedicated to my friends, family, and coworkers who helped me in various capacities to assist me with this research and this ensuing manuscript.

## ACKNOWLEDGMENTS

I am thankful to so many for the opportunity to work on this project. God has continuously blessed me with opportunities like these throughout my life and I am eternally grateful. I am also thankful for my friends, family, and colleagues for their help and encouragement throughout this research and the culminating creation of this thesis.

I am especially grateful to Dr. Andrew Graettinger for helping to guide me in various areas of my life both within and outside of academics. I would also like to thank Dr. Steven Parker at the University of Wisconsin-Madison for his help throughout this project. This research would not have been possible without the help and support of my friends and coworkers. I'd like to thank Zach Ryals, Blake Doherty, Michael Herron, Rachel Cary, Luke Taylor, Stephanie Farrell, Wil Stone, Anika Kuczynski, Michael Sherer, and Ashley Purkey for helping me in various ways with this research and for providing a great office environment.

Finally I thank The University of Alabama and the State of Alabama for providing me this opportunity. I hope this work, along with countless others' work that is far greater than my own, may continue to illustrate the incalculable value of public education.

## CONTENTS

ABSTRACT .....	ii
DEDICATION .....	iii
ACKNOWLEDGMENTS .....	iv
CONTENTS .....	v
LIST OF TABLES .....	
LIST OF FIGURES .....	i
INTRODUCTION .....	1
1.1 Introduction .....	1
1.2 Thesis Organization .....	4
LITERATURE REVIEW AND BACKGROUND .....	6
2.1 Conceptual Linear Referencing Systems .....	6
2.2 WisDOT Linear Referencing Systems .....	9
METHODOLOGY .....	14
3.1 Introduction .....	14
3.2 The STN and WISLR Systems .....	15
3.3 The Basic Link-Link Coding Process .....	17

3.4 Incorporating date information into the Link-Link Table.....	20
3.4.1 Record_Created Column.....	20
3.4.2 Record_Historic Column .....	21
3.4.3 Start_Valid Column .....	21
3.4.4 End_Valid Column .....	22
3.5 Populating Updated Information into the Link-Link Table.....	23
3.5.1 Simultaneous Date Column Population .....	23
3.5.2 Obtaining “Current” Data for Link-Link Updating .....	25
3.5.3 Identifying Changes and Updates to STN and WISLR .....	26
3.5.4 Updating Records in the Link-Link Table .....	28
3.6 Quality Assurance/Quality Control (QA/QC) .....	33
3.6.1 STN Link Check .....	33
3.6.2 WISLR Link Check .....	34
3.6.3 Gore Point Check.....	34
3.6.4 XY Connector Lines Check .....	35
3.6.5 Modified WISLR Point Moving Program .....	39
3.7 Update Issues .....	39
3.8 Using Link-Link for Data Translation .....	40
3.8.1 Approach.....	40
3.8.2 Data Translation.....	40
3.8.3 Problem 1: Intersection/Median Crossover Ambiguous Point Translation .....	41

3.8.4 Problem 2: Turn Lane Ambiguous Point Translation .....	42
3.8.5 Problem 3: Wayside Ambiguous Point Translation.....	44
3.8.6 Proposed Data Translation Rules .....	46
3.9 WISLR Editing Link Identification .....	49
3.10 Conclusion .....	51
RESULTS .....	53
4.1 Introduction.....	53
4.2 Dane County Pilot Study for Link-Link Update.....	53
4.3 Statewide Link-Link Coding Update Results .....	55
4.4 Data Translation Results .....	55
4.5 WISLR Editing Scope Identification .....	58
4.6 Conclusion .....	58
CONCLUSION AND FUTURE WORK .....	59
5.1 Conclusion .....	59
5.2 Future Work .....	60
REFERENCES .....	61
APPENDIX A - LINK-LINK DATE COLUMN POPULATION QUERIES .....	63
APPENDIX B - UPDATED WISLR POINT MOVING PROGRAM.....	68



## LIST OF TABLES

Table 2.1. Names and Descriptions for the Six Main Link-Link Columns and Three Relevant Link-Link Flag Columns (Ryals, 2011).....	21
---	----

## LIST OF FIGURES

Figure 2.1. Generalized LRS model known as NCHRP 20-27 model (Vonderohe & Hepworth, 1998) .....	7
Figure 2.2. Visualization of 20-27 (Wisconsin Department of Transportation, 1998) conceptual model (Scarponcini, 2002) .....	19
Figure 3.1. Example of the STN/WISLR relationship from the Madison, Wisconsin area. STN is represented by thick red lines, while WISLR is represented with thin black lines .....	26
Figure 3.2. Example of the basic link-link table coding process: (a) illustrates a basic roadway representation in WISLR and STN, (b) and (c) show basic attribute information from the shown links, and (d) demonstrates the basic link-link table associated with the shown representations	19
Figure 3.3. Example of county-specific data selection procedure: (a) and (b) select Dane County as the relevant county for updating, (c) the “Select by Attributes” user interface window, (d) Dane County STN links selected using “Select by Attributes” .....	29
Figure 3.4. Example of data population procedure associated with link-link table update due to a change in STN links: (a) example roadway section with a new and deleted STN link and unchanged WISLR link, (b) attribute information for the STN links in the roadway section, (c) attribute information for the WISLR link in the roadway section, (d) example link-link table entry for old and new STN link in the roadway section .....	31
Figure 3.5. Command line used for “XY_Lines” generation .....	38
Figure 3.6. Example of “XY” lines shape file. Lines connect related data points on STN (line on right side) and WISLR (line on left side).....	38

Figure 3.7. General example of ambiguous point placement due to incongruent intersection/median crossover representations: (a) example of data translation from STN to WISLR; (b) example of data translation from WISLR to STN.....	42
Figure 3.8. General example of ambiguous point placement due to incongruent turn lane representations: (a) example of data translation from STN to WISLR; (b) example of data translation from WISLR to STN.....	54
Figure 3.9. General example of ambiguous point placement due to a wayside: (a) example of data translation from STN to WISLR; (b) example of data translation from WISLR to STN. ....	45
Figure 4.1. Thematic map of the number of data changes associated with each county.....	54
Figure A.1. Start_Valid_Update1 query language.....	74
Figure A.2. Start_Valid_Update2 query language.....	74
Figure A.3. End_Valid_Update1 query language.....	75
Figure A.4. End_Valid_Update2 query language.....	75
Figure A.5. End_Valid_Update3 query language.....	76
Figure A.6. End_Valid_Update4 query language.....	77

## CHAPTER 1

### INTRODUCTION

#### 1.1 Introduction

The Wisconsin Department of Transportation (WisDOT) developed and maintains two separate and independent linear referencing systems (LRSs) for use with traffic and transportation business data in the State. The State Trunk Network (STN) was developed in the early 1990s for use on Wisconsin interstates and state roads. Within STN there exists a network of links and nodes that represent travel paths between state route intersections. The links are described with unique identifiers and as-driven distances. The Wisconsin Information System for Local Roads (WISLR) was developed approximately 10 years after STN for use on local roads throughout the State. WISLR originated from cartographic representations that were digitized from existing local, state, and federal sources. The digitized lines were converted into multi-directional links that were split at intersections. End points of links were used to create nodes. WISLR links are described by unique identifiers and although WISLR personnel collect linear distances for links, state route distances are not maintained because of existing length data in STN (A. J. Graettinger, Qin, Spear, Parker, & Forde, 2009). Although WISLR shares approximately 12,000 miles of roadway with STN, each system was developed and has progressed independent of each other in order to meet various business needs within WisDOT. However, due to differences in data types and formats, structural interoperability between the two systems is difficult.

A connection between STN and WISLR was created that relates a segment of a link in one system to the corresponding segment of link in the other system. This relationship is stored in the “link-link” table. A statewide link-link table was completed in 2011 and was employed to successfully move 2005-2009 state route crash data from STN to WISLR. Specifically, the link-link table has allowed crash data points described with STN links and offsets to be translated into crash data points described with WISLR links and offsets. Moreover, the link-link table allows for movement from STN to WISLR of any point that contains STN link and offset information. Sharing and translating crash data has been the main focus of this effort.

Although the link-link table was successful in providing a functional merge between STN and WISLR, the link-link table is not dynamic. Expressly, the original link-link table was created using data associated with 2009 conditions. Every year, STN and WISLR personnel make numerous changes to line work and various related attribute information to keep up with changing road networks. Updates are required of both systems any time there is construction, surrendering of former state roads to local organizations, and other common activities. As a result, the link-link table becomes, to an extent, outdated when updates are performed on WisDOT LRSs. This circumstance can cause problems if, for example, a crash is coded to an STN link that does not exist in the link-link table. The example crash would subsequently not move to any WISLR link.

Because of the need to keep pace with changes made to transportation networks throughout the State, updating the link-link table to reflect these changes is a priority. This research defines a method that allows for yearly updating of the link-link table to incorporate transportation network modifications into the functional merge between STN and WISLR. The

presented method allows for updates to be made to the table without disrupting current WisDOT work flow.

Another related area of research presented in this thesis relates to data translation from WISLR to STN, which is in the opposite direction of the original intent of the link-link table. To accommodate the creation of a statewide crash map consisting of crashes on both state and local roads, only data movement from STN (high resolution) to WISLR (low resolution) has been thus far refined to a functional operation level. Data translation from WISLR to STN – from low resolution to high resolution – is desired, in large part, because of current WisDOT crash data reporting procedures. These procedures are presented below.

WisDOT currently uses a statewide system of reference points to which to “code” crashes occurring on state routes. The Reference Point (RP) System allows designation of the linear location of features along a roadway (Wisconsin Department of Transportation, 1998). An RP can occur at a number of different physical landmarks including intersections, above ground bridges, railroad crossings, state boundaries, and other identifiable features. When the RP linear referencing method was originally developed in the 1970s, each RP was given a number and an offset or “plus distance” on the route on which the RP appeared. After the development of the STN link-offset linear referencing system, each RP number was assigned an STN link and offset. Current crash reporting procedures involve analyzing written crash record information created by law enforcement agents and assigning each crash that occurs on state routes an RP number and an offset. By using the RP and offset, WisDOT can assign each crash an STN link and offset.

Recently, WisDOT embarked on a project to develop and implement an Incident Location Tool (ILT) to assist law enforcement with crash location. This interactive tool will

implement a cartographic representation of the roadway network in patrol cars and will function by having the law enforcement officer place a point at the location of a crash. WISLR will be used as the roadway network in ILT. The tool will function in such a way that when an officer clicks a position on the map, the associated WISLR link and offset information will be captured and stored in the crash record. The link-link table will then be used to determine an STN link and offset for crashes occurring along state routes. WisDOT seeks to continue using STN link and offset information because of the numerous analysis tools available within the STN system that are not functional within WISLR. Unfortunately, inconsistencies between the STN and WISLR systems create problems when data is moved from WISLR to STN. Currently, WisDOT is planning to flag crashes that map from WISLR to multiple locations in STN, and manually use the existing RP coding method to assign an STN link and offset.

Because crash location data needs to be in three formats, STN link and offset, WISLR link and offset, and RP number and offset within the STN system, accurate data movement between the WISLR and STN systems is needed.

## 1.2 Thesis Organization

This thesis is organized into five chapters. Chapter 2, Literature Review and Background, presents not only a basic structure of a LRS, but also how updates to the general systems have been handled and how data translation between two systems has been previously approached. Chapter 3, Methodology, describes the link-link table update technique in detail and presents the technique for WISLR-to-STN data translation. Chapter 4, Results, presents the progress of the ongoing update project and the results of implementing a systematic data translation technique. Chapter 5, Conclusion and Future Work, provides closing remarks about the research presented and provides suggestions for valuable future work topics related to this

research. Appendices A and B provide additional detail relating to technical information created during this research.



## CHAPTER 2

### LITERATURE REVIEW AND BACKGROUND

A linear referencing system is a computer based model that often represents transportation data for use by Departments of Transportation. An LRS stores not only graphical data, but also associated attribute data such as link ID and length. This attribute data allows for the placement of transportation data using only a link ID and a measure value of a data element. This data placement method has become particularly valuable for various transportation business data such as roadway crashes. The basic concept of a linear referencing system and the tools associated with an LRS are utilized within the WisDOT systems, STN and WISLR.

#### 2.1 Conceptual Linear Referencing Systems

Linear Referencing Systems are made up of multiple levels of related information. The conceptual model in Figure 2.1 shows a LRS as a compilation of three main parts: a datum, network(s), and linear referencing methods (LRMs).

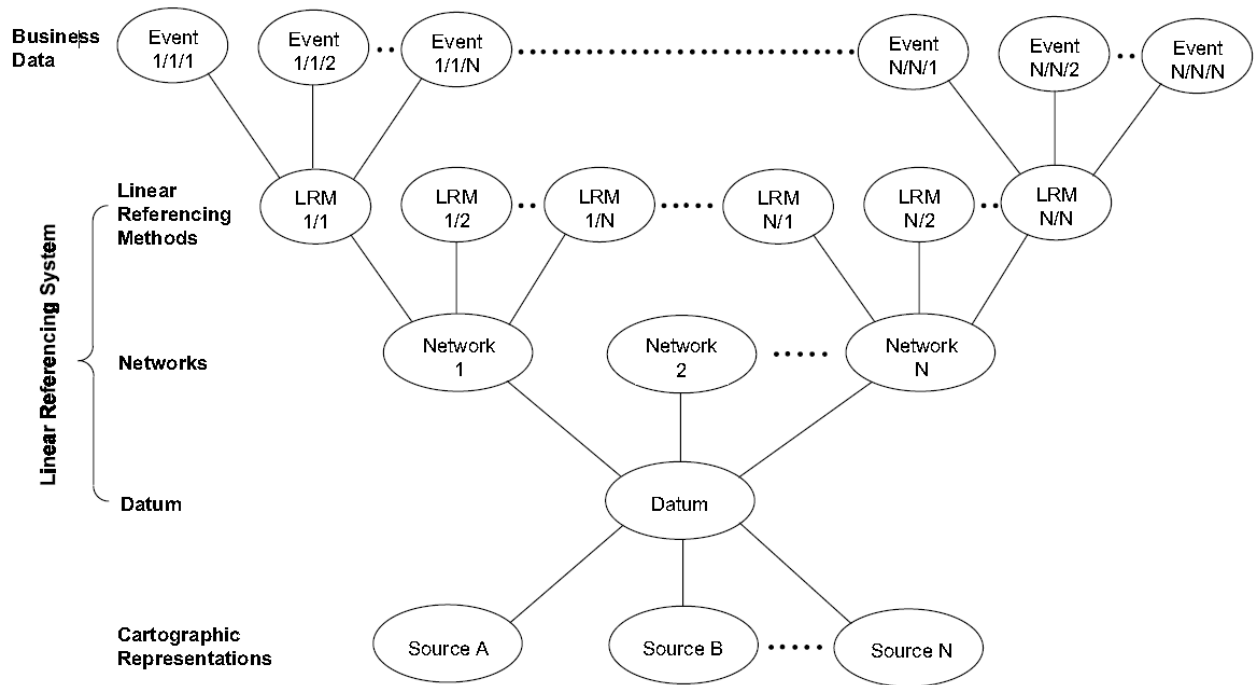


Figure 2.1. Generalized LRS model known as NCHRP 20-27 model (Vonderohe & Hepworth, 1998)

The datum, shown in the center of Figure 2.1, is an absolute set of anchor point and anchor sections. These anchors points and segments relate to real locations and act as a platform for movement among the other parts of the conceptual model. Anchor points require some detailed explanation of the location in the field, which can be quantitative and/or qualitative. Anchor sections are solely a connection between two anchor points. The length of the anchor section can be calculated in the field to provide an accurate relationship between the anchor points (Vonderohe, et al., 1997).

A network, as seen in Figure 2.1, is best described as a means for communication and movement among point locations (Miller & Shaw, 2001). There are different types of networks that can be present simultaneously through a common datum that is associated with an LRS, as shown by Network N in Figure 2.1. A common network type is a link-node system, where links

are directional and act as flow conduits, and nodes are locations where links meet. Vice-versa, nodes can be described as locations where flow can change, and the links simply connect certain nodes, as described in the WisDOT Location Control Management Manual (Wisconsin Department of Transportation, 1998).

A linear referencing method (LRM) is a way of describing the location of transportation data on a given network. While there are several common LRMs, the link-offset method is employed by WisDOT in the STN and WISLR systems. This method uses the directional link on which the transportation data is located, as well as the distance down link that must be traveled from the beginning of the link to the event.

Events are the visual product of processing business information through a LRS and are at the center of spatial analysis. In a link-offset LRM, event points will be represented by a link ID and an offset (Vonderohe & Hepworth, 1998). Bridge locations and segments of pavement are physical data events, while crash points and project reference lines are intangible data events. Events are generated solely through a LRS and will not always correspond to the actual location of an event in the field due to fact that an LRS is only an abstract representation of actual conditions.

Multiple maps and cartographic representations can be related to an LRS based on the virtual anchors of the linear datum, meaning that cartography is not necessary to the function of an LRS. However, cartography provides a visual perspective to better understand the relationships of the network(s) and event data (Ryals, 2011). An LRM processes events by referencing the Network, the Network is located on the earth's surface by the Datum, and the cartography is overlaid onto the Datum for visualization. A graphic illustration showing the levels of the conceptual LRS data model is shown in Figure 2.2.

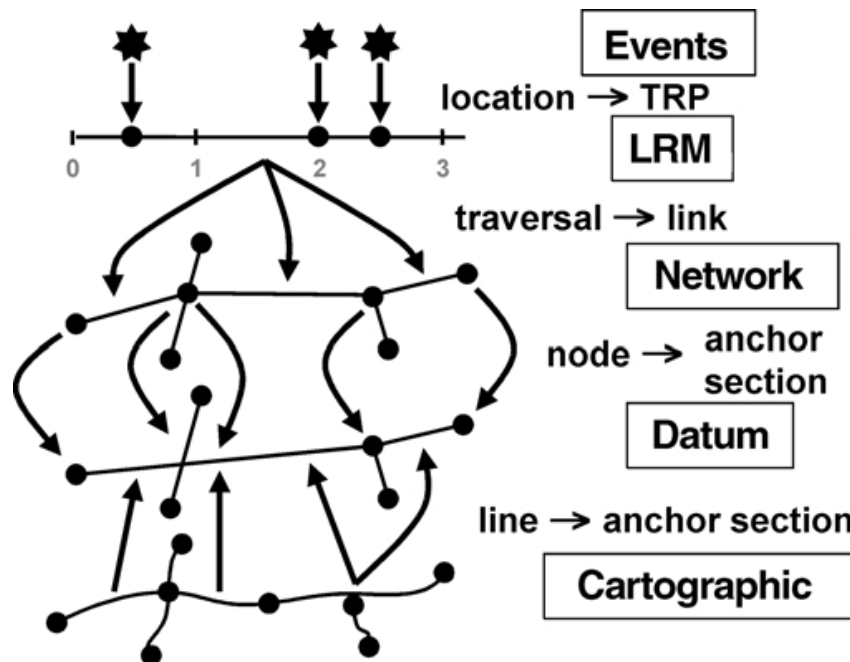


Figure 2.2. Visualization of 20-27 (Wisconsin Department of Transportation, 1998) conceptual model (Scarponcini, 2002)

Data maintenance is of utmost importance regarding LRSs. To continue functionality of an LRS it is important that all data, especially route and event data, within the systems are maintained properly (Curtin, et al., 2007).

Schema uniformity is of utmost importance regarding data translation between transportation systems. For data system interoperability to be possible, agreement must be created between the data models. The models must both identify transportation features with corresponding attributes and must be represented as line or point events using linear referencing (Dueker & Butler, 2000).

## 2.2 WisDOT Linear Referencing Systems

STN and WISLR follow the basic components of an LRS. Additionally, WisDOT business data reporting methods follow the requirements for events within an LRS. The

structure of WisDOT systems and data reporting methods allow for typical LRS rules to be applied to translate data between STN and WISLR.

WisDOT chose to keep STN and WISLR separate in almost all respects, even when devising a linear datum. STN was constructed utilizing a datum that ties features in the network to field data positions that translate the theoretical location of network features. WISLR, on the other hand, was built utilizing a datum that is directly tied to the spatial location as accurately as possible (Ryals, 2011).

STN and WISLR are similar in that both lack a distinct datum. Instead, each system has the datum embedded in the respective network. Explicitly, the network and datum are functionally equivalent. Additionally, each system employs a different LRM (A. J. Graettinger et al., 2009; A. J. Graettinger, et al., 2008). The WISLR system is described best by an integrated method of network design. That is, WISLR consists of segmented routes that provide more calibration in displaying measured transportation data than long, multiple-intersection spanning features. The STN system could be described similarly, but it typically allows multiple road features to comprise one link.

WisDOT necessarily maintains its LRSs by updating line work along with attributes or databases. Extensive tools are implemented for this effort. Continuous update of the two systems allow for STN and WISLR to be available for uninterrupted use and analysis.

WisDOT now seeks to use the functional join between STN and WISLR, basically the link-link table, to move from an event on a WISLR chain, or cartographic representation, to an event containing an STN link and offset. This translation is direct when a one-to-one relationship between the two systems can be defined by the link-link table. However, ambiguities occur when one-to-many relationships exist.

The RP system discussed previously would be considered a linear referencing method (LRM) as defined by NCHRP. LRMs allow measurements along some type of linear element. Each RP in the RP system is an LRM with an absolute distance defined. The location of the point is given with an offset distance from the beginning of the STN link on which the RP exists. The NCHRP model allows for translation expressions to convert traversals, or routes through a part of a transportation network made up of one or more segments, from a LRS to cartographic representation lines (Dueker & Butler, 2000; Scarponcini, 2002).

Given that the WISLR and STN systems exist on different scales, or resolution levels, linking the data sets had to utilize non-arbitrary rules to describe relationships within the data (Sester, et al., 1998). During link-link coding, rules were formulated in this way but only with consideration of one direction: STN to WISLR (Ryals, 2011). STN-to-WISLR relation was accomplished by linking the two systems with the link-link table. The design and basic description of the link-link table is illustrated in Table 2.1. The STNid and WISLRid columns were populated with the unique ID number of the STN and WISLR link lengths that were found to correspond to each other. The STNstart and STNend columns identify the section of an STN link that corresponds to the section of the WISLR link identified by the WISLRstart and WISLREnd columns.

Table 2.1. Names and Descriptions for the Six Main Link-Link Columns and Three Relevant Link-Link Flag Columns (Ryals, 2011)

STNid	STNstart	STNend	WISLRid	WISLRstart	WISLREnd	T	M	W
Unique identifier for the STN link	Start measure for the STN section	End measure for the STN section	Unique identifier for the WISLR link	Start measure for the WISLR section	End measure for the WISLR section	Flag for Turn Lanes	Flag for Median Crossovers	Flag for waysides

Additionally, rules were established to deal with areas that did not directly correspond to each other. Rules associated with the link-link table were implemented with the use of flag columns within the table. Although five main flag columns were used during coding of the link-link table, three are relevant to this research: turn-lane (T), median crossover (M), and wayside (W). The turn-lane column identified areas in which STN represented an intersection with physically separated turn lane links, while WISLR did not. The median crossover column similarly identified areas in which STN represented median crossover with a link, while WISLR did not. The wayside flag identified roadside areas that were represented in STN with links, but were not represented in WISLR. The flag columns were created to manage discrepancies such that every point along an STN link was able to translate to some point along WISLR.

Although not previously considered, rules must be determined to similarly move data sets in the opposite direction, from WISLR to STN. One way to determine these rules is to visually inspect existing data sets and determine regularities between the data. Problems arise in this approach because of the difficulty to find and determine every possible exception to the rules (Sester et al., 1998).

Data translation between linear referencing methods is said to occur in the following way. An offset from a reference point such as a WisDOT RP with an STN link and offset is transformed into an offset along an anchor section and subsequently transformed into reference point in the other system such as in WISLR with a WISLR link and offset (Vonderohe & Hepworth, 1998).

In earlier research regarding data sharing between WISLR and STN, it was found that a methodology could be successfully implemented between two large LRSs while each system continued regular independent functions. This methodology was refined with quality assurance

and quality control measures that allowed full crash data sharing from the STN to WISLR systems with limited sharing from WISLR to STN. This research seeks to refine data translation techniques in the opposite direction, from WISLR to STN.



## CHAPTER 3

### METHODOLOGY

#### 3.1 Introduction

This research was performed in order to design, analyze, and apply a technique for updating a previously-defined functional merge between the State Trunk Network (STN) and the Wisconsin Information System for Local Roads (WISLR). These two linear referencing systems (LRSs) were designed and are maintained by the Wisconsin Department of Transportation (WisDOT). STN corresponds to state routes only, while WISLR corresponds to all roads with specific emphasis and detail paid to local roads. Previous research centered on defining a functional merge between the two systems to allow for more complete analysis of statewide business data. Within this business data, the most attention was paid to crash data. WisDOT business data can include pavement data, bridge data, and other information.

The previous research created a functional LRS merge through the “link-link” table. The link-to-link merge method matched a section of a link in one system to a section of a link in the other system. This table was systematically hand populated in a county-by-county manner. Each completed county was then appended to a statewide link-link table. The table was created based on 2009 line work and data provided by WisDOT. However, roadway construction, line work editing needs and changes in roadway attributes within WISLR and STN necessitates a constant update and enhancement of the link-link table in order to maintain robustness of the

merge between the two systems. To incorporate various updates and changes within the roadway network into the link-link table, it is necessary to define an updating process.

Due to the need for data analysis within both STN and WISLR and to the fact that the resolution level between the two systems is dissimilar, data translation procedures must be formulated. Data translation rules can assist the accurate translation of information between the two LRSs. To avoid needing to formulate data translation rules WISLR, the lower resolution system, could be upgraded to implement higher detail. This can be done by identifying areas that require edits by WisDOT personnel. A methodology to formulate this identification process is also necessary.

This chapter discusses details of the STN and WISLR systems and briefly details the basic link-link table population process. Next, how date information was incorporated into the table to accommodate table updating is described and the methodology used for updating the link-link table is discussed. Then, issues that arose during updating of the link-link table are discussed. The modified Quality Assurance/Quality Control programs and processes that were used for accuracy verification in the table are also described. Data translation rules that can be used to move data from high resolution STN to low resolution WISLR and back are discussed. Finally, this chapter discusses how areas of WISLR that need editing due to discrepancies will be identified for correction.

### 3.2 The STN and WISLR Systems

The State Trunk Network (STN) was implemented in 1993 by WisDOT to manage and sustain transportation information on state highways and interstates in the State. STN is made up of links and nodes along state-controlled roads only. In 2002, WisDOT implemented the

Wisconsin Information System for Local Roads to accommodate federal mandates requiring more complete road network information reporting to allow for more complete maintenance and safety analyses for all roads throughout the states. An example of how STN and WISLR relate to each other is shown in an example in Figure 3.1. The STN system (thick red lines) is overlaid with the WISLR system (thin black lines) in and around Madison, Wisconsin.

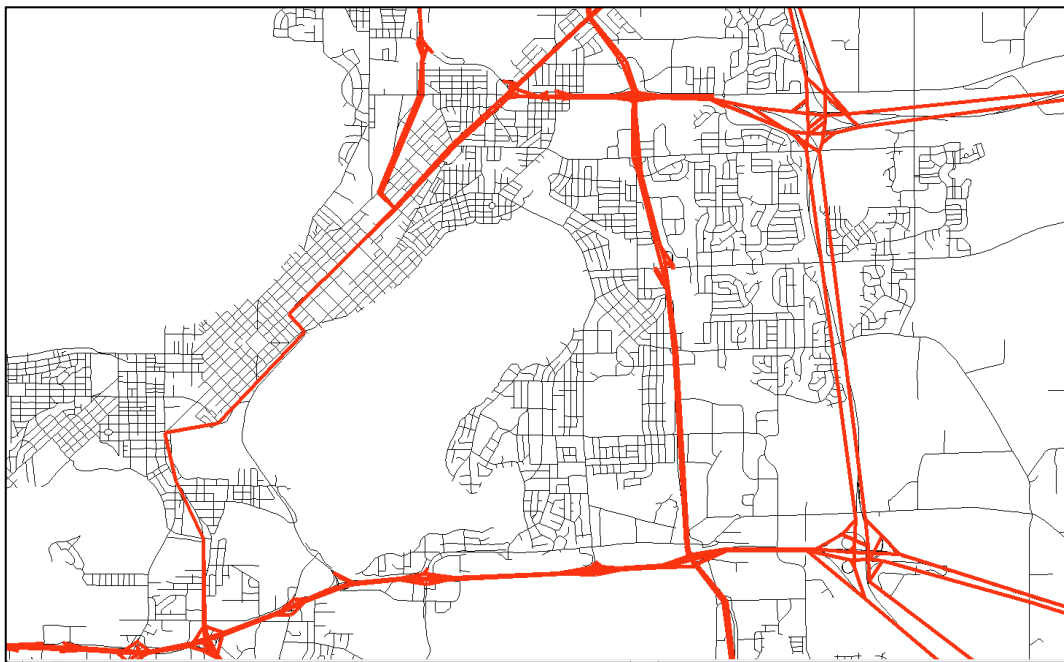


Figure 3.1. Example of the STN/WISLR relationship from the Madison, Wisconsin area. STN is represented by thick red lines, while WISLR is represented with thin black lines

STN contains links and sites (or nodes), cartographically-representative chains, and transportation business data. Every direction of state roadway is represented in STN with a link containing attributes such as: a link ID, length measurement, from-site, and to-site. The sites determine directionality of links and are stored as link attributes. STN link distance is stored in thousandths of a mile. However, the links are only measured to the accuracy of a hundredth of a mile. Additionally, STN links are not cartographically representative. Instead, they are drawn

from one node to another node using straight lines. Nodes are placed where a turn from one state owned road to another state owned road can occur.

WISLR contains cartographically-representative links, sites, and business data. Every direction of every roadway throughout the state is represented in WISLR. However, WISLR focuses more on local road than it does on state roads. Typically, sites in WISLR represent roadway intersections. WISLR lengths are stored in feet and are recorded to the nearest foot.

### 3.3 The Basic Link-Link Coding Process

The link-link table essentially matches sections of links in STN to sections of links in WISLR. An example of the basic link-link coding process is presented in Figure 3.2. Figure 3.2(a) shows a basic, one-direction example of a roadway section represented in STN and WISLR. Figure 3.2(b) and (c) shows relevant STN and WISLR link attribute information needed for link-link coding. Figure 3.2(d) presents a basic coded section of the link-link table. Each row (or record) of the table is believed to be one equal segment of pavement described in two separate LRSs (STN and WISLR). The first three columns of the link-link table relate to the STN system, while the last three columns relate to the WISLR system.

The coded link-link data presented in Figure 3.2(d) is populated using the basic link-link coding process. As can be seen in the figure, STN link A relates to four WISLR links: b, c, d, and e. Initially, the only data known is the total STN link length (200), and the corresponding WISLR link lengths. This allows for the table to be initially populated with four lines. The first line is populated with the STNid (A), the STNstart (0), and all three of the WISLR link attributes for the first WISLR link corresponding to STN link A. The next two lines are then populated with the STNid and WISLR link attributes only. Next, the fourth line of data is populated with

the STNid, the STNend (200), and the WISLR link attributes for the last WISLR link corresponding to STN link A.

To populate the missing offset values (line 1 STN end, lines 2 and 3 STNstart and STNend, line 4 STNstart), multiple approaches can be used. If an access point exists on the STN link at points corresponding to each WISLR node, the offset values related to the access points can be used for the offset values. Access points exist to denote driven offset information for intersections occurring along state routes. However, in this case, the ratio approach is used. This approach uses ratios equal to each WISLR link length divided by the sum of WISLR link lengths to determine equivalent STN offset values for each WISLR link. For example, WISLR link b corresponds to the length of link b (158) divided by the total corresponding WISLR length ( $158+304+164+430=1057$ ) multiplied by the length of STN link A ( $158/1057 * 200 = 30$ ). This offset value of 30 is populated into the STNend field for line one and the STNstart field for line two. This link-link table population procedure is repeated for each link-to-link relationship.

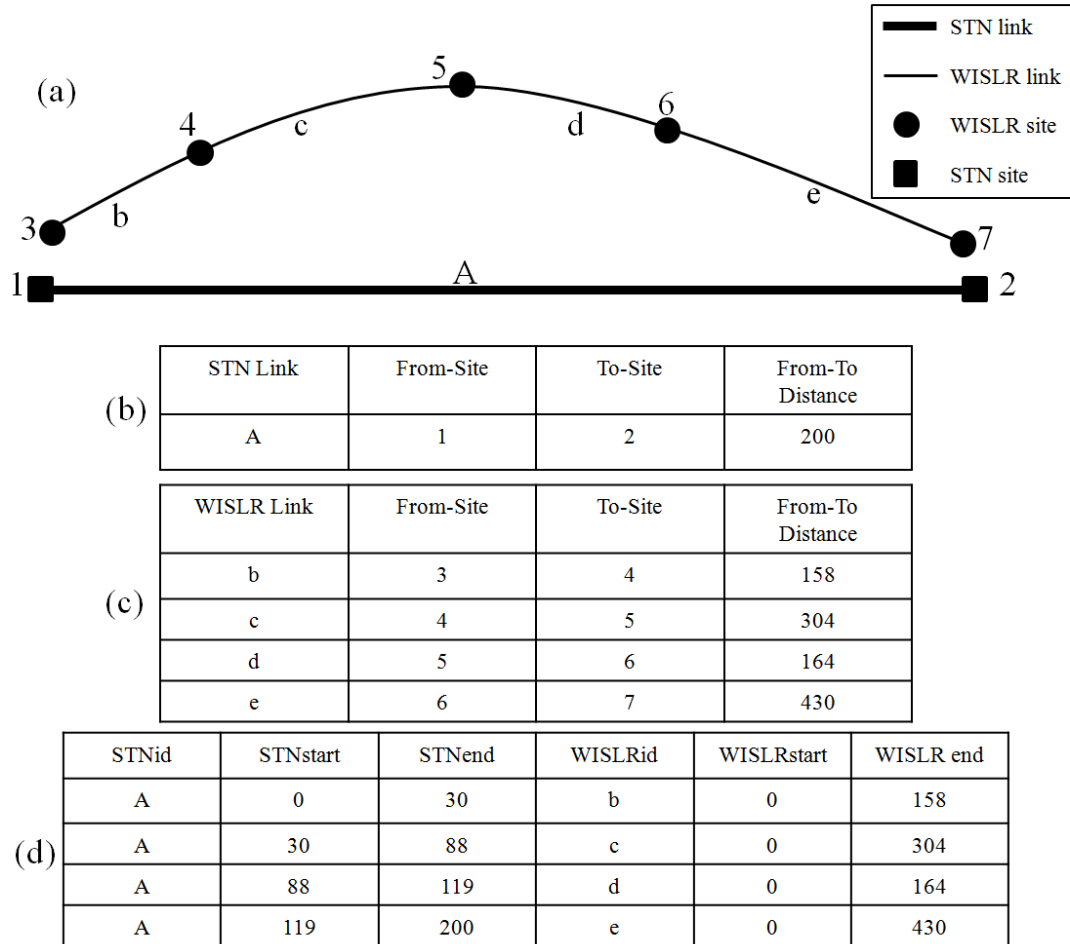


Figure 3.2. Example of the basic link-link table coding process: (a) illustrates a basic roadway representation in WISLR and STN, (b) and (c) show basic attribute information from the shown links, and (d) demonstrates the basic link-link table associated with the shown representations

Various discrepancies are known to occur when trying to relate STN and WISLR. To accommodate these known differences, five flag columns were used in link-link coding: turn-lane, median crossover, gore point, wayside, and problem. The turn-lane flag column exists to flag any location in which STN represents an intersection with a turn-lane, when WISLR does not. That is, the resolution of STN is higher than that of WISLR. This means that a one-to-one relationship is not possible at a certain location. The median crossover flag column exists for a similar reason; because STN represents a median crossover at an intersection for example, where WISLR represents the same location with a single point. The gore-point flag exists to flag any

location at which a gore point exists. Gore points occur anytime roads merge or split. Because of differences in measuring techniques, gore situations provide distinct length discrepancies between STN and WISLR. The wayside flag column exists to identify any location in which a wayside, or a roadside area, is represented in STN and not in WISLR. Finally the problem flag column exists to identify any record with a problem not covered in the other four columns. There also exists a comments column to explain the problem encountered. While the basic make-up and population of the link-link table is straightforward, there are caveats that exist in the process. The basic coding process has been extensively presented by Ryals (2011). More detailed information regarding original link-link coding can be found in the Link-Link Table Population Manual by Ryals (2011).

### 3.4 Incorporating date information into the Link-Link Table

To accommodate yearly updates of the link-link table, it is necessary to incorporate date information into the link-link table. This was accomplished by not only modifying the existing date column (“datemod”) in the table, but also adding three additional date columns. In the updated link-link table, there are now four date columns: Record\_Created, Record\_Historic, Start\_Valid, and End\_Valid. Each of these new columns will be presented in detail, followed by how the dates within the columns are populated in link-link coding and used in data translation.

#### *3.4.1 Record\_Created Column*

The Record\_Created column functions the same as the datemod column in the previous link-link table layout. This column is populated by a coder and corresponds to the date on which any record is created or entered into the link-link table. The dates that were entered into the

datemod column in original link-link coding were retained to populate this column for original records in the table.

### *3.4.2 Record\_Historic Column*

The Record\_Historic column is populated by the dates when a particular row in the link-link table becomes historic. The dates in this column are generated by a coder. A record could become historic for many reasons. The reasons include changes to STN or WISLR link line work, changes to an access point (i.e. the offset measure of the access point), and the discovery of an error in the original link-link record. Entering dates associated with various data modifications will be presented later in this thesis. Generally, dates in the Record\_Historic column correspond to Record\_Created dates in updated or new records.

### *3.4.3 Start\_Valid Column*

The Start\_Valid column is populated with dates corresponding to the links contained in a record. These dates correspond to the more recent validity start date of the STN or WISLR link in a link-link table record and are drawn from database tables generated by WisDOT. The Route-Link (RTE\_LINK) database table from each system was selected as the basis for acquiring Start\_Valid dates.

The Route-Link database table contains information linking routes and links and contains every route-link relationship, for every link and route in the State. The data columns of interest in the Route-Link database table are the route-link current and route-link historic date columns. The route-link current date is the date on which the route and link combination contained in the record first became valid. That is, when the route *and* the link contained in the record were both valid in WisDOT databases. Similarly, the route-link historic date is the date on which the route



and link combination contained in the record first become historic. When *either* the link or route in the record becomes historic, the route-link combination becomes historic.

The validity start dates are drawn from the link-route current date in the route-link table associated with STN and WISLR. This date is independent of when the link or the attributes of the link were physically entered into the database.

Within each link-link table record there are two possible dates that could be used for the Start\_Valid date. The STN link in the record will have an associated link-route current date as will the WISLR link in the record. The date that is chosen to populate the Start\_Valid field in the link-link table is the more recent of these two dates. This date is chosen because the record should only become valid when both links in the record are valid.

#### *3.4.4 End\_Valid Column*

The End\_Valid column is also populated with dates corresponding to the links contained in a record. These dates correspond to the older validity end date of the STN or WISLR link in a link-link table record. These dates are also drawn from the route-link database table from each system. The validity start dates are drawn from the link-route historic date in the route-link table associated with STN and WISLR. This date is independent of when the link and the attributes of the link were physically retired from the database.

Within each link-link table record there are two possible dates that could be used for the End\_Valid date. The STN link in the record will have an associated link-route historic date as will the WISLR link in the record. The date that is chosen to populate the End\_Valid field is the

older of these two dates. This date is chosen because the record should only be valid when both links in the record are valid and not historic.

A detailed example of how these four date columns are populated is shown in sections 3.5.1, Simultaneous Date Column Population and 3.5.4, Updating Records in the Link-Link Table.

### 3.5 Populating Updated Information into the Link-Link Table

Four basic steps were taken to update the link-link table. First, dates for each link-link record were populated simultaneously. Second, updated line work was obtained from WisDOT and refined to obtain “current” data. Third, changes and updates to STN and WISLR were identified. Finally, the table was manually updated using a systematic approach. The complete update process is explained in this section.

#### *3.5.1 Simultaneous Date Column Population*

The first step required in updating the link-link table was the population of the four date columns that were added to the link-link table. Six separate queries were written within Microsoft Access to programmatically populate these date fields from the STN and WISLR route-link tables. Two queries pertained to the Start\_Valid column while the other four queries pertained to the End\_Valid column.

The two Start\_Valid queries compare the date related to the route-link current date in the STN route-link table with the date related to the route-link current date in the WISLR route-link table. The more recent date between the two was populated into the Start\_Valid field in the record. The four remaining queries relate to populating the End\_Valid column.

The first two End\_Valid queries determine if the links in a link-link record contain a date in the route-link historic column, i.e. if the STN *or* WISLR link in the link-link record is historic. If a historic date is found for only one of the STN or WISLR links in the record, then the historic date that exists in in the route-link historic attribute of that one link is populated into the End\_Valid field in the link-link record. The remaining two End\_Valid queries function similarly to the two Start\_Valid queries. That is, these queries compare the date related to the route-link historic date in the STN route link table with the date related to the route-link historic date in the WISLR route-link table. The older date between the two was populated into the End\_Valid field in the link-link record.

Before these six queries were executed, each route-link table was initially refined to ensure that each link was associated with only one record in the route-link table. This was done by the Traffic Operations and Safety (TOPS) lab using a query that retained records that did not contain current and historic dates that were equal to each other and also retained the minimum and maximum dates for the current and historic dates associated with each link, respectively. This “refined” table was then imported into Access for use with the six additional queries. Each of the six queries used in date identification can be seen in Appendix A.

For an immediate accuracy check following the execution of the six Start\_Valid and End\_Valid queries, rows with missing date data were identified. Each link-link record should contain at least two populated date fields, the Record\_Created and the Start\_Valid fields. Any record that did not have one or both of these fields populated was analyzed for an error and manually corrected. Any row that was populated with an End\_Valid date but not a Record\_Historic date was analyzed more closely because any record that is historic due to link

dates (i.e. the End\_Valid date is populated) should also have a populated Record\_Historic field. Each identified row that had this issue was checked for accuracy and changed.

### *3.5.2 Obtaining “Current” Data for Link-Link Updating*

The first step required in updating the link-link table for yearly line work and database changes within STN and WISLR is obtaining “current” data which was compared to previous data. First, 2010 data was obtained for link-link updating. With respect to current links, this was accomplished by looking at the STN and WISLR route-link tables. “Current” links were identified by selecting records where the route-link combination is current. For 2010 data, this was done by selecting records with a route-link current date earlier than 2011 and a route-link historic date that was null or newer or equal to the year 2011. A list of current links was obtained by summarizing this list of current route-links.

A shape file of current links was obtained by joining the lists of current links to the ArcMap coverage files generated by WisDOT and then exporting joined records to a new shape file. This shape file was then joined to relevant date information by joining the “refined” route-link tables with the previously created shape files. The shape files were then cleaned up to contain only relevant coding information. The relevant columns included link ID number, from and to site numbers, display name, and the date fields appended from the “refined” route-link table.

Current sites, or nodes, within STN and WISLR are obtained by performing a process similar to the process used to identify current links. WisDOT reference site tables (DT\_REF\_SITE) are used to obtain current sites within each system. The reference site tables contain every site for STN and WISLR across the entire state. Each site is uniquely numbered,

with location and validity date information in the site attributes. Current sites were identified for the 2010 update by selecting records with a route-link current date earlier than 2011 and a route-link historic date that is null or newer or equal to the year 2011.

Current access points were obtained more easily. The WisDOT access point table (DV\_ACSI\_PT) contains all of the statewide access points and the location points' location information. Location attributes are stored as latitude, longitude, and STN link-offset. However, each access point is not uniquely identified. This means that the access point table is not embedded with date information. As new access point data becomes available, the table is simply updated with new access points, and the outdated access points are physically deleted from the database. Moreover, the updated (2010) access point table was used in its entirety and considered as current access point table.

### *3.5.3 Identifying Changes and Updates to STN and WISLR*

After current data was obtained, it was necessary to identify what changes occurred between data used in initial link-link coding and current data. There are six basic changes that directly affected records in the link-link table. These changes are the addition of a new STN link, a new WISLR link, or a new access point, and the deletion of an STN link, a WISLR link, or an access point. Identifying these changes was accomplished completely within ArcMap, using joins.

First, new links were identified. This was done by joining old line work (i.e. data used in the formation of the 2009 link-link) to new line work (i.e. data that was created in the previous section) based on the link ID number. In the joined data, records that did not join were selected. Selecting non-joined records identified new links, that is, links that have been added to the “old”

STN and WISLR data sets. These selected links were then exported into a new shape file that is named “NEW\_(data)\_2010,” where (data) is the data source being analyzed (STN links/sites, WISLR links/sites, access points, etc.). This process was repeated for STN and WISLR links.

Second, deleted links were identified. This was done by joining new line work to old line work based on the link ID number. In the joined data, records that did not join were selected. Selecting non-joined records identified deleted links, that is, links that have been deleted from the “old” STN and WISLR data sets. These selected links were then exported into a new shape file that is names “DELETED\_(data)\_2010.” This process was repeated STN and WISLR links.

Finally, changes to access points were identified. Identifying these changes required a slightly different approach. Initially, to aid in identification, a column named “concat” was created in the old and new access point data tables. This column was populated with a text concatenation of the link ID (RWLK\_ID), intersecting road name (ASCI\_INTS), and offset value (ASCI\_PT1) associated with each access point. The “concat” column was then used as the basis to join “new” data to “old” data and vice-versa, similarly to the previous process for link identification. Again, un-joined records in each step were selected to identify new and deleted access points, respectively. Next, the selected records were summarized within ArcMap based on the STN link ID on which the access point exists or existed. The combination of the two summarized lists created a full list of links affected by access point changes, deleted or new. This full “affected links” list was then joined to the new STN links shape file previously generated. Finally links associated with joined records were exported to generate a shape file representing only links that have been affected by changes in access points. This new shape file was named “AP\_AFFECTED\_STN\_LINKS\_(year).”

#### *3.5.4 Updating Records in the Link-Link Table*

The process to update the link-link table proceeded very similarly to original link-link coding, with a few exceptions. First a county to update was identified. The relevant shape files were then imported into the working map. These relevant shape files included current data sets for STN links, sites, and chains; WISLR links and sites; access points; new and deleted STN links; new and deleted WISLR links; access-point-affected STN links; and statewide county outlines. County-specific data was obtained by using the “select by location” feature in ArcMap with the “intersects with the source layer” option, using a search radius of 1000 meters. Selected links from each shape file were then exported to county-specific shape files to use for the update. An example of this selection procedure regarding STN links in Dane County is illustrated in Figure 3.3. Figure 3.3(a) and (b) select Dane County as the relevant county for updating. Figure 3.3(c) shows the “Select By Location” tool user interface and the needed selections and values. Figure 3.3(d) shows the STN links selected using the tool.

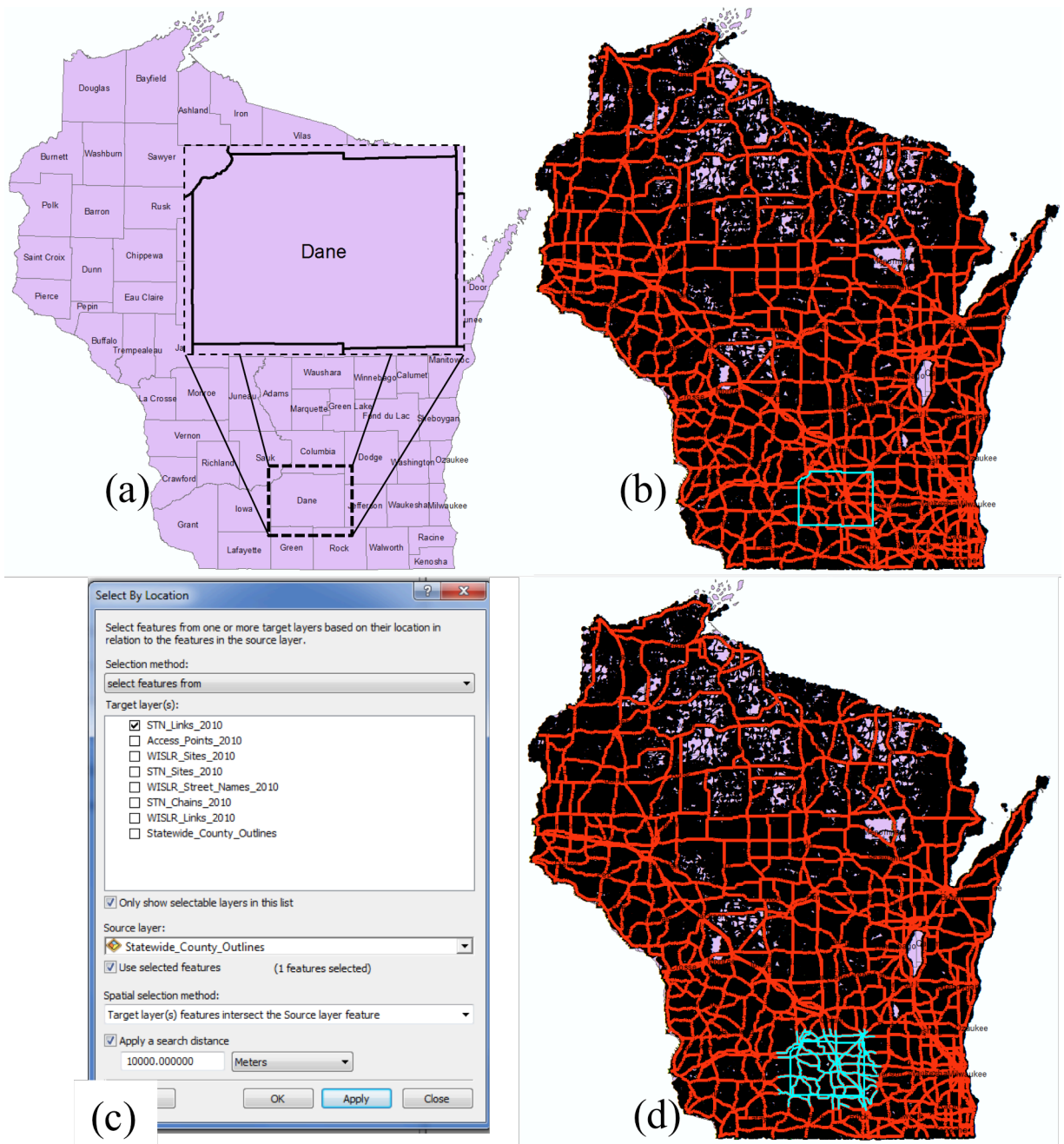


Figure 3.3. Example of county-specific data selection procedure: (a) and (b) select Dane County as the relevant county for updating, (c) the “Select by Attributes” user interface window, (d) Dane County STN links selected using “Select by Attributes”

Next, a copy of the “old” link-link table was obtained and saved in the folder with the other files being used in the county update. This copy was saved with the name “(County)\_link-link\_(year).” The copied file was then open and all records that were not coded to the county



that being updated were deleted; deleting these records left only records that were coded to the county being updated. An excel spreadsheet that served as the working document for the updating process was then created. This spreadsheet contained the same column headings that are contained in the updated version of the link-link table (i.e. with the four date columns).

After data acquisition, the physical update process began. The general update process proceeded as follows. County line work was analyzed, starting with the STN links (new and deleted). Looking first at the deleted STN links shape file, each deleted link was identified. The record(s) associated with each link ID was cut from the county-specific copy of the link-link table and pasted in the excel spreadsheet used for editing. In the excel spreadsheet, relating to the link-link record associated with the deleted STN link, the Record\_Historic field was populated with the date of the edit (i.e. the date that the record was cut from the link-link table), and the End\_Valid date was populated with the earliest route-link historic date associated with the links in the record. In the case of a deleted STN link, this date was the route-link historic date associated with the STN link. This occurred given that the STN link was the only link in the record that has been updated (i.e. the WISLR link in the link-link record had not been edited).

There was, in most cases, a new STN link associated with locations where an STN link had been deleted. The new STN link is entered similarly to how original link-link coding proceeded. In this case, the Record\_Created date was the date that the record was entered into the link-link table. The Start\_Valid date was populated with the route-link current date associated with the STN link, given that the WISLR link in the record was not edited later than the date when the STN link was created. An example of the date population procedure associated with link-link table update due to a change in STN links can be seen in Figure 3.4.

Figure 3.4(a) illustrates a basic example section of roadway showing a new and deleted STN link associated with an unchanged WISLR link. Figure 3.4(b) and (c) show example attribute information for the STN and WISLR links, respectively. Figure 3.4(d) presents an example of how the change is coded into the link-link table.

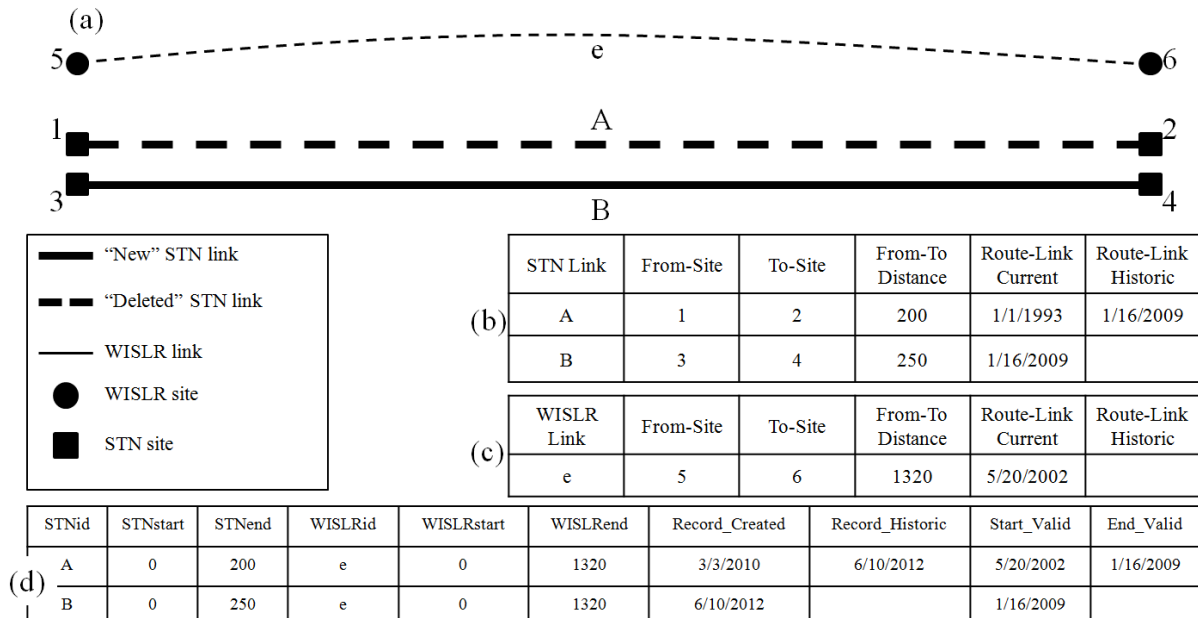


Figure 3.4. Example of data population procedure associated with link-link table update due to a change in STN links: (a) example roadway section with a new and deleted STN link and unchanged WISLR link, (b) attribute information for the STN links in the roadway section, (c) attribute information for the WISLR link in the roadway section, (d) example link-link table entry for old and new STN link in the roadway section

After all new and deleted STN links had been viewed and subsequently updated in the Excel editing spreadsheet, WISLR links were analyzed. WISLR changes were addressed in the same way that STN changes were addressed. However, some WISLR link changes occurred in the same location as STN link changes, so some WISLR changes were accounted for in the STN updating stage.

After all WISLR changes were identified and reflected in the link-link table, access point changes were investigated. Using the previously created access-points-affected-STN-links shape file, STN links that were associated with access point changes were identified. Once a link had been identified, the access points along the link were investigated. At this point, it was necessary to identify whether any of the access points along a link represented a difference from offsets already used within the existing record(s) in the county-specific link-link table. If a change was found that does affect link-link, the link-link record associated with the changed-access-point was cut and pasted into the Excel file similar to the previous update method for STN and WISLR. Additionally, the link record associated with the direction opposite of the changed access point was also updated. This was done because typically access points are defined for only one direction of a roadway, but apply to both directions. Given that access points are frequently updated to reflect accurate offset information, dates associated with changes to access points on STN links must be managed differently than earlier presented link modifications.

Link records that contained offsets that, according to the updated access points along the link, no longer reflected accurate offset information were made historic. This was accomplished by populating the Record\_Historic column with the date of the record edit. The End\_Valid date field was populated with the same date as was present in the Start\_Valid date field in the link-link record. The new records associated with updated offset information from the updated access points along the link were populated with a Record\_Created date equal to the date of edit and a Start\_Valid date equal to the Start\_Valid/End\_Valid date on the outdated record for the link that was updated.

When all identified changes were reflected in the working Excel spreadsheet, all records were copied and pasted into the county-specific link-link database table.

### 3.6 Quality Assurance/Quality Control (QA/QC)

Similar QA/QC procedures are used in updating the link-link table as were used in original link-link coding. However, some basic differences were needed to accommodate the nature of updating the table. The four basic QA/QC procedures used to assure accuracy of the updated link-link table were the STN and WISLR Link visual checks, Gore Point check, and the XY Connector Line check. All of these checks were established during original link-link coding, but will be briefly discussed with basic differences detailed. The first three checks require a modified version of the link-link table to ensure that only current records in the link-link table are used in accuracy checking. The fourth check requires the use of a modified point moving program that is explained following the discussion of the checking procedure.

#### *3.6.1 STN Link Check*

Keeping with the format and procedure associated with original link-link coding, all STN links in the state must be coded into the link-link table and subsequently updated as needed. The STN link check involves adding the link-link Access database table to the editing map and joining the table to the STN links shape file based on STN link ID. Every link should join since each STN link should be used in coding. Any link does not join in the process (i.e. an STN link has a null value in the joined information) is not represented in the link-link table. This process identifies un-coded STN links. These un-coded links should be added to the link-link table.

One modification was implemented in this check. In order to check the accuracy of only the current records in the link-link table, a “trimmed” version of the link-link table is obtained

for the county being checked. This was accomplished in Microsoft Access by saving a copy of the particular link-link table as a “trimmed” version, then opening the “trimmed” version and deleting all records that have a populated Record\_Historic field. Once this was performed, the “trimmed” table contained only records with a populated date in the Record\_Created and Start\_Valid fields. The “trimmed” table was then joined to the STN links shape file as was previously presented in basic STN link check.

### *3.6.2 WISLR Link Check*

To ensure that all state route WISLR links are used in link-link coding, a visual check is performed. This is accomplished by joining the link-link Access database table to the WISLR link shape file based on WISLR ID. After the join is complete, the symbology of the WISLR link shape file is adjusted to assist in visual inspection. This is done by making the symbol for joined records one color and the symbol for un-joined records another color or invisible. After the symbology is adjusted, it is possible to pan across a county to assure WISLR connectivity along state routes and also that no WISLR links not representing state routes are used in the link-link table. In updating the link-link table, the WISLR link check also required the use of the “trimmed” table created in the previous check.

### *3.6.3 Gore Point Check*

Gore points typically occur where roads merge or split at an intersection. Records that reflect where gore points occur are flagged in the link-link table. The gore point check ensures that each gore point situation has been flagged in the link-link table. This check is performed by first joining the link-link table to the STN links shape file based on STN link ID, then changing the symbology of the shape file. Each link should be given a different colored symbol based on

whether the link is coded as a to (T), from (F), or both (B) gore point situation. This allows for a quick visual check to ensure that each gore point situation is coded correctly. In updating the link-link table, the gore point check also required the use of the “trimmed” table created in the first check.

#### *3.6.4 XY Connector Lines Check*

Perhaps the most powerful check associated with original link-link coding as well as updating the link-link table is the XY connector lines check. This check consists of moving data points from STN to WISLR and checking how the points spatially relate to each other. It is a check of the basic link relationships defined in the link-link table.

The process associated with the XY connector lines check begins by creating data points along all the STN links in a county every hundredth of a mile, or every ten units in STN measurements. This is accomplished using the previously created “STN Points Generator” program (Ryals, 2011). Inputs required for this program to function are an Excel file containing current links, offsets, and associated county for the county being coded and the link-link table associated with the county being coded. This program and process functions the same as in original link-link coding, however a current link list had to be generated for use by the program. The program creates a table (the “STN\_Points” table) within the link-link database file that contains a unique identifier column, an STN link column, and an STN offset column. The table is populated with as many data rows as is necessary to create a uniquely identified point every hundredth of a mile on every current STN link in a county. The full, or “untrimmed,” county link-link table (current and historic records) was used for this program because the program populates the STN\_Points table in the link-link database based solely off of the Excel file

containing current links, full link lengths, and the county associated with each link. As long as the list of links in the Excel file is current, the current or historic status of a link-link record is irrelevant at this stage.

The next process involved in performing the XY connector line check involves moving, or translating, the STN points to WISLR points. This is accomplished by using “WISLR Point Moving Program” created in the previous link-link coding procedures (Ryals, 2011). The WISLR Point Moving Program requires only the link-link table as an input. The program functions by taking each data point created in the “STN\_Points” table and calculating an equivalent WISLR link and offset based off relationships existing in the link-link table. This program functions similarly in updating the link-link table to how the program functioned in original link-link coding. However, because some relationships can be duplicated in an updated link-link table, in the case of an access point change or corrected error affecting the table, it was necessary to perform one of two possible modified approaches to the WISLR point moving process. One approach used the original “WISLR Point Moving Program” with the trimmed version of a county link-link table as the input file. This approach moved points correctly, using only current link-link records. The other approach used a modified WISLR point moving program with the full version of a county link-link table as the input file. A modified WISLR point moving program was created in this research and is explained in detail in the next section. This approach also uses only current link-link records, without the need for trimming the link-link table.

Both WISLR data translation approaches used the same basic process. The program obtained each uniquely identified point from the “STN\_Points” table with the STN link and

offset associated with each point. The WISLR point moving program then used link-link table relationships to determine the WISLR link and offset that is functionally equivalent to the STN point and offset being analyzed. The program then creates a table named “WISLR\_final” that contains the same uniquely identified point numbers from the STN table, with the equivalent WISLR link and offset associated with each point.

At this point, the “STN\_Points” and “WISLR\_final” tables were used to create the lines to check the link-link table for accuracy. First, these two tables were added to the ArcMap editing map that was used in coding the updated link-link table. Each table was then physically displayed using the “display route events” feature within ArcMap. This feature requires a route file for STN and WISLR links to display the points existing in the STN and WISLR points tables within the link-link database. These two event files were then exported as shape files named “STN\_Points” and “WISLR\_Points” respectively. The “STN\_Points” shape file was then opened and two columns were added: “STN\_X” and “STN\_Y.” Each column was then populated with the coordinate of each point using the “calculate geometry” function, using the coordinate system of the data frame. The “WISLR\_Points” shape file was then opened and two columns were added: “WISLR\_X” and “WILSR\_Y.” Each of these columns was similarly populated with the coordinate of each point using the coordinate system of the data frame.

Once the x and y coordinates were populated in each shape file, the STN and WISLR points shape files were joined together based on Unique\_ID. The attribute table associated with the joined shape files was then exported and named “XY\_Connector\_Table.” The connector table was then used to create the XY connector lines. Before creating the lines, a blank shape file named “XY\_Lines” was created in the same folder as the other county link-link coding files.



Next, Geospatial Modeling Environment program created by Beyer (Beyer) was used with defined command lines within the program. The “XY\_Connector\_Table” was then translated to lines in the “XY\_Lines” shape file. The command line used to accomplish this task can be seen in Figure 3.53.5, and a visual example of “XY\_Lines” is shown in Figure 3.6.

```
convert.tabletolines(in(file path for "XY_Connector_Table"), uidfield="Unique_ID",
fromx="STN_X", fromy="STN_Y", tox="WISLR_X", toy="WISLR_Y", out(file path for
"XY_Lines" shapfile))
```

Figure 3.5. Command line used for “XY\_Lines” generation

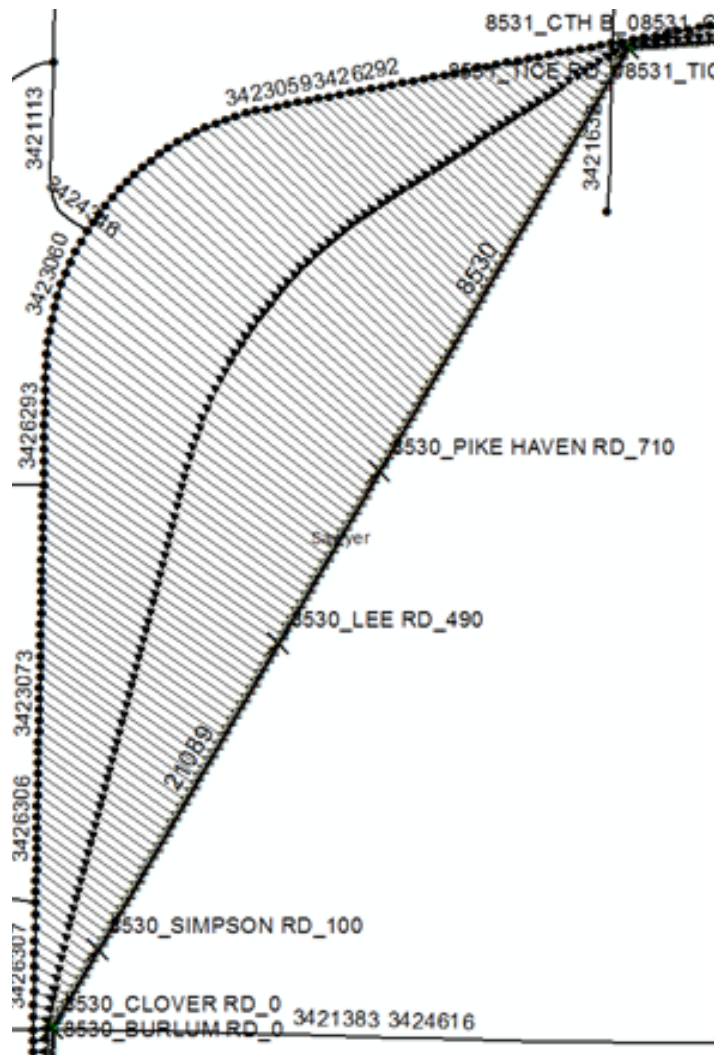


Figure 3.6. Example of “XY” lines shape file. Lines connect related data points on STN (line on right side) and WISLR (line on left side)

### *3.6.5 Modified WISLR Point Moving Program*

A modified version of the WISLR Point Moving Program was developed in this research. The new code for the program can be seen Appendix B. The program works similarly to the previous version, however, one major difference is that the program decides which link-to-link relationships to use for computation based on the status of the record. A “Status” column within the link-link table is created and populate with a “C” or an “H” based on whether the link-link record is current or historic, respectively. The program then uses only records with a “C” status for computation. This feature allows for the use of the entire link-link table in data point moving. The modified version of the WISLR Point Moving Program was also modified to determine current records by the date columns, avoiding the need to create a new “Status” column in the table.

### *3.7 Update Issues*

Various issues were encountered during the update of the link-link table. The main issue encountered that created the most relational problems is due to the lag time between when STN updates line work and when WISLR updates line work. STN tends to update line work before WISLR reflects changes along state routes. This lag time causes some changes in STN to not have a good match within WISLR in yearly line work snapshots.

Another issue encountered in updating link-link relates to the date fields in the table. Some link dates did not seem to be valid, meaning a link had a route-link current date that occurred after the route-link historic date. This issue was due mostly to discrepancies within the route-link databases for STN and WISLR. It was determined to be an inconsequential problem because of its limited frequency of occurrence.

### 3.8 Using Link-Link for Data Translation

Previous sections have illustrated how the link-link table and the subsequently updated table can be used to move data from STN to WISLR. However, a methodology has not been refined to move data from WISLR to STN. Part of this research was to define data translation techniques for the WISLR to STN direction. The approach, data analysis, and process definitions of this effort are presented in the following sections.

#### *3.8.1 Approach*

Given that data sharing and translation from STN to WISLR was successfully accomplished through earlier phases of research with the link-link table, an approach to translate data from WISLR to STN was most practically accomplished by using the link-link table as well. As was the case for STN to WISLR translation, the ability of the link-link table to define link-by-link relationship data between the two systems was the building block for the WISLR to STN translation effort.

#### *3.8.2 Data Translation*

Basic data translation requires moving data containing one method of location definition such as an STN link and offset to another method of location definition such as WISLR link and offset. Translating data in this direction has been previously refined. The approach to translate data from WISLR to STN required a similar approach. This approach involves using mathematical relationships presented in link-link records to relate one location definition method to the other.

The first step in the WISLR-to-STN data translation effort was to create a general computer program that translated data containing WISLR link numbers and offsets into data

containing STN link numbers and offsets. The WISLR-to-STN program, modeled after the original STN-to-WISLR Point Moving Program, selects a table containing unique identifiers with a WISLR link and offset and finds a matching record in the link-link table that contains a matching WISLR link number. The program then determines, through mathematical ratios, where on the corresponding STN link in the link-link record the point should be located. The program then creates a new record in a new table with the same unique identifier as was read from the record being analyzed and populates an STN link and offset reflecting the relationship determined mathematically.

Three data sets were used to test the general data translation method from WISLR to STN. First, statewide RPs were moved from STN to WISLR then back to STN for initial data analysis. Next, hundredth-of-a-mile points along each STN link in Dane County were moved to WISLR then back to STN. Finally, Dane County crash points were moved.

Although moving points from WISLR to STN was successful, due to resolution issues some points landed at multiple ambiguous locations when moved back to STN. To find these points that mapped to multiple locations, the moved points were summarized within ArcMap based on the unique identifiers. This process produced a count for each point; a count greater than one indicated that the point from WISLR moved to multiple locations in STN. There were three common ambiguous point placement problems identified in this research. The problems were due to (1) median crossovers, (2) turn-lanes, and (3) waysides. These problems are discussed in the following subsections.

### *3.8.3 Problem 1: Intersection/Median Crossover Ambiguous Point Translation*

The first common issue causing ambiguous data point placement occurs because of intersection representations in STN and WISLR. This can be seen with an example intersection

illustrated in Figure 3.7. This example shows the intersection of two divided highways. As can be seen, WISLR (thin black lines) represents the intersection with a point, while STN (thick black lines) represents the intersection with four links. These four links represent median crossovers. As shown by the dashed arrows in the figure, data points at the ends of the median crossings move to a single point in WISLR (Figure 3.7(a)), in accordance with the link-link relationship. However, when moving back to STN (Figure 3.7(b)) each point is moved to every one of the four unique locations because one point in WISLR is represented by four nodes and four median cross over links in STN. This multiple point placement pattern does not represent the original location for all of the moved data points.

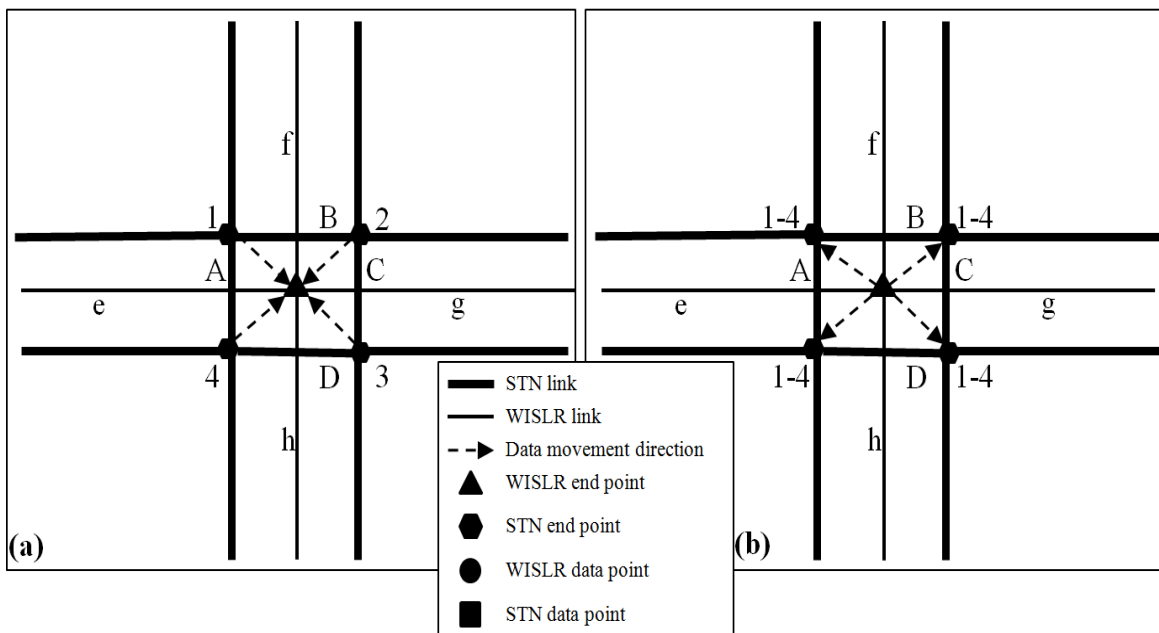


Figure 3.7. General example of ambiguous point placement due to incongruent intersection/median crossover representations: (a) example of data translation from STN to WISLR; (b) example of data translation from WISLR to STN.

### 3.8.4 Problem 2: Turn Lane Ambiguous Point Translation

A second common problem occurs at many state route intersections that include roadway entrance or exit ramps and other similarly designed intersections. This problem primarily

involves resolution differences between STN and WISLR at intersections with turn lanes. This issue can be seen in the example in Figure 3.8. Illustrated in this figure is an intersection where STN accounts for turn lanes with physically separated links while WISLR represents the intersection with a single link and single node. Link C in STN represents a right turn lane while link B represents a straight or left turn lane. Data points 1 and 2 move correctly to the intersection point represented in WISLR as shown in Figure 3.8(a). However, when these points are moved back to STN (Figure 3.8(b)), points 1 and 2 both move to each original location, and therefore, do not accurately represent the original conditions in STN. Additionally, data point C represents some event on STN occurring on link B. The point is moved to WISLR correctly, but then moves back to both turn lanes in STN. Given that the event did not occur in the right turn lane, data point 3 placed on link C is a completely erroneous data point.

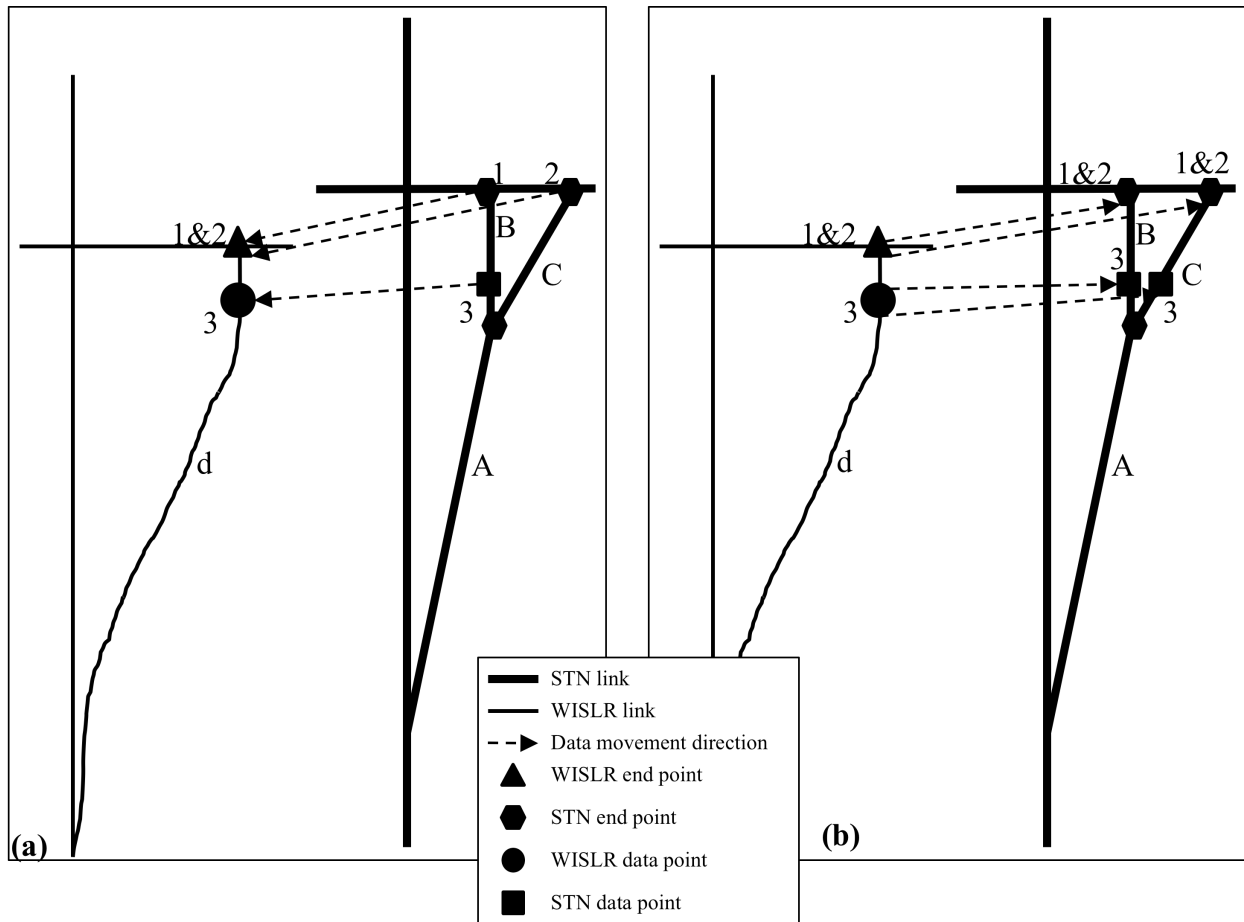


Figure 3.8. General example of ambiguous point placement due to incongruent turn lane representations: (a) example of data translation from STN to WISLR; (b) example of data translation from WISLR to STN

### 3.8.5 Problem 3: Wayside Ambiguous Point Translation

A final common problem that produced ambiguous data point placement situations was associated with waysides along state routes. An example of this problem is shown in Figure 3.9. In the figure, a wayside area is illustrated. In Figure 3.9, data points one through eight are translated to a single point on WISLR in accordance with relationships defined in the link-link table. This is because the wayside does not exist in the lower resolution WISLR system. When this single point on WISLR is translated back to STN (Figure 3.9(b)), all eight of the uniquely identified data points move back to the endpoints of the STN links on the wayside as well as the point on the “mainline” of the roadway. This point placement is again in accordance with

relationships defined in the link-link table. This example explicitly illustrates the resolution level difference between the two systems. Although the WISLR point moves correctly back to five unique locations, *every* point that moved to the WISLR point moves back to each of the unique locations. This causes eight data points to appear at each location. It is not desirable for all eight data points to move back to five different locations given that these five locations were not necessarily the “original starting point” of the STN data point. Because of discrepancies between the resolutions of STN and WISLR, detail is lost when moving between the systems.

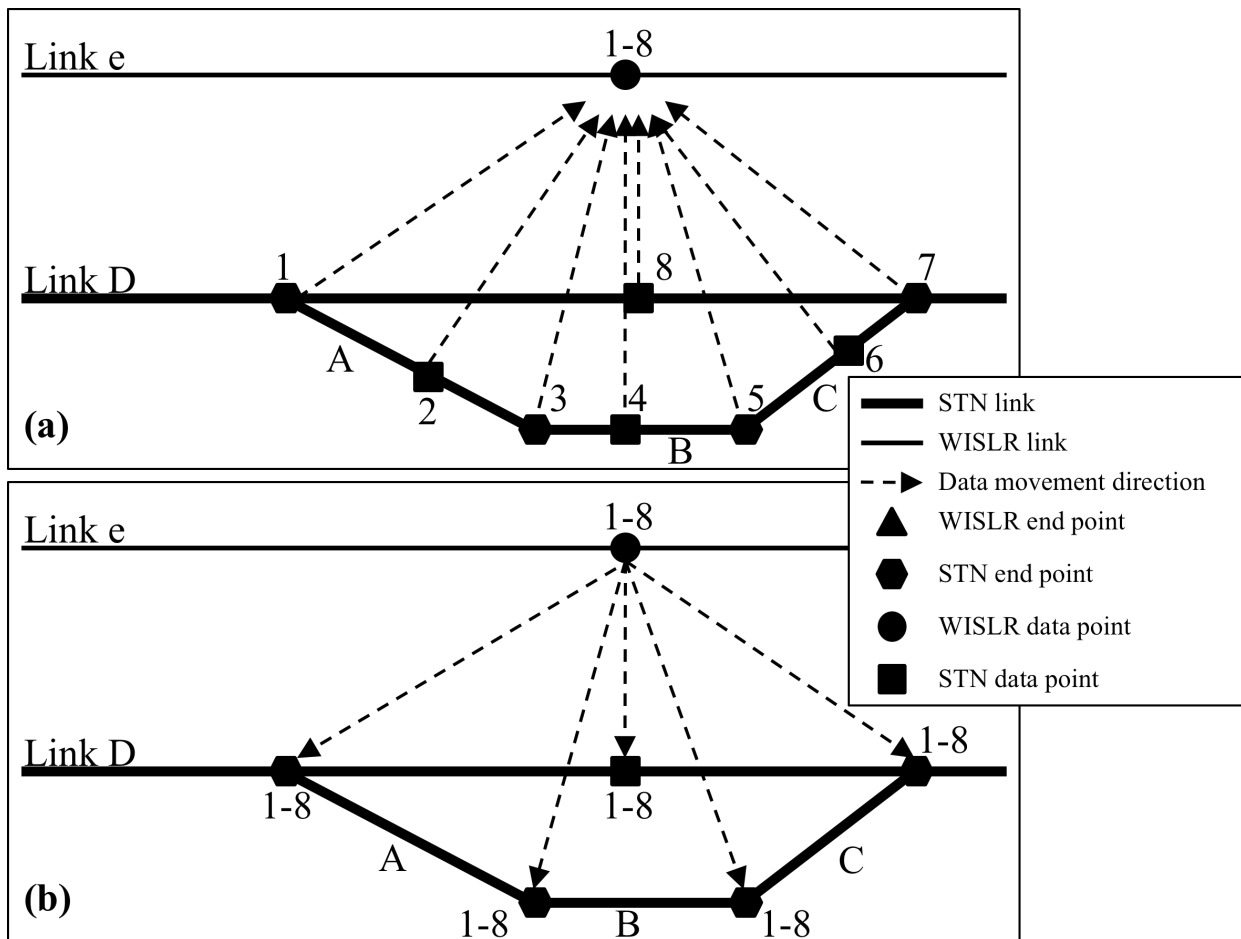


Figure 3.9. General example of ambiguous point placement due to a wayside: (a) example of data translation from STN to WISLR; (b) example of data translation from WISLR to STN.

To deal with ambiguous data placement, a process that simply flags crashes that move to ambiguous locations will be implemented in the short term. These crashes will then be manually



coded to STN using manual crash mapping methods. The objective of this phase of the research is to identify ways to avoid manual coding in order to save time and resources.

### *3.8.6 Proposed Data Translation Rules*

An absolute technique to eliminate ambiguities between the WISLR and STN systems, or between any two linear referencing systems with different resolution levels, is simply to improve the resolution level of the lower resolution system. This improvement will allow for complete one-to-one relationships between the two systems. However in light of the extensive time and effort associated with improving large linear referencing systems, it is necessary to address common resolution differences. To accomplish this, rules were established that are associated with not only linking linear referencing systems with different resolution levels, but also associated with how data is coded in the lower resolution system before the data is translated to the higher resolution system.

#### *3.8.6.1 Data Processing Rules*

The first potential rule relates to pre-processing of data before the data is translated from lower resolution to higher resolution. In the case of crash data, reporting intersection details in the crash record will allow for automatic location of the crash in both high and low resolution versions of the intersection. For example, in the interstate interchange shown in Figure 3.8, if the crash record denoted whether a crash occurred on the straight or right turn lane then an automatic routine could be coded to place the crash. If this is recorded on the front end of data translation, there is less ambiguity associated with moving data back to higher resolutions. A simple way to do this would be to note if a crash at a selected WISLR intersection point occurs in a median crossing. Additionally, at intersections that are represented with turn lanes in STN, the crash record could note which turn lane contains the crash. While this rule would be useful for data

translation, far-reaching changes would have to be made to crash reporting methods and forms requiring extensive time and effort.

To account for institutional constraints on implementing procedural changes in data reporting, additional rules were formulated that relate solely to data translation. Fortunately, the original link-link table was equipped with flag columns to identify common resolution discrepancies between STN and WISLR, allowing for easier rule implementation at these locations.

#### *3.8.6.2 Median Crossover Rules*

The second set of potential rules is associated with ambiguous data translation due to resolution differences at median crossovers. This problem causes points from WISLR (lower resolution) to move to multiple locations in STN (higher resolution). If an intersection point is selected in WISLR, there are multiple locations to which the location can be placed in STN. To reduce ambiguous event data placement at median crossovers, two rules can be implemented regarding median crossovers. The first median crossover rule would place each data point associated with a median-crossover-flagged-STN link at the center of the median crossing instead of the ends of the link. This rule would eliminate two points from being translated to the ends of the median crossover link. This rule could also be modified to place only one point at the beginning of a link flagged as a median crossover, instead of placing a point at each end. The second median crossover rule would ignore records in the link-link table associated with median crossovers when moving data from WISLR to STN. This would occur by ignoring all link-link records with a flagged median crossover field and creating a report with all of the data associated with a median crossover record for future manual placement.

### *3.8.6.3 Turn Lane Rules*

Four rules could be similarly applied to turn lanes. One rule for turn lanes could be implemented by simply mapping all crashes that would have mapped to two turn lane links to a single link. The single link to which a point would be placed would be arbitrarily determined from which data-associated STN turn lane was first processed by the computer program. The second rule could similarly implement single-link data placement. However in this rule, the longest turn lane link would be used for data placement. A third rule would cause the point moving program to ignore all turn-lane associated data and generate a report showing these data points. Finally, an additional table could be included in the LRS to indicate a preferred single location in the high resolution system for any data coming from a specific point in the low resolution system.

### *3.8.6.4 Wayside Rules*

The wayside rule allows for four user-controlled options regarding point placement at waysides. One option would place data points at all possible locations, basically not changing current data translation methods. The second option would place each data point at only one location. This location would be arbitrarily determined by which data-receiving valid STN link was processed first by the computer program. The third option would create a report with all of the data points that would have moved to a wayside-denoted STN link while not actually translating the data. Finally, an additional table could be created that would specify a preferred single location in the higher resolution STN system for any data point that originates from a specific location in the lower resolution system.

### *Rule Implementation*

To implement the previous rules when translating data from WISLR to STN, a point moving program was written. Radio buttons were added to a user interface that indicate how waysides, median crossovers, and turn lanes should be implemented in data translation. Each previously discussed rule is implemented in data translation simply by selecting one of the rule-related radio buttons on the user interface.

To test the general effectiveness of implementing data translation rules, one of the median crossover rules was implemented after the initial, rule-free data translation was performed. In this test, the second median crossover rule was implemented. This rule ignores all link-link records flagged as median crossovers.

### 3.9 WISLR Editing Link Identification

The five flag columns in the link-link table exist to track inconsistent areas, or discrepancies between STN and WISLR. In general, the flag columns identify pervasive resolution differences between STN and WISLR (i.e. turn lanes, median crossovers, and waysides). These three common problems generally require adding data to the WISLR system such that the systems' resolutions are equivalent. The problem flag column was created to identify WISLR/STN discrepancies not represented in the other flag columns. In combination with the comments columns, the problem flag identifies various problems associated with the WISLR and STN systems that directly affect the link-link table.

Analysis was performed on the comments associated with the problem flag column. The comments column was summarized within ArcMap to compile a problem list identified during original coding. It was found that a large majority of the line work problems flagged in the table

was associated with problems in WISLR. Because of this fact, it was determined that attention would be paid by WisDOT to updating WISLR to address these problems and increase the accuracy of the link-link table. It became necessary to identify the links that needed attention.

Correcting problem links was accomplished by first creating a concise list of general problems in categories. It was determined that the largest category of problems would be addressed first. The scale of link problems was determined by the number of links that each category.

The three largest issues that were reversed WISLR links, WISLR link length errors, and unnecessary WISLR links. Reversed WISLR links can be made up of one of a few conditions. The most common type of reversed WISLR is one in which the attribute information is correct, but the link is not physically drawn in the correct direction. These drawing-related reversed WISLR links were the most common problem. WISLR links containing length errors are links that appear to have errant measured length values. Related link length portions in the link-link table should relate with a common ratio, 5.28. This is given that STN is measured in hundredths of a mile while WISLR is measured in feet (1 hundredth of a mile = 5.28 feet). WISLR links containing measured length errors were considered to be those that did not relate approximately with this value. If the STN and WISLR portions were more than 53 feet different, the measured length of the WISLR link was determined to be an error. The measurement value of 53 feet was chosen because it is the equivalent of 10 STN units; 10 units in STN is the maximum accuracy level of measurements in STN. All links containing lengths that were different by at least 53 feet were added to a separate length error list. Finally, unnecessary links are those that are not used in the link-link table, do not represent a traveled roadway direction, and have no value being

included in WISLR line work. Unnecessary links commonly occur in two locations. The first occurrence is at entrance or exit ramps where both directions are represented with links while only one direction is a traveled way. The second occurrence is at divided highways and one-way streets. This occurrence similarly exists because two directions are represented by links, while there exists only one traveled direction in reality.

To generate the list of unnecessary WISLR links, the WISLR overlay table was used. This was because the overlay table contains vast amounts of information. The column that was used in unnecessary link identification was the “opposite link” column. This column contains the opposite link associated with each WISLR link in the database. The table was used in the following way. First, the link-link table was summarized based on WISLR link ID. This summarized WISLR link list was then joined to the overlay table based on WISLR link ID. Next, all of the joined records were selected and separated from the un-joined records. The “opposite link” column was then summarized and a list of opposite links associated with links used in the link-link table was created. Logically, all opposite links should be used in the link-link table. However, opposite links that were not used in link-link coding represent unnecessary links. That is, links that represent an untraveled direction in reality. To obtain a list these unnecessary links, the link-link table was joined to the summarized opposite link table. Any records that did not join were considered unnecessary links. A list was created containing the links associated with the un-joined records

### 3.10 Conclusion

This research centered on determining first how a functional merge between two LRSs could be updated to reflect frequently changing conditions. It was found that an update could be

performed given that date information was added into the merged data, and that some changes were made to the original link-link updating procedure. Secondly, this research sought to determine a methodology for translating data from a lower resolution LRS to one of higher resolution. It was found that, with the implementation of specific data processing and translation rules, data can be moved to distinct locations in the higher resolution LRS. This chapter discussed the WISLR and STN systems, the basic link-link coding procedures, changes made to the coding procedure to accommodate updating the table for changes in roadway network data, low-to-high resolution LRS data translation, and methods to determine WISLR links needing edits to aid in the accuracy of the link-link table. Chapter 4, Results, will explain how the link-link table update process was implemented, the progress of this implementation, results of the STN-to-WISLR data translation effort, and the success of determining WISLR links needing edits.

## CHAPTER 4

### RESULTS

#### 4.1 Introduction

The link-link table update technique is able to incorporate changes and updates to two linear referencing systems: STN and WISLR. This chapter presents the results associated with implementing the link-link table update methodology. Additionally, this chapter presents results associated with STN-to-WISLR data translation and the results from implementing data translation rules. Finally, results are presented regarding the volume of identified WISLR links requiring edits by WisDOT.

#### 4.2 Dane County Pilot Study for Link-Link Update

The link-link table coding procedure was formulated and tested in Dane County, the second most populous county in the State. After the pilot study began, it was found that Dane County also contained the largest number of required updates, identified with the methodology described in Chapter 3. An illustration can be seen in Figure 4.1 of the number of changes associated with each county in the State. As can be seen, approximately 9% of the 6274 identified Statewide LRS updates were found to have occurred in Dane County.

The Dane County pilot study was completed in approximately one month and incorporated all updates and changes made to WISLR and STN data between 2009 and 2010. In total, 377 records were turned into historic records, while 438 records were added to the table.



These changes were due to the addition of 74 new STN links, the addition of 12 WISLR links, the deletion of 54 STN links and access point changes made to 73 STN links.

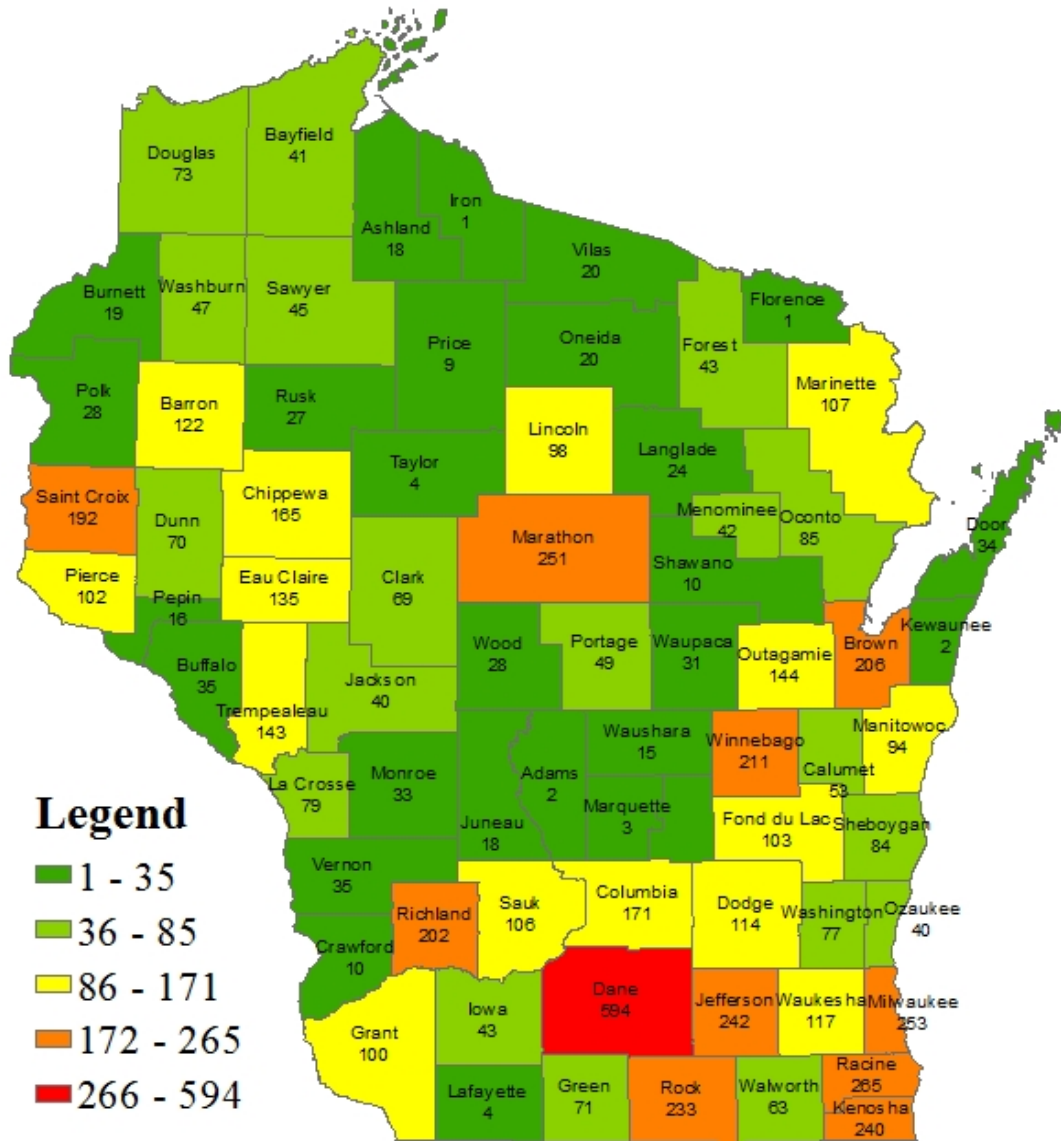


Figure 4.1. Thematic map of the number of data changes associated with each county.

To test the viability of the updated 2010 link-link table, Dane County crashes from 2010 were analyzed with the table. In 2010, there were 3784 crashes coded to an STN link in Dane County. Of these crashes, 3636 moved, through use of the updated link-link table, to a WISLR

link and offset. The 148 crashes that did not move to a WISLR link and offset were associated with 44 STN links. Forty of these links were links that had been made historic in the 2010 link-link table, thus the crashes associated with these 40 links could be moved to a WISLR link and offset using the older 2009 link-link table. The four remaining STN links containing crashes that did not move to a WISLR link and offset either were very historic links or links that do not exist in Dane County.

#### 4.3 Statewide Link-Link Coding Update Results

The 2010 update of the link-link table has been completed. To test the usage of the table, crash data from 2010 was analyzed using the table. In 2010, there were 50,166 crashes that occurred on state routes and contained an STN link and offset. Of these crashes, 50,162 (~100%) were translated to a WISLR link and offset using the updated link-link table. Four crashes did not move to WISLR because these four crashes were coded to a link that was never valid. That is, the link became valid and historic at the exact same moment. This link might have been created as a placeholder in the STN database or for another use by WisDOT. These results illustrate the usefulness of the updated link-link table.

#### 4.4 Data Translation Results

Data analysis regarding WISLR-to-STN data translation was performed on two Dane County data sets: crash data, hundredth-of-a-mile points, and on a statewide data set: the WisDOT RP database. There were 17,170 crash points in Dane County between 2005 and 2009 that moved from an originally coded STN link and offset to a corresponding WISLR link and offset. All of the crashes moved back from WISLR to STN, however, 17,919 crash points translated back to an STN link and offset, which is 647 crash locations more than the original

data set. The additional 647 (4%) crashes mapped back to multiple STN links. Of these 647 crashes, 28 (4%) were associated with turn-lane-flagged link-link records, 313 (48%) with median crossovers, and 2 (0.3%) with waysides. If median crossovers are ignored, 17,512 crash points translated to an STN link and offset; only 301 (2%) crashes mapped to multiple STN links. Additionally, all 17,170 crashes again mapped to a location thus no crash data was lost by ignoring median crossovers.

The second data set used for analysis was hundredth-of-a-mile points in Dane County. Points are placed on every STN link every hundredth of a mile, the points are moved to WISLR and visually inspected, and then moved back to STN and the initial location and final location are compared. There are 86,178 hundredth-of-a-mile points in Dane County along STN links. All of the points moved from STN to WISLR. When moving the points back to STN, 88,318 points moved, representing 1433 points (1.7%) that moved to multiple links. Of these 1433 points, 85 (6%) were associated with turn-lane-flagged link-link records, 260 (18%) with median crossovers, and 63 (4%) with waysides. When median crossovers were ignored, 87,958 points moved back to STN, representing 1149 points (1.3%) that moved to multiple links. Additionally, all 86,178 hundredth-of-a-mile points mapped to a location on STN; no data points were lost by not allowing data to land on median crossovers. The implementation of the rule required data placement at a particular location on STN that may or may not have been the original crash location. However, in implementing this rule, imprecise data placement would only be affected at median crossings.

An extensive analysis was done on the third and final data set: the WisDOT RP points database. This was chosen for extensive analysis because of the existence of the database as a statewide list that is a size that provides a substantial sample of conditions around the state. The

most recent statewide RP database consists of 64,131 points with STN links and offsets. Of these points, 37,562 moved to WISLR. This value consists of all but one current RPs and 3500 historic RPs. The historic RPs are not expected to all move to WISLR because some are located on historic STN links not included in the link-link table. When the initial data translation was performed, 44,123 point moved back to STN. All of the RP points on WISLR moved successfully back to STN, however 4355 (12%) moved to multiple links. Of these points, 1623 (32%) are associated with link-link records that are flagged as median crossovers, 1150 (26%) are associated with turn lanes, and 209 (5%) are associated with waysides. Additionally, of the 4355 points that mapped multiple times, 2560 (59%) moved to single unique locations multiple times. These 2560 points lie on top of each other and contain the same coordinates, but exist on different links, i.e. at the beginning of one link and the end of another link. Visual inspection of the problems associated with multiple-mapped points showed that the problems were representative of the problem categories that were previously identified. After the median crossover rule was implemented, 2102 (6%) RPs moved to multiple links. Again, no RP data was lost with the implementation of this rule. However due to resolution differences, even with implementation of this rule an RP point that could be placed at multiple locations may not necessarily move back to the original STN location of the RP.

Through analysis of each of these data sets, it was found that, in general, allowing a user to choose how to handle ambiguous data placement locations can reduce the number of multiples by up to 50% without losing any data. While only the median crossover rule was implemented in this analysis, it can be reasonably expected that the other rules presented in this research would have similar results in reducing multiples of translated data.

#### 4.5 WISLR Editing Scope Identification

WISLR links needing edits were identified with the methodology set forth in the previous section. Using this methodology, 4155 unused opposite links were identified, 288 reversed links were identified, and 80 links with length adjustments needed were identified. The full list of editing links was provided to WisDOT for further analysis.

#### 4.6 Conclusion

The results set forth in this section confirm the success of the methodology presented in this thesis. The link-link table was successfully updated to reflect the current WisDOT road network data. WISLR-to-STN data translation rules were formulated and tested on multiple data sets. Finally, WISLR links requiring edits were identified and reported to WisDOT.

## CHAPTER 5

### CONCLUSION AND FUTURE WORK

#### 5.1 Conclusion

The results of this research illustrate (1) the validity of the link-link table update procedure and the general approach to updating a functional merge between two Linear Referencing Systems (LRSs), (2) the necessity of data translation rules when translating data between two LRSs with differing resolutions, and (3) how the link-link table can be used to identify LRS links requiring edits. The link-link table is the functional merge between the State Trunk Network (STN) and the Wisconsin Information System for Local Roads (WISLR). These two LRSs were created by the Wisconsin Department of Transportation (WisDOT) to manage and analyze business data on state routes and local roads.

The major goal of this thesis was to update the link-link table to reflect changes made after original link-link coding. This was accomplished without disruption to normal WisDOT operating procedures. A second goal was to define rules regarding data translation between dissimilar-resolution LRS. This was accomplished by creating data collection pre-processing rules as well as data translation rules. Finally, a third goal was to identify links requiring edits by WisDOT to improve the accuracy of the link-link table. This was accomplished by grouping 4523 WISLR links in three editing type categories. The results show that all three of the stated goals of this research were successfully accomplished.

## 5.2 Future Work

The updating process presented in this research will need to be implemented on a yearly basis until link-link updating is incorporated into WisDOT operational procedures. This would allow for concurrent updating of the table with STN and WISLR updating. Concurrent updating would allow for the most up-to-date link-link table to exist at all times.

Although WISLR edits were identified in this research, more general updating is needed regarding problems associated with all of the flag columns in the link-link table. Editing issues relating to these flag columns - especially turn lanes, median crossovers, and waysides –would drastically improve the resolution of WISLR and limit the number of discrepancies between STN and WISLR. This resolution update is essential for the continued accuracy improvement of data translation between STN and WISLR without the need for translation rules.

The research performed for this thesis provided an effective methodology for updating the link-link table, however further research is recommended to perfect the functional relationship between WISLR and STN through the inclusion of the information provided by the link-link table into WisDOT operations. This thesis concludes with the expectation that the methodology previously presented can be implemented to accomplish this goal.

## REFERENCES

- Beyer, H. L. *Hawths analysis tools for ArcGIS* Retrieved 8/1/2012, 2012, from <http://www.spatial ecology.com/htools/index.php>
- Curtin, K. M., Nicoara, G., & Arifin, R. R. (2007). A comprehensive process for linear referencing. *Journal of the Urban and Regional Information Systems Association*, 12(2), 13-36.
- Dueker, K. J., & Butler, J. A. (2000). A geographic information system framework for transportation data sharing. *Transportation Research Part C: Emerging Technologies*, 8(1-6), 13-36. doi: 10.1016/S0968-090X(00)00006-1
- Graettinger, A. J., Qin, X., Spear, G., Parker, S. T., & Forde, S. (2009). Combining state route and local road linear referencing system information. *Journal of the Transportation Research Record*, 2121, 152-159.
- Graettinger, A. J., Qin, X., Spear, G., Parker, S. T., & Forde, S. (2008). State and non-state network mapping integration. *Proceedings of the 2008 Mid-Continent Transportation Research Forum*, Madison, Wisconsin.
- Miller, H. J., & Shaw, S. (2001). GIS-T data models. *Geographic Information Systems for Transportation: Principles and Applications*,
- Ryals, Z. T. (2011). *A technique for merging state and non-state linear referencing systems*. (Unpublished Master of Science degree). The University of Alabama, Tuscaloosa, Alabama.
- Scarponcini, P. (2002). Generalized model for linear referencing in transportation. *GeoInformatica*, 6(1), 35-55. doi: 10.1023/A:1013716130838
- Sester, M., Anders, K., & Walter, V. (1998). Linking objects of different spatial data sets by integration and aggregation. *GeoInformatica*, 2(4), 335-358.
- Vonderohe, A., Chou, C., Sun, F., & Adams, T. (1997). A generic data model for linear referencing systems. Paper presented at the *Research Results Digest 218. National Cooperative Highway Research Program. Transportation Research Board*, Washington, DC.



Vonderohe, A., & Hepworth, T. (1998). *A methodology for design of a linear referencing system for surface transportation, final report.* ( No. Project AT-4567).Sandia National Laboratories.

Wisconsin Department of Transportation. (1998). *Wisconsin location control management manual*

## APPENDIX A

### LINK-LINK DATE COLUMN POPULATION QUERIES

This appendix will present and describe the six link-link table date column population queries used in Microsoft Access and discussed in section 3.5.1 of this thesis. In general, mass population of dates should only be performed within the Start\_Valid and End\_Valid columns in the link-link table. Since dates contained within the Record\_Created and Record\_Historic columns reflect dates when a record was physically entered into the link-link table, these columns should only be populated by a link-link coder. The six date population queries are presented below and are named according to the column to which the query applies and the order in which each should be executed for each column: Start\_Valid\_Update1, Start\_Valid\_Update2, End\_Valid\_Update1, End\_Valid\_Update2, End\_Valid\_Update3, and End\_Valid\_Update4.

To execute all of these queries, three tables are required: the link-link table, the “refined” STN route-link table, and the “refined” WISLR route-link table. The latter two tables are briefly discussed in the body of this thesis and were generated by the TOPS Lab. For simplicity, the link-link table was renamed link\_link in the database, the STN route-link table was renamed STN\_LINK in the database while the WISLR route-link table was renamed WISLR\_LINK in the database. These names are subsequently used in the queries. Additionally, the queries are written for the 2010 link-link table update but can be easily adjusted for future years. All six queries also implement an inner join between the link\_link table, the STN\_LINK table, and the WISLR\_LINK table based on STN link ID and WISLR link ID, respectively.

The STN\_Valid\_Update1 query populates the Start\_Valid field in a record with the date route-link current date queried from the STN\_LINK table if the route-link current date associated with the STN link in the link-link record is more recent than the validity start date associated with the WISLR link in the link-link record. The WISLR date is queried from the WISLR\_LINK table and is the route-link current date associated with the WISLR link. The more recent validity date is used for population because unless *both* the STN and WISLR links in a link-link record are valid, then the record is not valid. The exact query language used in the Start\_Valid\_Update1 query can be seen in Figure A.1.

```
UPDATE WISLR_LINK INNER JOIN (link_link INNER JOIN STN_LINK ON
link_link.STNid=STN_LINK.RDWY_LINK_ID) ON
WISLR_LINK.RDWY_LINK_ID=link_link.WISLRid SET link_link.Start_Valid =
STN_LINK.DT_RTE_LINK_CURR
WHERE ((STN_LINK.DT_RTE_LINK_CURR)>WISLR_LINK.LCM_CURR_DT)
```

Figure A.1. Start\_Valid\_Update1 query language

The Start\_Valid\_Update2 query populates the Start\_Valid field in a record with the route-link current date queried from the WISLR\_LINK table if the route-link current date associated with the WISLR link in the link-link record is more recent than the validity start date associated with the STN link the link-link record. The STN date is queried from the STN\_LINK table and is the route-link current date associated with the STN link. The exact query language used in the Start\_Valid\_Update2 query can be seen in Figure A.2.

```
UPDATE WISLR_LINK INNER JOIN (link_link INNER JOIN STN_LINK ON
link_link.STNid=STN_LINK.RDWY_LINK_ID) ON
WISLR_LINK.RDWY_LINK_ID=link_link.WISLRid SET link_link.Start_Valid =
WISLR_LINK.LCM_CURR_DT
WHERE ((STN_LINK.DT_RTE_LINK_CURR)<WISLR_LINK.LCM_CURR_DT)
```

Figure A.2. Start\_Valid\_Update2 query language

The End\_Valid\_Update1 query populates the End\_Valid field in a record with the route-link historic date queried from the STN\_LINK table if and only if a route-link historic date associated with the WISLR link in the link-link record does not exist and the route-link historic date associated with the STN link is prior to January 1, 2011. Expressly, this means that the query populates the End\_Valid column with the route-link date associated with the STN link in the record, if the STN link is historic while the WISLR link is still current. The exact query language used in the End\_Valid\_Update1 query can be seen in Figure A.3.

```
UPDATE WISLR_LINK INNER JOIN (link_link INNER JOIN STN_LINK ON
link_link.STNid=STN_LINK.RDWY_LINK_ID)
ON WISLR_LINK.RDWY_LINK_ID=link_link.WISLRid
SET link_link.End_Valid = STN_LINK.LCM_DT_HSTL
WHERE ((WISLR_LINK.LCM_HSTL_DT Is Null) And (STN_LINK.LCM_DT_HSTL <
#1/01/2011#))
```

Figure A.3. End\_Valid\_Update1 query language

The End\_Valid\_Update2 query populates the End\_Valid field in a record with the route-link historic date queried from the WISLR\_LINK table if and only if a route-link historic date associated with the STN link in the link-link record does not exist and the route-link historic date associated with the WISLR link is prior to January 1, 2011. Expressly, this means that the query populates the End\_Valid column with the route-link date associated with the WISLR link in the record, if the WISLR link is historic while the STN link is still current. The exact query language used in the End\_Valid\_Update2 query can be seen in Figure A.4.

```
UPDATE WISLR_LINK INNER JOIN (link_link INNER JOIN STN_LINK ON
link_link.STNid=STN_LINK.RDWY_LINK_ID)
ON WISLR_LINK.RDWY_LINK_ID=link_link.WISLRid
SET link_link.End_Valid = WISLR_LINK.LCM_HSTL_DT
WHERE ((STN_LINK.LCM_DT_HSTL Is Null) And (WISLR_LINK.LCM_HSTL_DT <
#1/01/2011#))
```

Figure A.4. End\_Valid\_Update2 query language

The End\_Valid\_Update3 query populates the End\_Valid field in a record with the route-link historic date queried from the WISLR\_LINK table if route-link historic dates associated with the STN and WISLR links in the link-link record do, in fact, exist *and* the route-link historic date associated with the WISLR link in the record is prior to January 1, 2011 *and* the route-link historic date associated with the WISLR link in the link-link record is prior to the end validity date associated with the STN link in the link-link record. This query applies to records in which both the STN and WISLR links in the record are historic, and the WISLR link became historic in 2010, before the date when the STN link became historic. The exact query language used in the End\_Valid\_Update3 query can be seen in Figure A.5.

```
UPDATE WISLR_LINK INNER JOIN (link_link INNER JOIN STN_LINK ON
link_link.STNid=STN_LINK.RDWY_LINK_ID)
ON WISLR_LINK.RDWY_LINK_ID=link_link.WISLRid
SET link_link.End_Valid = WISLR_LINK.LCM_HSTL_DT
WHERE ((WISLR_LINK.LCM_HSTL_DT Is Not Null) And (STN_LINK.LCM_DT_HSTL Is
Not Null) And (WISLR_LINK.LCM_HSTL_DT < #1/01/2011#) And
(STN_LINK.LCM_DT_HSTL>WISLR_LINK.LCM_HSTL_DT))
```

Figure A.5. End\_Valid\_Update3 query language

The final query, the End\_Valid\_Update4 query, populates the End\_Valid field in a record with the route link historic date queried from the STN\_LINK table if route-link historic dates associated with the STN and WISLR links in the link-link record do, in fact, exist, *and* the route-link historic date associated with the STN link in the record is prior to January 1, 2011 *and* the route-link historic date associated with the STN link in the record is prior to the route-link historic date associated with the WISLR link in the record. This query applies to records in which both the STN and WISLR links in the record are historic, and the STN link became historic in 2010, before the date when the WISLR link became historic. The exact query language used in the End\_Valid\_Update4 query can be seen in Figure A.6.

```
UPDATE WISLR_LINK INNER JOIN (link_link INNER JOIN STN_LINK ON
link_link.STNid=STN_LINK.RDWY_LINK_ID)
ON WISLR_LINK.RDWY_LINK_ID=link_link.WISLRid
SET link_link.End_Valid = STN_LINK.LCM_DT_HSTL
WHERE ((WISLR_LINK.LCM_HSTL_DT Is Not Null) And (STN_LINK.LCM_DT_HSTL Is
Not Null) And (STN_LINK.LCM_DT_HSTL < #1/01/2011#) And
(STN_LINK.LCM_DT_HSTL<WISLR_LINK.LCM_HSTL_DT))
```

FigureA.6. End\_Valid Update4 query language

The queries shown in this appendix were used to populate the link-link table date columns systematically.

## APPENDIX B

### UPDATED WISLR POINT MOVING PROGRAM

```
Option Explicit On
Imports System.Data.OleDb
Imports System
Imports System.IO
Imports System.Math
Imports System.Text
Public Class Form1
    Inherits System.Windows.Forms.Form
    Private STN_Table, link_link_file As String
    Dim Unique_ID, Link_ID, Link_Offset
    Dim STNid, STNstart, STNend, WISLRid, WISLRstart, WISLREnd
    Dim Status As String

    Friend WithEvents Button5 As System.Windows.Forms.Button
    Const epsilon As Double = 0.000001

    Private Sub btnBrowse_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnBrowse.Click
        With OpenFileDialog1
            .Filter = "mdb Files (*.*)|*.mdb"
            .CheckFileExists = True
            .CheckPathExists = True
            .RestoreDirectory = True
            If .ShowDialog = DialogResult.OK Then
                txtfilepath.Text = .FileName
                link_link_file = txtfilepath.Text
            Else
                MsgBox("You didn't select your link_link file, or any file for that
matter.")
            End If
        End With
    End Sub
    Private Sub tableCheck(ByVal link_link_file As String)
        'This was to check the tables to see if there are any WISLR links the link_link
table that are not found in the WISLR points table,
        'but I don't think this will work because not every WISLR link is included in the
link_link table and not every WISLR link in the link_link table will have a point on it.
        Dim cn As New OleDbConnection("Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" &
link_link_file)
        Dim cmd2 As New OleDbCommand("DROP TABLE WISLR_Final", cn)
        cn.Open()
        If DoesTableExist("WISLR_final") = True Then
            Dim buttonYes As DialogResult
```

```

buttonYes = MsgBox("A table entitled 'WISLR_Final' already exists in the selected
database. Do you want to remove this table?", MsgBoxStyle.YesNo, "Table Exists")
    If buttonYes = Windows.Forms.DialogResult.Yes Then
        Dim buttonSure As DialogResult
        buttonSure = MsgBox("Are you sure?", MsgBoxStyle.YesNo, "Delete Table")
        If buttonSure = Windows.Forms.DialogResult.Yes Then
            cmd2.ExecuteNonQuery()
        Else
            Exit Sub
        End If
    Else
        Exit Sub
    End If
End If
cn.Close()
End Sub
Private Sub Read_Calculate(ByVal link_link_file As String)

    Dim rate, Unique_ID1, WISLR_Link, WISLR_Offset As Double
    Dim i As Integer = 0
    Dim currentrecord As String
    Dim STN_String As String

    Dim STNPointsDataTable As New DataTable
    Dim STNPointsDataRow As DataRow
    Dim LinkDataTable As New DataTable
    Dim LinkDataRow As DataRow
    Dim finaldatatable As New DataTable
    Dim finaldatarow As DataRow
    Dim UIColumn As DataColumn = New DataColumn("Unique_ID")
    UIColumn.DataType = System.Type.GetType("System.String")
    finaldatatable.Columns.Add(UIColumn)
    Dim LinkIDColumn As DataColumn = New DataColumn("Link_ID")
    LinkIDColumn.DataType = System.Type.GetType("System.String")
    finaldatatable.Columns.Add(LinkIDColumn)
    Dim OffsetColumn As DataColumn = New DataColumn("Link_Offset")
    OffsetColumn.DataType = System.Type.GetType("System.String")
    finaldatatable.Columns.Add(OffsetColumn)

    'Connects to, selects the records for, and fills the datatables to be used for
this part of the program
    Dim cn As OleDbConnection
    cn = New OleDbConnection("Provider=microsoft.jet.OLEDB.4.0;Data source= " &
link_link_file)

    STN_String = "SELECT * FROM STN_Points"
    currentrecord = "SELECT STNid,STNstart,STNend,WISLRid,WISLRstart,WISLRend,Status
FROM link_link"

    Dim STNadapter As New OleDb.OleDbDataAdapter(STN_String, cn)
    STNadapter.Fill(STNPointsDataTable)

    Dim rows As Integer = STNPointsDataTable.Rows.Count
    Dim linkadapter As New OleDb.OleDbDataAdapter(currentrecord, cn)
    linkadapter.Fill(LinkDataTable)
    Dim linkrows As Integer = LinkDataTable.Rows.Count

```



```

Dim criteria As String

Dim cn2 As OleDbConnection
cn2 = New OleDbConnection("Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" &
link_link_file)
Dim table_name As String
table_name = "WISLR_final"
Dim cmd As New OleDb.OleDbCommand("CREATE TABLE " & table_name & "(Unique_ID1
double, WISLR_Link double, WISLR_Offset double)", cn2)

cn2.Open()
Try
    cmd.ExecuteNonQuery()
Catch ex As OleDb.OleDbException
    MessageBox.Show(ex.Message, "OleDbException")
Exit Sub
Catch ex As Exception
    MessageBox.Show(ex.Message, "GeneralException")
Exit Sub
End Try
cn2.Close()

For Each STNPointsDataRow In STNPointsDataTable.Rows 'Cycles through the entire
set of STN_Points one at a time, selects matching records from
'the link_link table and runs calculations on them
Unique_ID = STNPointsDataRow(0)
Link_ID = STNPointsDataRow(1)
Link_Offset = STNPointsDataRow(2)

criteria = "STNid = " & Link_ID 'to select rows from the link_link table with
matching link_IDs

Dim selrows As Integer = LinkDataTable.Select(criteria).Length
If selrows = 0 Then 'Writes any records to a separate CSV file that have a
link_ID not in the link_link table

Else
    'Cycles through the selected set of records from the link_link table;
find which meets the criteria in the IF statement
For Each LinkDataRow In LinkDataTable.Select(criteria)
    STNid = LinkDataRow(0)
    STNstart = LinkDataRow(1)
    STNend = LinkDataRow(2)
    WISLRid = LinkDataRow(3)
    WISLRstart = LinkDataRow(4)
    WISLREnd = LinkDataRow(5)
    Status = LinkDataRow(6)

If Status = "C" Then
    'Performs math on any offset that falls within the correct range
If (Link_Offset = 0 And STNstart = 0) And (Status = "C") Then
    rate = (WISLREnd - WISLRstart) / (STNend - STNstart)
    WISLR_Offset = (rate * (Link_Offset - STNstart)) + WISLRstart

```

```

        WISLR_Link = WISLRid
        Unique_ID1 = Unique_ID
        Dim NewRow As DataRow = finaldatatable.NewRow() 'creates a
new row in the final datatable with the new values
        NewRow("Unique_ID") = Unique_ID1
        NewRow("Link_ID") = WISLR_Link
        NewRow("Link_Offset") = trunc(WISLR_Offset, 3)
        finaldatatable.Rows.Add(NewRow)
    ElseIf (Link_Offset > STNstart And Link_Offset <= STNend) And
(Status = "C") Then
        rate = (WISLREnd - WISLRstart) / (STNend - STNstart)
        WISLR_Offset = (rate * (Link_Offset - STNstart)) + WISLRstart
        WISLR_Link = WISLRid
        Unique_ID1 = Unique_ID
        Dim NewRow As DataRow = finaldatatable.NewRow() 'creates a
new row in the final datatable with the new values
        NewRow("Unique_ID") = Unique_ID1
        NewRow("Link_ID") = WISLR_Link
        NewRow("Link_Offset") = trunc(WISLR_Offset, 3)
        finaldatatable.Rows.Add(NewRow)
    End If
    ElseIf Status = "H" Then
    End If
Next
End If
i = i + 1
ProgressBar1.Value = (i / rows) * 100
Next

'Creates a Hashtable and an arraylist
'hashtable receives all values that are processed; if value exists already in the
hashtable, that row is placed in the arraylist
'the arraylist is then processed, and all records that are in the arraylist are
deleted from the finaldatatable
'Dim htable As New Hashtable
'Dim duplicatelist As New ArrayList

'For Each finaldatarow In finaldatatable.Rows
'    If htable.ContainsKey(finaldatarow(0)) Then
'        duplicatelist.Add(finaldatarow)
'    Else
'        htable.Add(finaldatarow(0), String.Empty)
'    End If
'Next

'For Each finaldatarow In duplicatelist
'    finaldatatable.Rows.Remove(finaldatarow)
'Next

i = 0
Dim cn3 As OleDbConnection
cn3 = New OleDbConnection("Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" &
link_link_file)

cn3.Open()
'Writes the values in the corrected finaldatatable to a CSV file

```

```

Dim final_ID, Final_Link, Final_Offset As String
Dim finalcount As Integer = finaldatatable.Rows.Count
For Each finaldatarow In finaldatatable.Rows
    final_ID = finaldatarow(0)
    Final_Link = finaldatarow(1)
    Final_Offset = finaldatarow(2)

    Dim sql As String = "INSERT INTO " & table_name & "(Unique_ID1, WISLR_Link,
WISLR_Offset) VALUES " & "(" & final_ID & " , " & Final_Link & " , " & Final_Offset &
""")

    Dim cmd1 As New OleDbCommand(sql, cn3)
    Dim records As Integer = cmd1.ExecuteNonQuery()
    i = i + 1
    ProgressBar3.Value = (i / finalcount) * 100
Next
cn3.Close()

End Sub
Public Function DoesTableExist(ByVal table_name As String) As Boolean
    Dim dbconn As New OleDbConnection("Provider=Microsoft.Jet.OLEDB.4.0;Data Source=
" & link_link_file)
    dbconn.Open()
    Dim restrictions(3) As String
    restrictions(2) = table_name
    Dim dbTable As DataTable = dbconn.GetSchema("Tables", restrictions)
    If dbTable.Rows.Count = 0 Then
        DoesTableExist = False
    Else
        DoesTableExist = True
    End If
    dbTable.Dispose()
    dbconn.Close()
    dbconn.Dispose()
End Function
Private Sub btnExecute_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnExecute.Click

    Me.Enabled = False
    tableCheck(link_link_file)
    Read_Calculate(link_link_file)

    MsgBox("Done. A table entitled WISLR_Final has been created to the Access
database.")

    Me.Close()

End Sub
Public Function trunc(ByVal number As Double, ByVal digits As Integer) As Double
    Return ((Truncate(number * Pow(10, digits))) / Pow(10, digits))
End Function

End Class

```