Project Title: **Wireless Data Collection Retrievals of Bridge Inspection/Management Information**

Research Reference Number: **MDOT # 2013-0067, Auth. No. 2 (R1, R2)**

Proposing Research Agency: **Michigan Technological University**
1400 Townsend Drive
Houghton, MI 49931

Principal Investigator(s): **PI: Colin N. Brooks, MEM**
Environmental Science Lab Manager, Research Scientist
Michigan Tech Research Institute (MTRI)
3600 Green Court, Suite 100, Ann Arbor, MI 48105
734-913-6858 ◆ cnbrooks@mtu.edu

Co-PI: Theresa (Tess) M. Ahlborn, Ph.D., P.E., FPCI, FACI
Professor, Civil and Environmental Engineering
Director, Center for Structural Durability
1400 Townsend Drive, Houghton MI 49931
906-487-2625 ◆ tess@mtu.edu

Contracting Authority: **Lisa Jukkala**
Manager, Government Contracts/Training, Sponsored Programs
Michigan Technological University
1400 Townsend Drive, Houghton, MI 49931
906-487-2226 ◆ 906-487-2245 fax ◆ lajukkal@mtu.edu

Revised February 28, 2017

1. Report No.	2. Government Accession No.	3. MDOT Project Manager
RC-1634	N/A	Rich Kathrens

4. Title and Subtitle	5. Report Date
Wireless Data Collection Retrievals of Bridge Inspection/Management Information	2/28/2017
	6. Performing Organization Code 38-6005955
7. Author(s) C. Brooks, R. Sawtell, G. Sullivan, R. Dobson, T. Ahlborn, N. Jessee, H. Kourous-Harrigan, S. Aden	**8. Performing Organization Report No.**
9. Performing Organization Name and Address Michigan Technological University 1400 Townsend Drive Houghton, MI 49931	**10. Work Unit No. (TRAIS)** N/A
	11. Contract No. 2013-0067
	11(a). Authorization No. No. 2 (R1, R2)
12. Sponsoring Agency Name and Address Michigan Department of Transportation Research Administration Section 425 West Ottawa Street Lansing, MI 48933	**13. Type of Report and Period Covered** Final Report (Revised), including Phase II through 2/28/2017
	14. Sponsoring Agency Code N/A

15. Supplementary Notes
This final report includes the content of the first report delivered Sept. 30, 2015, with updates to reflect Phase II work completed through Feb. 28, 2017.

16. Abstract
To increase the efficiency and reliability of bridge inspections, MDOT contracted to have a 3D-model-based data entry application for mobile tablets developed to aid inspectors in the field. The 3D Bridge App is a mobile software tool designed to facilitate bridge inspection processes by enabling inspectors to enter element-level bridge condition data using 3G/4G network-enabled tablet devices. The system collects information from MDOT's bridge management database, and then renders a dynamic, interactive 3D model of the desired bridge. The bridge inspector is able to record the locations and attributes of new defects in an element-level form by touch interaction and manipulation of the 3D model. The interactive model, marked up with existing defects, also allows for bridge inspectors to better visualize past inspection data. The inspector can also take pictures of the defects using the tablet's camera, as well as record comments. This gives users further insight into the progression of the defect over time. The bridge inspector is able to navigate along the bridge model just as he or she would during a normal inspection. Results can be integrated into bridge management workflows.

17. Key Words	18. Distribution Statement
Bridges, tablets, inspection, element, 3D, database, data, interactive, Unreal Engine 4	No restrictions. This document is available to the public through the Michigan Department of Transportation.

19. Security Classification - report	20. Security Classification - page	21. No. of Pages	22. Price
Unclassified	Unclassified	101	

Acknowledgments

The project team would like to thank and acknowledge the MDOT program manager, Rich Kathrens, the research manager, Michael Townley, and the members of the research advisory committee for their advice and oversight during this project. The help of Michigan Department of Technology, Management and Budget staff in detailing database integration requirements was also appreciated. The PIs would like to thank their project team of Reid Sawtell (lead developer), Glenn Sullivan, Helen Kourous-Harrigan, Rick Dobson, Sam Aden, and Nate Jessee for their dedication to helping MDOT rapidly, safely, and effectively collect element-level bridge condition data through this new tool. The help of MDOT staff, especially bridge inspectors such as Janiene Devinney, and private companies such as Great Lakes Engineering Group in increasing our understanding of real-world inspection processes was critical to the success of the project. We thank everyone who advised the project and met with our team.

Disclaimer

Table of Contents

List of Figures

List of Tables

Executive Summary

Current bridge inspection practices at the Michigan Department of Transportation (MDOT) utilize paper forms followed by a manual data entry step to populate the Bridge Management System (BMS) database with information needed for bridge management and repair. Faced with an aging bridge inventory and increasing federal regulations regarding collection of element-level data, MDOT wishes to increase the efficiency and reliability of collected data. To achieve this, MDOT requested a 2D/3D application that can utilize mobile tablet technology to aid inspectors in the field.

To develop this application, a Michigan Technological University applied research team, led by staff from the Michigan Tech Research Institute (MTRI), first examined the state of practice across the nation to better understand currently available options. They found that as of 2014, no application assisted with collection of element-level data. Next, MTRI met with experienced bridge inspectors (from the consulting firm Great Lakes Engineering Group as well as MDOT staff inspectors) to better understand the needs of bridge inspectors so the application design could be tailored to their input.

Because MDOT does not have 3D bridge models available for all bridges, MTRI developed a server application using Django (a Python web framework) to generate Extensible Markup Language (XML) files using data from MDOT's BMS database. Each XML file provides a generic bridge model that is sufficiently representative for inspection purposes; it contains information about the element-level components of a bridge, including location and size. The server application includes a user tuning component to correct initial erroneous assumptions due to lack of information, such as placement of bearings per beam.

To produce the client application, MTRI selected the Unreal Engine 4 (UE4) game engine by Epic Games to provide cross-platform rendering capability. The application itself is built using C++ interfaced with the UE4 engine, as well as UE4 Blueprints for high-level functionality. It uses Java for integration with native camera functionality on Android devices, and Objective-C for iOS devices. The client application receives a XML file from the server application and constructs an interactive 3D model. Using a set of intuitive navigational views, the inspector can traverse the bridge and mark the surface of the model with element-level defect information, photos, and comments. Defect markers are proportionally sized based on the defect quantity and are color-coded to match condition states. The application also has a summary view for reviewing the aggregate defect information and for editing National Bridge Inventory (NBI) ratings.

The project's second phase focused on further development to bring the application closer to implementation. MDOT-requested enhancements included import/export XML functionality to enable integration of inspection results with MDOT's BMS database, NBI reporting functionality, and element transparency. A potential third phase would focus on moving the app into day-to-day usage by MDOT, with the potential to bring the tool into national usage by working with the American Association of State Highway and Transportation Officials (AASHTO) to integrate it into AASHTOWare. Recommendations included in the Implementation Action Plan for a potential third phase include fully integrating the app with MDOT's BMS database, updating the app with key features suggested during user testing, enabling the app to support a wider set of bridges, and moving into the deployment phase so that MDOT can start using the tool as part of its standard inspection procedures.

1. Introduction

Collecting bridge inspection data is a key component of assessing bridge condition and managing MDOT's infrastructure. Regulations issued by the Federal Highway Administration require states to use a data-driven process to check the completeness and accuracy of bridge data and to verify compliance with the National Bridge Inspection Standards. States are also required to collect and maintain element-level inspection data as prescribed by the American Association of State Highway and Transportation Officials (AASHTO), a provision that increases the time and complexity of the inspection process.

Current inspection practices have inspectors using paper forms in the field to collect condition-state information and to provide historical reference data. These data must then be entered manually into the Michigan Bridge Inspection System (MBIS) and Michigan Bridge Reporting System (MBRS) (now both part of MiBRIDGE), which adds yet another task to the process and introduces potential for error. Photographs documenting bridge deterioration must be taken and stored as well, which requires additional documentation to be generated linking individual photographs with the locations they were taken. Finally, inspectors must carry relevant reference materials to verify the accuracy of the data they are collecting. Together, these demands burden inspectors with a growing load of devices and physical information that they must manage, often in unfavorable or hazardous conditions.

Given these issues, MDOT wishes to increase the efficiency and accuracy of the data collection process. Since mobile computing and wireless data transfer are now ubiquitous, these technologies offer a promising alternative to the current paper solution. Tablet devices are relatively inexpensive, can be made ruggedized for outdoor use or come ruggedized, can communicate directly with MDOT online services, and typically include cameras with acceptable resolution. A digital inspection process can leverage all of these features to streamline data entry, rapidly collect more detailed inspection information, and reduce the physical inventory needed by inspectors.

1.1 Objectives

This project had the following objectives:

1. Review and evaluate ongoing and recently completed research involving the bridge inspection process.
2. Review MDOT's process of collecting National Bridge Inventory (NBI) and AASHTO Element Level inspection data.

3. Develop an application to collect NBI and Element Level inspection data using visual methods and 2D drawings or 3D models of the bridge elements.
4. Develop and test a wireless data collection and display system to meet MDOT's bridge inspection and management needs which can be integrated with MDOT's existing web applications and database structure. Determine alternatives that will work on multiple mobile platforms.

1.2 Scope

To realize the overall project goal of developing an application that improves accuracy and efficiency of MDOT's bridge inspection process, the following 10 tasks were performed (Tasks 1-6 were part of Phase I, and Tasks 7-10 were added with Phase II):

- Task 1: Literature Review Document
- Task 2: Web/tablet application integrated with MDOT's current MBIS and MBRS Systems (now known as MiBRIDGE).
- Task 3: Field demonstration of application
- Task 4: Application User's Manual
- Task 5: Complete documentation of the application and source code
- Task 6: Final Report
- Task 7: Integrate System With MDOT Database
- Task 8: Finalize Cross Platform Support
- Task 9: Finalize 3D Model User Tuning
- Task 10: Add Support for collecting NBI Ratings

Task 1 was needed to evaluate what options currently exist. Determining how bridge inspections are carried out nationwide helped shape the application's features so it will meet or exceed MDOT's needs.

Task 2 included the development of the application itself and occurred throughout the project time frame. Task 3 was imperative for garnering feedback from inspectors and ensuring that the system was usable and successful. As Task 2 proceeded, Task 3 was executed from the first prototype of the application through the conclusion of the project.

Similarly, Task 4 was ongoing throughout the project lifetime (including Phase II) to reflect the evolving functionality of the application.

Task 5 provided a smooth transfer of the application from the research development team to MDOT ownership.

Tasks 7 to 10 were part of a supplemental development plan following the initial project to enhance the application's functionality and bring the application closer to release and integration with MDOT's inspection routine.

2. Literature Review

While federal guidelines for bridge inspection reporting must be met nationwide, individual states are free to meet those requirements in different ways. This has led to the use of diverse methodologies and a host of commercial solutions addressing the states' needs. The literature review for this project determined the state of the practice for bridge inspections across the country and summarized the tools currently available to facilitate the process, including devices that could be used to deploy a mobile bridge inspection application. Unfortunately, at the time of the project's literature review in 2014, none of these solutions, mobile or otherwise, were capable of handling AASHTO element-level data collection. The full state-of-the-practice report generated for this project is contained in Appendix 8.2.

In addition to evaluating current software solutions, the project team developed a survey to assess the methodologies used by bridge managers throughout the nation (Figures 2-1 and 2-2). Twenty-one responses were received from 21 states. This survey concluded that over 70 percent of the responding states used some electronic hardware in the data collection process, and over half of that hardware was laptops. Many agencies used custom software for inspection and management, including in-house software and modified or customized commercial solutions. See Appendix 8.3 for further details on the survey results.

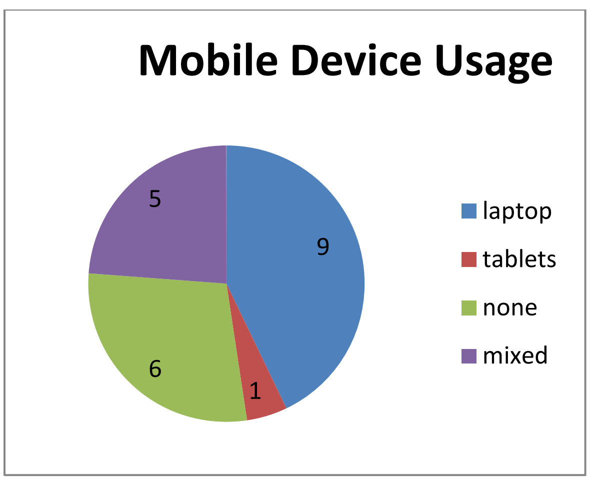


Figure 2-1: Mobile device usage of the responding agencies. The "mixed" category includes agencies that use both tablets and laptops.

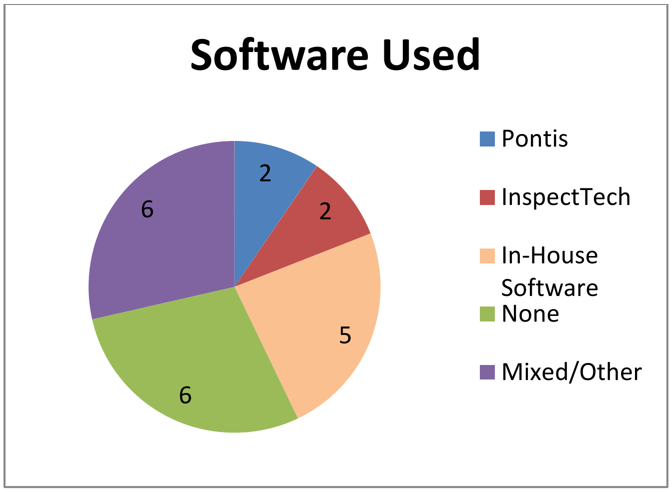


Figure 2-2: Types of inspection and management software currently being used. Many agencies use custom software. The "mixed/other" category represents modified or customized commercial solutions.

3. Review of MDOT Practices

The successful design and implementation of an application for collecting MDOT bridge inspection data hinged on understanding the current practices of MDOT bridge inspectors. By understanding the challenges and procedures inspectors deal with, the project team could develop an application with the functionality needed to help MDOT improve efficiency and accuracy. Project staff met with MDOT staff, including bridge inspectors, on several occasions to learn about and document current practices.

3.1 Inspection Forms

At the core of the inspection process are the forms that define what data must be collected to complete a bridge inspection. These forms include the NBI Safety Inspection Report, NBI CoRE Elements Report, and the Structure Inventory and Appraisal (SI&A) form.

The SI&A form (Figure 3-1) largely serves as a reference for the bridge being inspected by providing information such as component material types, dimensions, load ratings, and inspection frequency. It also contains a few fields for overall ratings of structure components such as superstructure, substructure, deck, and paint.

Inspection Data		
90 - Inspection Date	05/07/2013	
91 - Inspection Freq	24	
92A - Frac Crit Req/Freq	N	
93A - Frac Crit Insp Date		
92B - Und Water Req/Freq	N	
93B - Und Water Insp Date		
92C - Oth Spec Insp Req/Freq	N	
93C - Oth Spec Insp Date		
92D - Fatigue Req/Freq	N	
93D - Fatigue Insp Date		
176A - Und Water Insp Method	0	
58 - Deck Rating	6	
58A/B - Deck Surface/Bottom	7	6
59 - Superstructure Rating	7	
59A - Paint Rating	6	
60 - Substructure Rating	6	
61 - Channel Rating	6	
62 - Culvert Rating	N	
Navigation Data		
38 - Navigation Control	0	
39 - Vertical Clearance	0	
40 - Horizontal Clearance	0	
111 - Pier Protection		
116 - Lift Brdg Vert Clear		

114 - Future ADT	9000
115 - Year Future ADT	2028
Freeway	0
Structure Appraisal	
36A - Bridge Railing	1
36B - Rail Transition	1
36C - Approach Rail	1
36D - Rail Termination	1
67 - Structure Evaluation	6
68 - Deck Geometry	8
69 - Underclearance	N
71 - Waterway Adequacy	8
72 - Approach Alignment	8
103 - Temporary Structure	
113 - Scour Criticality	3
Miscellaneous	
37 - Historical Significance	5
98A - Border Bridge State	
98B - Border Bridge %	
101 - Parallel Structure	R
EPA ID	MIK812263424
Stay in Place Forms	
143 - Pin & Hanger Code	4
148 - No. of Pin & Hangers	12

56 - Left Horiz Clearance	0		
100 - STRAHNET			
102 - Traffic Direct			
109 - Truck %			
110 - Truck Network			
114 - Future ADT			
115 - Year Future ADT			
Freeway			
Proposed Improvements			
75 - Type of Work			
76 - Length of Improvement			
94 - Bridge Cost			
95 - Roadway Cost			
96 - Total Cost			
97 - Year of Cost Estimate			
Load Rating and Posting			
31 - Design Load	9		
41 - Open, Posted, Closed	A		
63 - Fed Oper Rtg Method	6		
64F - Fed Oper Rtg Load	2.47		
64MA - Mich Oper Rtg Method	6		
64MB - Mich Oper Rtg	1.64		
64MC - Mich Oper Truck	18		
65 - Inv Rtg Method	6		
66 - Inventory Load	1.48		
70 - Posting	5		
141 - Posted Loading			
193 - Overload Class	A		N

Figure 3-1: A section of the Structure Inventory and Appraisal form.

The NBI Safety Inspection Report contains the bulk of what the inspector must collect. It is organized first by overarching categories such as Deck, Superstructure, and

Substructure. Each of these categories is then broken into subcategories, such as Stringer, Paint, Section Loss, and Bearings (for Superstructures). The inspector must assign each subcategory a 0 to 9 condition rating that factors in all of the deterioration or flaws present in those components throughout the bridge. To aid the inspector's decision, a history of ratings for previous years is included, as well as past comments. When the report is completed, the combination of current and historical inspection information gives an overall picture of the progress and rate of bridge deterioration (see Figure 3-2 for an example).

SUPERSTRUCTURE				
	05/09	05/11	05/13	
9. Stringer (SIA-59)	7	7	7	Painted A588 steel I beams with staggered diaphragms. Stainless steel pins. Few areas of light LOS less than 10% near beam ends, cleaned and painted. (05/13) Painted A588 steel I beams with staggered diaphragms. Stainless steel pins. Few areas of light LOS less than 10% near beam ends, cleaned and painted. (05/11) Painted A588 steel beams and diaphragms. Tight vertical cracks in backwalls. Surface coat applied to backwalls. Staggered diaphragms. Stainless steel pins. (05/09)
10. Paint (SIA-59A)	8	8	6	Painted A588 steel I beams. Minor rust on few top flanges at leaching deck cracks. (05/13) Painted A588 steel I beams. (05/11) Painted A588. (05/09)
11. Section Loss	2	2	2	Few areas of light LOS 10% or less near beam ends, cleaned and painted. (05/13) Few areas of light LOS 10% or less near beam ends, cleaned and painted. (05/11) Few areas of light LOS 10% or less near beam ends, cleaned and painted. (05/09)
12. Bearings	7	7	7	Bearings cleaned and painted. Minor rust on few abutment bearings. (05/13) Bearings cleaned and painted. (05/11) Rockers and abutment bearings have been cleaned and painted. (05/09)

Figure 3-2: A section of the NBI Safety Inspection Report. The report combines historical and current ratings and comments to fully document deterioration.

The NBI CoRe Elements Report captures AASHTO element-level information on condition state. Each component of the bridge is assigned an element type number (there are approximately 158). For a given bridge, applicable element types have a total quantity and a unit of measurement (linear, area, or both). When inspectors look at a bridge, they must quantify the units and condition states of defects for each element type for the whole bridge. The condition states are Good, Fair, Poor, and Severe. To aid in the inspection process, each element type has a table listing the possible defects that can be associated with it and descriptions of the defect for each condition state (see Table 3-1 and Figure 3-3).

Table 3-1: Condition State Table for Prestressed Concrete (from the Michigan Bridge Element Inspection Manual).

Michigan Bridge Element Inspection Manual

CS TABLE 2 – PRESTRESSED CONCRETE

Defects	Condition State 1	Condition State 2	Condition State 3	Condition State 4
	GOOD	FAIR	POOR	SEVERE
Spalls/ Delaminations/ Patch Areas (1080)	None.	Delaminated. Spall 1 in. or less deep or less than 6 in. diameter. Patched area is sound.	Spall greater than 1 in. deep or greater than 6 in. diameter. Patched area is unsound or showing distress. Does not warrant structural review.	The condition warrants a structural review to determine the effect on strength or serviceability of the element or bridge; OR a structural review has been completed and the defects impact strength or serviceability of the element or bridge.
Exposed Rebar (1090)	None.	Present without section loss.	Present with section loss that does not warrant structural review.	
Exposed Prestressing (1100)	None.	Present without section loss.	Present with section loss that does not warrant structural review.	
Cracking (1) - PSC (1110)	Insignificant cracks or moderate-width cracks that have been sealed.	Unsealed moderate-width cracks or unsealed moderate pattern (map) cracking.	Wide cracks or heavy pattern (map) cracking.	
Efflorescence / Rust Staining (1120)	None.	Surface white without build-up or leaching without rust staining.	Heavy build-up with rust staining.	
Settlement - Substructure (4000)	None.	Exists within tolerable limits or arrested with effective actions taken to mitigate.	Exceeds tolerable limits but does not warrant structural review.	
Scour - Substructure (6000)	None.	Exists within tolerable limits or arrested with effective countermeasures.	Exceeds tolerable limits but is less than the limits determined by scour evaluation, and does not warrant structural review.	
Damage (7000)	Not applicable.	The element has minor damage caused by vehicular or vessel impact.	The element has moderate damage caused by vehicular or vessel impact.	The element has severe damage caused by vehicular or vessel impact.

AASHTO ELEMENTS (English Units)

Element Number	Element Name	Total Quantity	Unit	Good CS1	Fair CS2	Poor CS3	Severe CS4
Decks/Slabs							
803	Conc Deck - Coated Bars	8262	sq.ft	0	8226	36	0
				0%	100%	0%	0%
815	Rigid Overlay	8262	sq.ft	8240	0	22	0
				100%	0%	0%	0%
811	Conc Deck - Btm Surface	8262	sq.ft	0	8226	36	0
				0%	100%	0%	0%
812	Reinf Conc Fascia	360	ft	270	90	0	0
				75%	25%	0%	0%

Figure 3-3: A section of the NBI CoRe Elements Report. Deterioration is classified by element type, quantity, and condition state.

3.2 Inspection Procedures

While the inspection forms determine which data need to be collected, of equal importance is how those data are collected. There are no rigid rules that define how a bridge inspector should go about collecting the necessary information to fill out the forms, so there is a natural variability in how individuals and organizations will handle the process. However, guidelines and the physical nature of the task ensure that there should be sufficient overlap in practices to define a generalized procedure. Capturing this process was essential to the design of the inspection application since it directly reflects the needs of the application's users, who are in turn trying to meet the needs of bridge managers.

MTRI staff began by meeting with Amy Trahey, president of Great Lakes Engineering Group, LLC, and a former MDOT bridge inspector. Trahey provided a virtual walk-through of a bridge inspection. (Figure 3-4 represents the inspection process as Trahey described it.) The process is nonlinear—inspectors do not simply go down the list of items on the form and evaluate each one. This is largely a matter of efficiency. For example, evaluating the railings on a bridge requires walking both sides of the bridge, and in doing so the inspector will pass many other components. Trahey also provided a listing of tools and materials an inspector would require during the inspection, such as manuals, ratings guides, cameras, previous inspection reports, and pencils. Another important consideration is that inspections are routinely performed by two inspectors. Typically, one inspector will proceed with the inspection itself, filling out the forms, while the other inspector will photograph bridge deterioration and areas of concern.

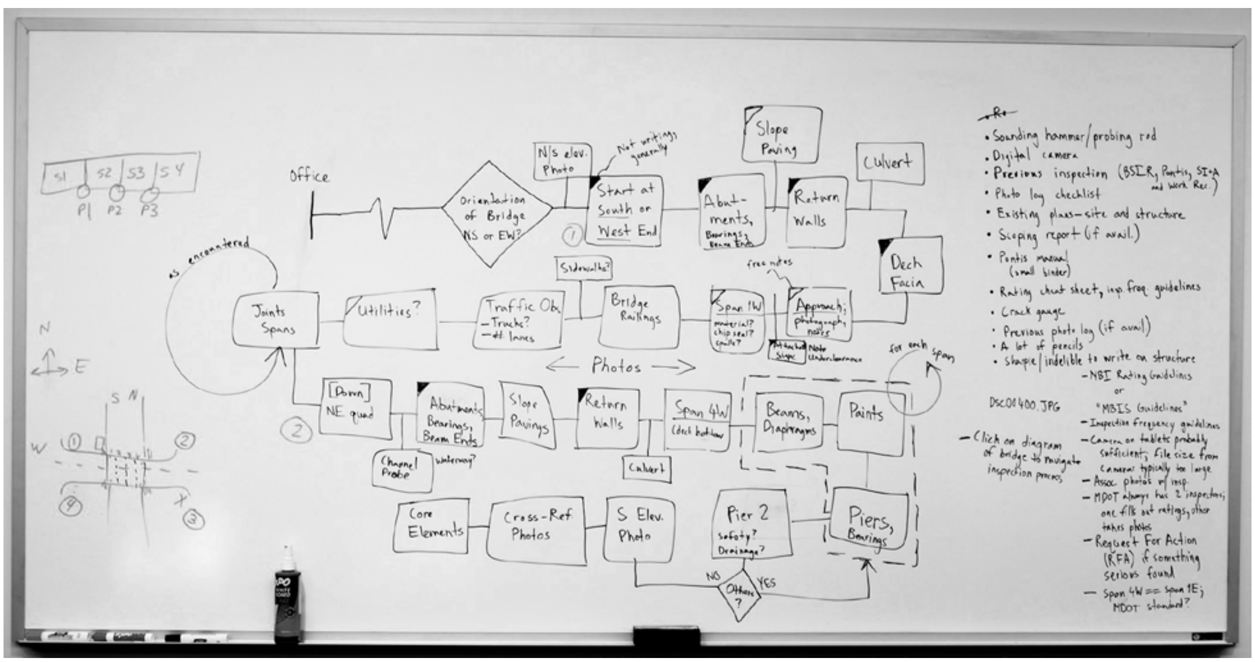


Figure 3-4: Inspection flow diagram and tool/material listing.

To supplement their understanding of the bridge inspection process, MTRI staff accompanied MDOT inspectors Janiene DeVinney and Lindsey Renner for a mock inspection of the Curtis Road Bridge over M-14 northeast of Ann Arbor (Figure 3-5). This served as a very useful demonstration of the workflow process outlined by Amy Trahey and gave the application developers a chance to see firsthand what a bridge inspector deals with. Of particular note, inspectors write a great deal of information on scratch paper or in the margins of the paper forms, since the generalized ratings are formed from a comprehensive view of the bridge while the inspection process itself must proceed piecemeal. The group also discussed office practices, because inspectors must transfer information from paper forms to MDOT's database after the inspection is completed. They also discussed task assignment authentication/security practices since inspectors are responsible for the quality of their inspections.

Figure 3-5: MTRI staff observe an MDOT inspector examining joint condition.

4. Application Design and Requirements

Using the information gained from the literature review and from observing MDOT's current practices, the project team formulated requirements and design parameters for the application. The software requirements specification is designed to encapsulate what the application will and will not do. Its primary purpose is to ensure clear communication between the client and the developer concerning the application's functionality. It is not meant to be a rigid constraint; it can be revised as needed given further clear communication between parties. The original document can be seen in Appendix 8.4.

4.1 Requirements

The primary requirement of the application was that it collect and aggregate AASHTO element-level inspection data. It was MDOT's desire that this would involve a 2D or 3D interface (preferably 3D) depicting the bridge elements, which could then be tagged with relevant information such as element type, defect type, condition state, and defect quantity. Such an application would have the advantage of not only capturing element-level data, but also capturing the location and size of individual defects, which opens up new opportunities for monitoring deterioration. This primary requirement was of keen importance since at the time of the literature review, no software or procedure existed to efficiently gather element-level data.

Of secondary importance was the collection of comments and photographs concerning the defects, preferably utilizing a device's built-in camera. This information, along with the element-level data, is vital to maintaining a historical record of the bridge's condition so appropriate deterioration monitoring can occur and response decisions can be made. Additionally, it was desired that the application automatically compile the recorded information into the broader categories used in the various forms, thereby eliminating the need for inspectors to keep track of it themselves. Compiling individual defects also would dovetail well with the inspectors' practice of recording information as it is observed.

MDOT was also interested in viewing historical information during the inspection process. This feature would be similar to the previous ratings available on the NBI Safety Inspection Report, which provide additional input for the inspector to consider. Finally, since the application would already be in a digital format, it should be designed to enable communication with the MDOT BMS database to store finalized inspection data and photographs, eliminating the need for inspectors to do so manually.

4.2 Design Considerations

In developing the application, several important design decisions had to be considered. The first of these was device compatibility, since a wide range of portable electronic devices are available, including laptops, tablets, and smartphones. MDOT was primarily interested in tablet devices as a good compromise between the bulk and power of a laptop and the portability but small screen size of a smartphone. However, the tablet operating system (OS) universe is quite diverse, and different options are often incompatible with one another: Any application developed natively for one device would need to be completely reprogrammed to work on another OS. Web applications are promising in that they run via browsers instead of natively, but they require an active Internet connection. This may not be available in rural areas, rendering the application useless. Additionally, the desire for either 2D or 3D interaction is not well-suited to a Web application, primarily for performance reasons. Fortunately, MTRI was able to identify an alternative development strategy that sidesteps these challenges: software packages used to design video games for multiple mobile platforms.

Game design software, referred to as game engines, are software packages used by game developers to create interactive applications. They can be either 2D or 3D, and many of them promise cross-platform compatibility. With such packages, the task of device interoperability falls to the engine creators rather than the individual developers. The MDOT application is not a game, but it does share many common elements with video games, such as a need for 2D/3D rendering, geometry modeling, touch-based interaction and Web access. MTRI investigated a variety of available engine platforms to select one as the foundation for the MDOT bridge inspection application. From the large pool of available platforms, MTRI narrowed the list to three for final consideration, detailed below in Table 4-1.

Table 4-1: Game Engine Comparison.

Library	License/Cost	Pros	Cons
OSG	Based on Lesser General Public License (LGPL), a free software license	Free, low-level access, open-source code	Small community, poor documentation/support, low cross-platform compatibility (must develop natively)
Unity	$3,000 per developer	Very large community, good support, game industry standard, feature-rich, great performance	Expensive, must purchase licenses per developer, must purchase per additional platform supported, closed-source code
Unreal Engine 4	Initially $19/month for MTRI (unlimited seats, and now free), plus 5% of revenue if selling on market under standard license	Cheaper than Unity, large community, feature-rich, cutting-edge development, source code available	Early in product life cycle (software bugs, low support initially for some features), 5% of revenue if selling on market

Based on the low cost, list of features and promise of cross-platform support, MTRI chose to proceed with application development using Unreal Engine 4 by Epic Games. While being on the cutting edge of development is always a risk, Epic has a long history of successful development (Unreal Engine 3 is widely used even today). Additionally, Unreal Engine 4 subscribers are granted access to the source code of the engine, a huge advantage in shaping the application to MDOT's needs and ensuring that MDOT and MTRI will always have access to the platform for future development. As an added bonus, Epic entirely dropped the monthly subscription fee in March 2015, so MTRI and MDOT were able to receive software updates at no charge during the remainder of the development period.

5. Server Implementation

Since previous 3D models of the state's bridges were not consistently available, a model had to be created from scratch. Given the large amounts of descriptive information within MDOT's Bridge Management System database, MTRI decided to build a model utilizing all of the relevant data. from the database This way, any bridge being inspected could be viewed with a sufficiently representative model. The data were retrieved from the database, missing information was derived from the data collected, and then a representative model was created as an XML file (See Figure 5-1). When requested, the XML file is then sent to the client application to render the 3D model.

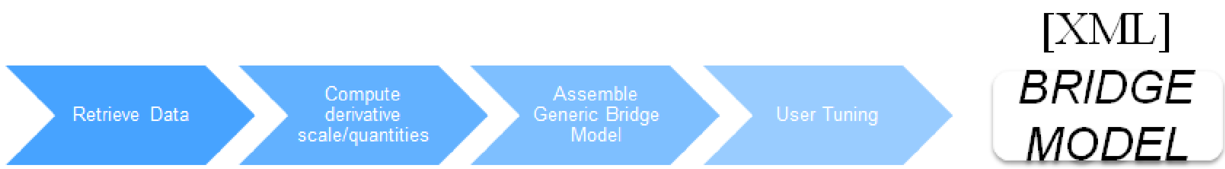


Figure 5-1: Bridge model generation.

5.1 Review of Bridge Fundamentals

To gain a better understanding of bridges, the MTRI team met with Tess Ahlborn, Co-PI and Michigan Technological University Professor of Civil and Environmental Engineering. Ahlborn gave a two-hour lecture on basic bridge fundamentals and addressed any of the staff's questions or misunderstandings about bridges. During the lecture, Ahlborn covered how a generic bridge works, explaining the function of the deck, superstructure, and substructure. The lecture also covered more specific bridge parts (such as pin and hanger assemblies, bearings, diaphragms, and girders) to provide further details about the basic components of a bridge. Ahlborn concluded the lecture by explaining all of the bridge elements that composed the Curtis Road Bridge over M-14 near Ann Arbor, which MTRI has been using as a test bridge for development (since the time MTRI staff visited it for the mock inspection). This in-depth explanation of bridges was instrumental in the project's development, as it provided further insight into how bridges work and fit together, allowing the programmers to better understand the process of making generic 3D bridge models.

5.2 BMS Database

The first step in building the 3D model was the retrieval of data from MDOT's BMS database. The database is composed of 16 tables. These tables were not intended to be used to generate 3D models, but they contain a wealth of information including bridge dimensions, bridge measurements, bridge form data and AASHTO element-level data. After copying the database onto MTRI's local server for development and testing, MTRI added one additional table to the database that would store all of the information needed to create a proportionally accurate representation of the bridge. This Bridge Model table draws from almost all of the other tables within the database and incorporates several new fields that MTRI, using generic assumptions about bridge construction, derived from the information in the database. To simplify the XML generation process, the application only pulls data from this new Bridge Model table. The Bridge Model table is very large, simplifies the process of exporting database information into an XML format, it also means that individual bridges can be modified without making changes to the rest of the BMS database.

Additionally, the BMS database contains a wealth of ancillary information such as sidewalk dimensions, traffic flow information and presence of water beneath the bridge The application uses some of this information to collect the most recent element and NBI report information, though there is other information that has not been utilized yet due to other tasks being prioritized to improve the functionality of the application first. However, this information lends itself to future improvements of the application that could make the model even more realistic.

5.3 Computing a Generic Bridge Model

After the Bridge Model table is created, MTRI utilizes Django, an open-source Python Web framework for managing websites while incorporating large amounts of data from databases (https://www.djangoproject.com/). This server application will output the requested XML file for the desired bridge when contacted by the client application. To generate the XML file, the server will query the appropriate information from the Bridge Model table, derive necessary quantities from the queried data, convert all variables to the appropriate units, and generate a list of member components (See Figure 5-2). The data needed for the NBI report information is shown in Figure 5-3. The client connects to the server using HTTP over a Wi-Fi or cellular connection. This Internet connection will be necessary for the application to load the appropriate Bridge Model XML file, but after the initial download of the XML file, no further Internet connection is necessary as the file can be stored on the tablet device.

The initial generation of the Bridge Model utilizes a set of assumptions to create values for variables that cannot be derived from the database, such as placement of bearings per pier, number of beams, and joint locations. These derived quantities should work for the majority of bridges, and all the necessary information to render the 3D model will be within the XML file. If these assumptions result in an erroneous model, administrative users can tune them to improve model fidelity (discussed in section 5.5).

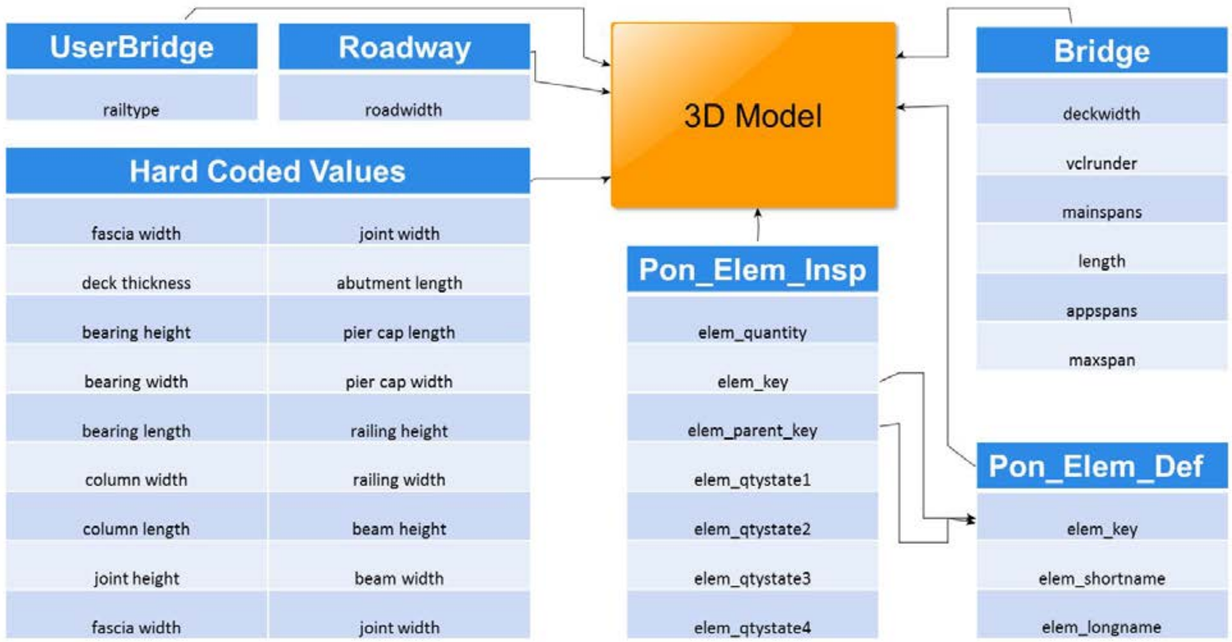


Figure 5-2: Flowchart of the back-end obtaining all of the data necessary to create the 3D model.

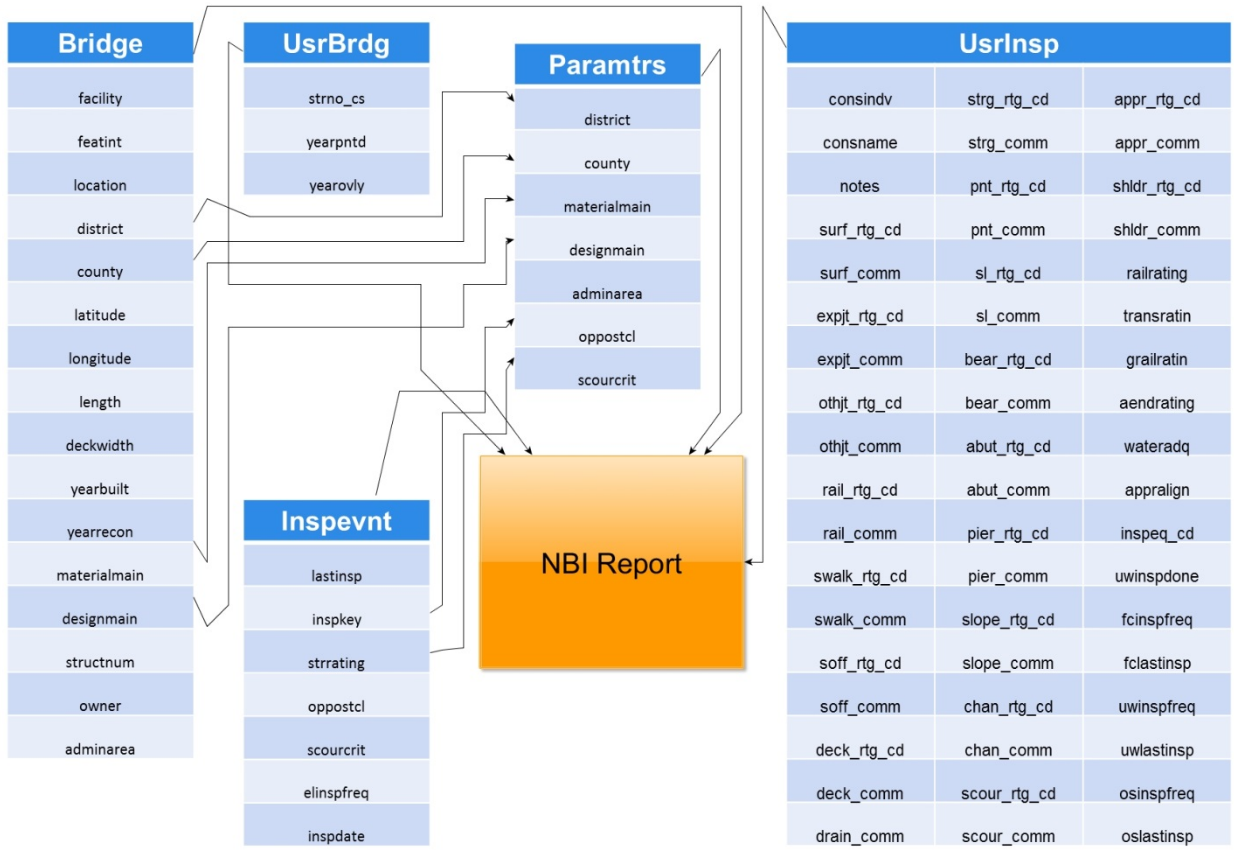


Figure 5-3: Flowchart of BMS data integrated into the NBI Safety Inspection Report.

5.4 XML File Structure

The entire XML file is arranged into six categories: basic bridge information, deck, superstructure, substructure, bearings, and culvert. Other than basic bridge information, all of the categories are created using AASHTO element-level data from the BMS database, which gives very specific details about all of the bridge parts that compose that bridge. The individual pieces of the bridge that will be rendered as parts of the 3D model are represented by the term Member in the XML file.

Each Member contains data for the Role, Type, Name, Length, Width, Height, X-coordinate, Y-coordinate, Z-coordinate, and the AASHTO Element Number associated with that Member. The Role is the category an individual member falls into, the Type is the exact name specified by the AASHTO element-level data within the database, and the Name is the identifier associated with the standard bridge inspection labeling schemes for elements such as 2 South or 1 West 2 South. The labeling scheme changes per element, and also depends on the bridge orientation, such as whether the bridge runs north and south or east and west. Each Member is associated with one label, so if the deck bottom surface is labeled as 1 South 3 West, that will be an individual bridge piece that will be rendered separately from 1 South 2 West. When rendered, the bridge parts will appear seamless, as if they were one bridge part, but they actually are multiple pieces that make up the entire deck bottom surface. The Length, Width, and Height are all values derived from the database to render a proportionally accurate representation of the bridge. The only information in the database relevant to member height is the vertical clearance of the bridge. All of the element Heights below the substructure (pier, pier cap, and abutments) are inferred, using fixed height for most elements and extending the pier and abutment heights to cover the remaining distance. Other dimensions are also inferred if they are not found in the database. The X-, Y-, and Z-coordinates are based on the Length, Width, and Height of the individual element as well as its location relative to the other components to get an exact centeral location for that element. The AASHTO Element Number is provided so the client application can determine the context of the member. (See Figure 5-4 for examples of the above data contained within the bridge XML file.)

```
- <Member>
      <role>Deck</role>
      <type>Concrete Deck - Coated Bars</type>
      <name>2S</name>
      <length>1451.98234368</length>
      <width>491.47385216</width>
      <height>15.0</height>
      <AASHTO_Element_803>803</AASHTO_Element_803>
      <x>1229.9850432</x>
      <y>265.73692608</y>
      <z>270.5133888</z>
```

Figure 5-4: Example of a bridge member variable in XML format.

Using Member variables to represent individual bridge pieces is critical since the unavailability in the database of some of the required information imposes certain limitations on creating a 3D model from the database. The Member variables are self-defining (they do not rely on relative information from any other part of the XML) so the client is more flexible for future model improvements. This will be helpful in the case of more unusual bridges such as those that have a varying number of beams per span, or those where the bearing placement per pier is abnormal.

5.5 User Tuning

As noted, MTRI made some generic assumptions in calculations for key variables used to render the 3D bridge model. To address the issue, an administrative website (separate from the client application) was developed through Django that enables the inspector or bridge engineer to verify and/or modify these assumptions to create a more accurate model. For the generic concrete overpass-style bridge, the calculations should be reasonably accurate. However, there are several outliers where key pieces of information about how the bridge is composed—such as numbers of beams per span and placement of bearings per pier—are abnormal. These bridges would be modeled incorrectly and therefore the inspector could not record defect data accurately. The website's administration tool allows bridge inspectors to fix any errors in the model (usually ahead of the inspection) to create a better replica of the bridge, and allows them to make any necessary changes to the data as they see fit. Within the administration tool, the information is divided into eight categories: Assumptions, General Bridge Information, Deck, Superstructure, Substructure, Bearings, Bearing Placement, and Culvert. The most important information that the bridge inspector will need to review are the Assumptions and Bearing Placement sections. These are the two areas where data are not present in MDOT's database but are derived from calculations and assumptions. In the future, more fields and categories may be incorporated in the administrative tools to make a more accurate model.

5.6 Other Services

The application requests different URLs for past NBI CoRe Element and NBI Inspection reports. Each URL sends back an XML file with the most recent report information for the bridge that was selected. Additionally, the server can accept newly collected NBI data to store to the database. When the user finishes an inspection, he or she can press the "Push" button, and the front-end application will send all of the element-level defect and report information in an XML file to the back-end. The back-end will then appropriately store the data in the correct variables to use in the future.

5.7 Limitations

As models do not exist for every bridge potentially needing inspection, MTRI needed to use information from MDOT's BMS database to create each 3D model. The current application is optimized to accurately model generic overpass-style bridges but will inaccurately model bridges that are irregular. This limitation is ameliorated using the Django administration site, which can correct many simple errors in the models. Another limitation is that the application does not yet handle "exotic" bridges such as cable bridges, culvert bridges, or truss bridges. These bridges will not cause the application to crash or behave improperly, but they will not be rendered properly in the current version. This limitation could be addressed through a future enhancement-focused project phase. Such bridges are not particularly common, and modeling them would be time-consuming; time was instead spent on higher-priority tasks during the project's first two phases. A final key limitation is that bridges that are not monitored by MDOT are particularly challenging to model properly, as no AASHTO element-level data have been captured for them. The application's model for these bridges would be limited by a lack of structural information and would be unlikely to represent the bridge accurately.

6. Client Implementation

Implementation of the client application, whose name has been changed from MDOT 3D Wireless Bridge Inspection System/3DWBIS to 3D Bridge App, is the primary product of this research. It is built on Epic Games' Unreal Engine 4 (UE4) and can work both in Windows desktop environments and on Android mobile devices such as tablets or smartphones. Taking advantage of UE4's rendering capabilities, the 3D Bridge App parses XML files delivered by the server and creates 3D representations of the bridge being inspected. Then, inspectors can dynamically tag the surface of the bridge with defects.

6.1 Coding

UE4 is primarily based on the C++ programming language using the Microsoft Visual Studio development environment. The engine relies heavily on macro functionality, adding its own particular flair of coding as well as an extensive application program interface (API) for interfacing with the engine. Any software development projects utilizing the engine include an Unreal-specific build program that automatically sets up the Visual Studio environment and pre-compiles specialized header files that prepare the macro interface. There is also a UE4 plug-in for Visual Studio that allows tighter integration with UE4 projects. The bulk of the new application is coded in this environment, but there are several important exceptions.

The first is UE4's Blueprint language (see Figure 6-1). This is essentially a visual coding language defined within the UE4 editor that allows for high-level interaction with game mechanics. This higher abstraction level, as compared to coding in C++, benefits certain tasks such as user interaction with objects and camera navigation. Functions, operators, events, and variables exist in Blueprint as blocks on the screen with inputs and outputs as tie-in points on the blocks. Different code blocks are then strung together, linking like variables across blocks as well as tying the execution flows together to form the program.

The second exception is native device coding. This is done within the UE4 source code rather than project code and is specific to the operating system targeted. In this case, use of the built-in cameras available on mobile devices must be developed separately for iOS and Android. For example, Android's native language is Java, so the camera functionality exists as a Java plug-in for UE4. While it is inconvenient to have to reproduce this functionality for each supported operating system, the extra effort needed is rather small compared to developing the entire application for multiple systems.

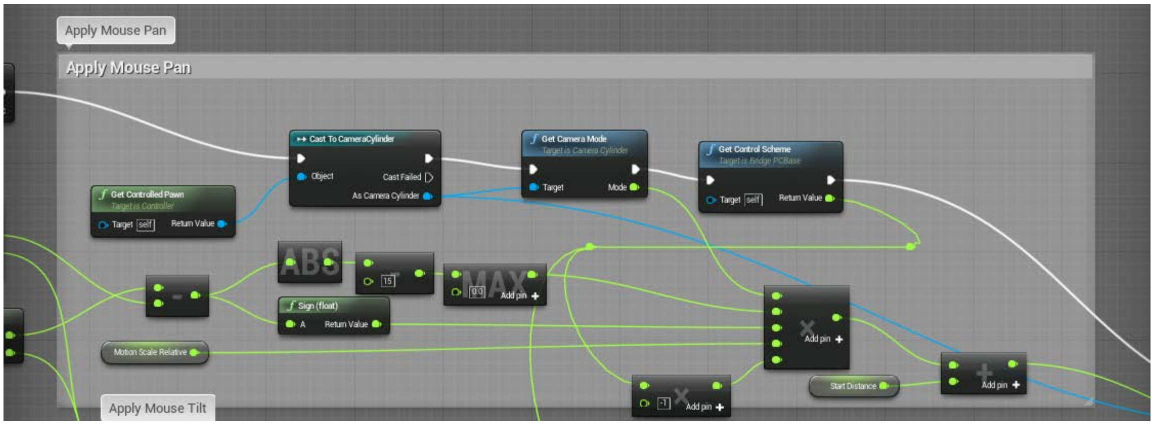


Figure 6-1: UE4's Blueprint coding language as used to implement the Client Application's user interface.

6.2 Loading Bridge XML files

The first step in using the application is to load the XML model for the bridge being inspected. The sidebar menu of the application includes a Load Bridge button, which polls the server for a list of bridge models available (see Figure 6-2). The user then selects the bridge of interest and can either load it or download it. The Download option copies the XML to the device's internal storage for offline use; such bridges will have their menu item display in green instead of blue. The Load option will use the downloaded XML if available or, if not, will ask the server for the XML instead. While in offline mode, only bridges with downloaded XML files will appear in the list. Once the server has responded, the application will parse the XML and generate a list of all the bridge member elements. Each member element is assigned appropriately scaled and positioned geometry within the application world, effectively constructing the bridge from its individual components. Each of these elements retains context-sensitive information about itself, such as the member's name,

Figure 6-2: Load Bridge menu.

that is displayed when the user interacts with the element.

6.3 Navigation

Navigation in a full 3D environment can be daunting since it involves motion with six degrees of freedom (6-DoF), three-axis translation and three-axis rotation. This problem is exaggerated in touch-based environments, which are limited to a 2D plane. Multi-touch, gestures where more than one finger is used, can help, but overreliance makes the user experience unintuitive. For the client application, multi-touch is limited to the familiar pinching gesture often used for zoom. Since this limits the application to 3-DoF input for a 6-DoF environment, some constraint on allowable motion is needed. To cope with this problem, two viewing methods have been implemented to allow for natural viewing of the bridge geometry while keeping user interaction simple and intuitive.

The first viewing method has been dubbed Camera Cylinder (see Figure 6-3). Essentially, the camera, or view angle of the user, is constrained to a cylindrical orbit along the bridge. Swiping left or right with mouse or touch interaction pans the view, while swiping vertically changes the orbit angle of the camera around the bridge. Since a full 360-degree orbit of the bridge would result in the camera viewing the bridge upside down, or, if the camera were flipped, would cause a control inversion that would be frustrating and confusing for users, viewing is limited to 180-degree arcs. However, the compass widget in the upper right of the application heads-up display (HUD) can be clicked to switch to the opposite arc. The final pinching gesture allows the camera to zoom in on a target area of interest to the inspector. The Camera Cylinder viewing mode is the default and allows the inspector to intuitively navigate most of the bridge, while the zoom option makes it easy to get close-in views.

The second viewing method is called Camera Rail (see Figure 6-4) and was created in response to feedback from MDOT bridge inspectors during a demonstration of the application. In this view, the camera is constrained to a box volume centered on the bridge. Vertical and horizontal swipes pan the camera in their respective directions, while the pinch gesture translates the camera forward or backward along the bridge. The compass widget switches the camera view direction 180 degrees. This viewing method is convenient for reproducing some of the viewing angles inspectors use in the field, such as looking at an abutment head-on.

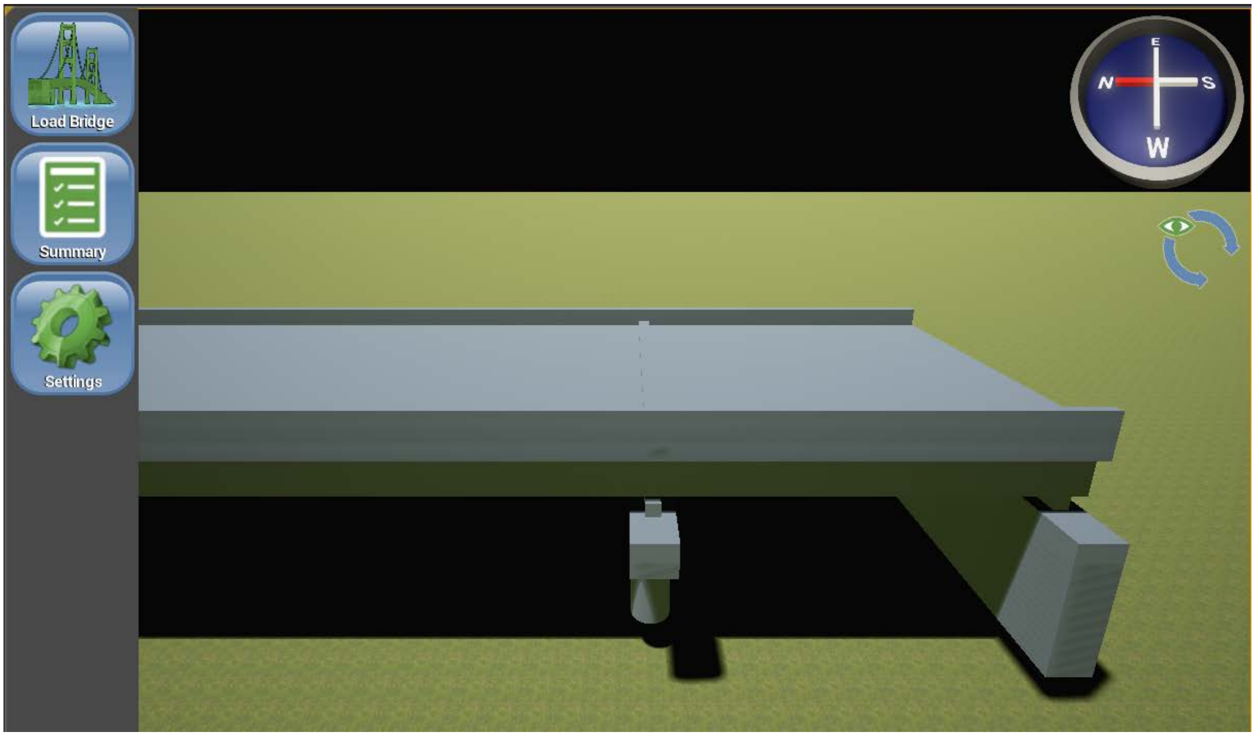


Figure 6-3: Camera Cylinder view orbits around and along the bridge.

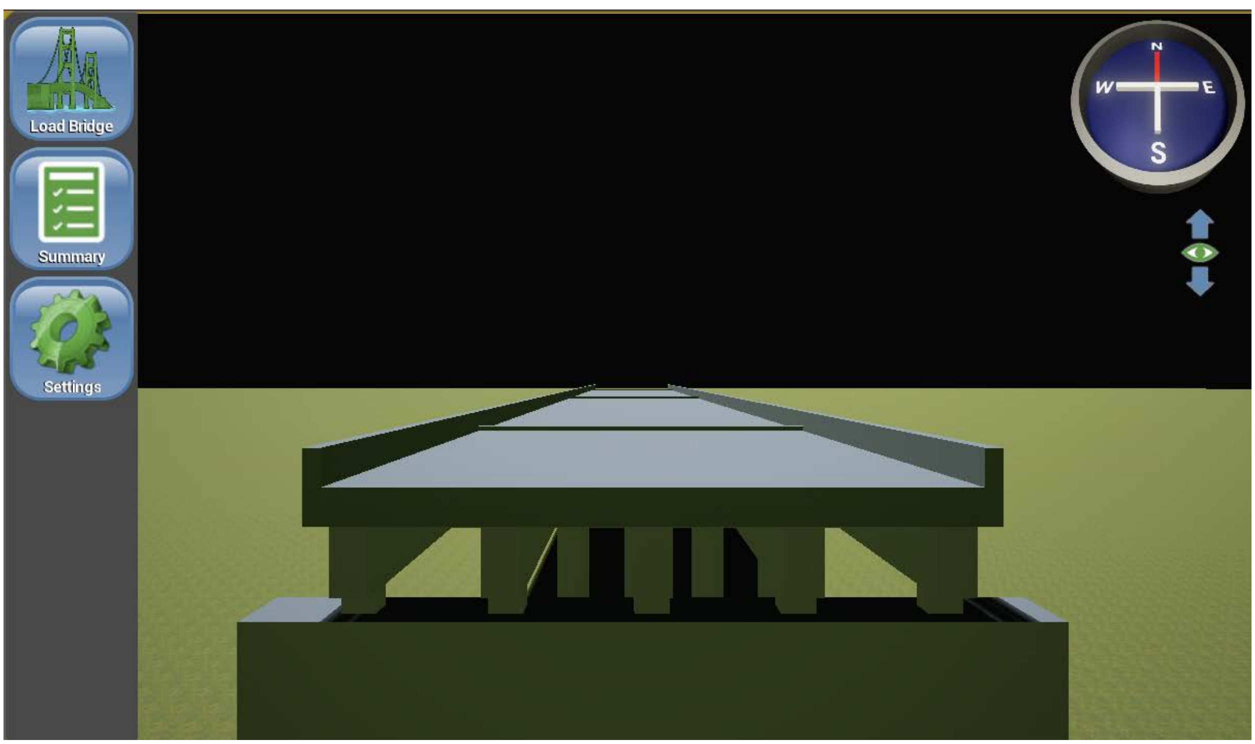


Figure 6-4: Camera Rail view allows head-on inspection of the bridge.

6.4 Element-Level Defects

The primary feature of the application is its ability to tag the bridge model with defects. After navigating to the bridge location being examined, the inspector can tap on the bridge surface to place a defect marker (see Figure 6-5). A menu pops up that allows the inspector to select an element type from a shortlist of elements most likely applicable based on context-sensitive information from the bridge XML file. A check box exists to disable the filtering and present the full list of elements should the inspector not find the one being examined. Once an element type has been selected, the inspector chooses the defect type. The defect type drop-down menu is populated only with types applicable to the selected element type. The inspector can also choose the condition state of the defect (the default state is Fair) and enter the unit quantity for the defect. The defect description is automatically updated according to the combination of defect type and condition state, allowing the inspector to quickly confirm that the option selected matches MDOT guidelines. The Add Picture button allows inspectors to attach an existing photograph or take a new one; clicking on an attached photo will display a full-screen image of the photo. At the bottom is a comment box where inspectors can add any additional information they wish to record.

Figure 6-5: Defect pop-up menu. Title and element shortlist are context-sensitive according to the bridge location touched.

Also part of the defect pop-up menu is the option to switch to the Edit Marker mode; this view removes the HUD and pop-up overlays to offer an unrestricted view of the bridge (see Figure 6-6). A minimal interface at the bottom presents the user with options to resize (according to unit quantity), relocate, and rotate the defect marker. The user also can manipulate the aspect ratio of the marker, allowing for an infinite variety of rectangular markers. Setting the aspect ratio to 0 will convert the marker to circular from rectangular.

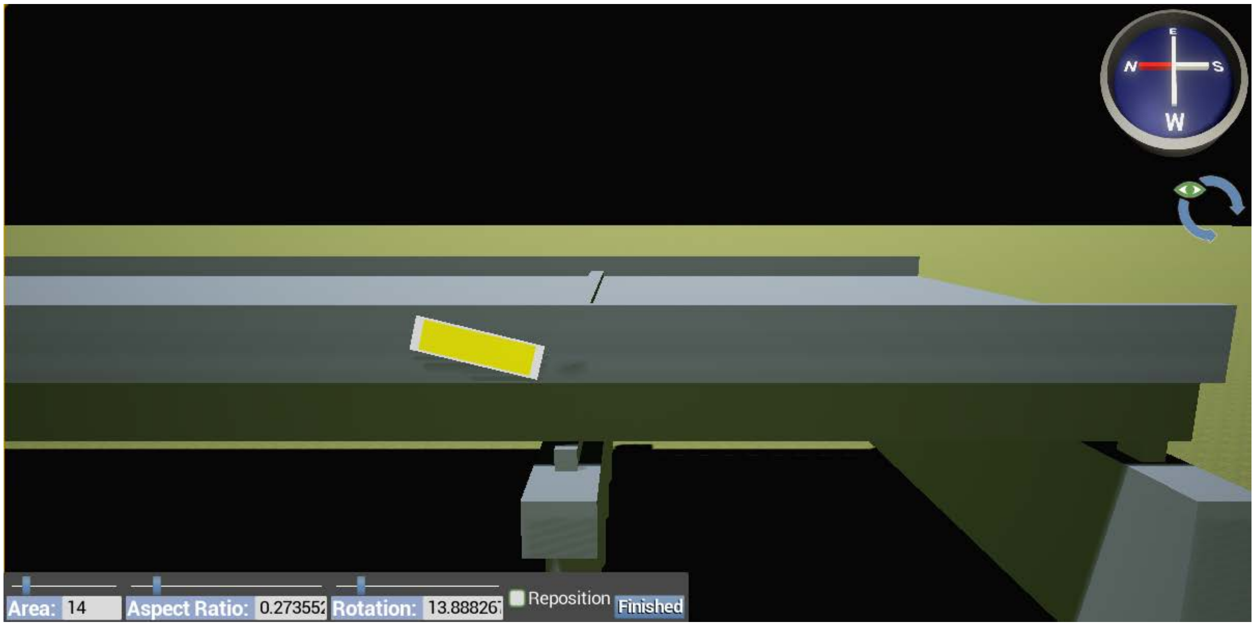
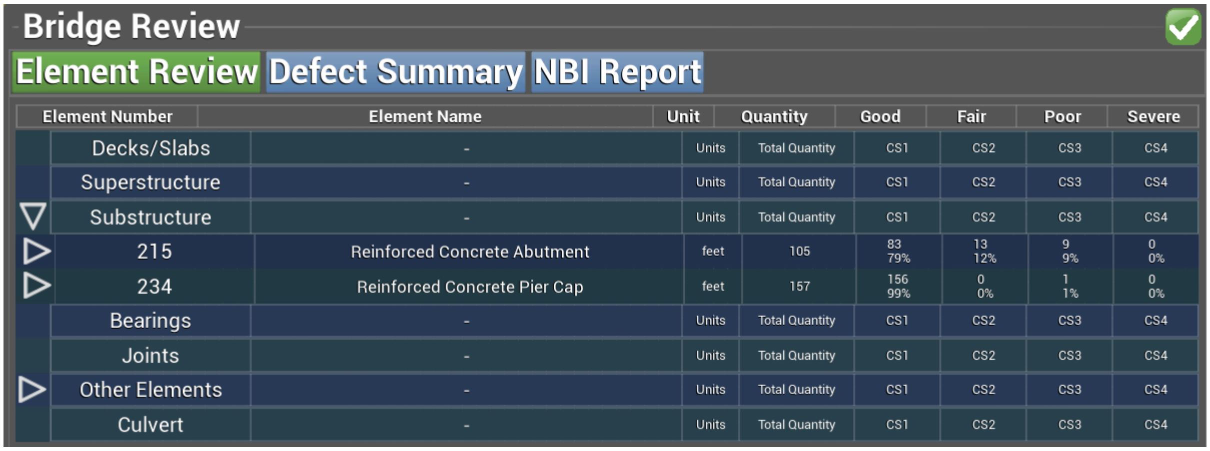


Figure 6-6: Marker Editor offers an unrestricted view so the inspector can position and manipulate the defect.

A button on the bottom left of the defect menu links the defect to the NBI rating entry menu, through which the inspector can pull up the NBI section most relevant to the current defect (see Section 6.6).

6.5 Bridge Review

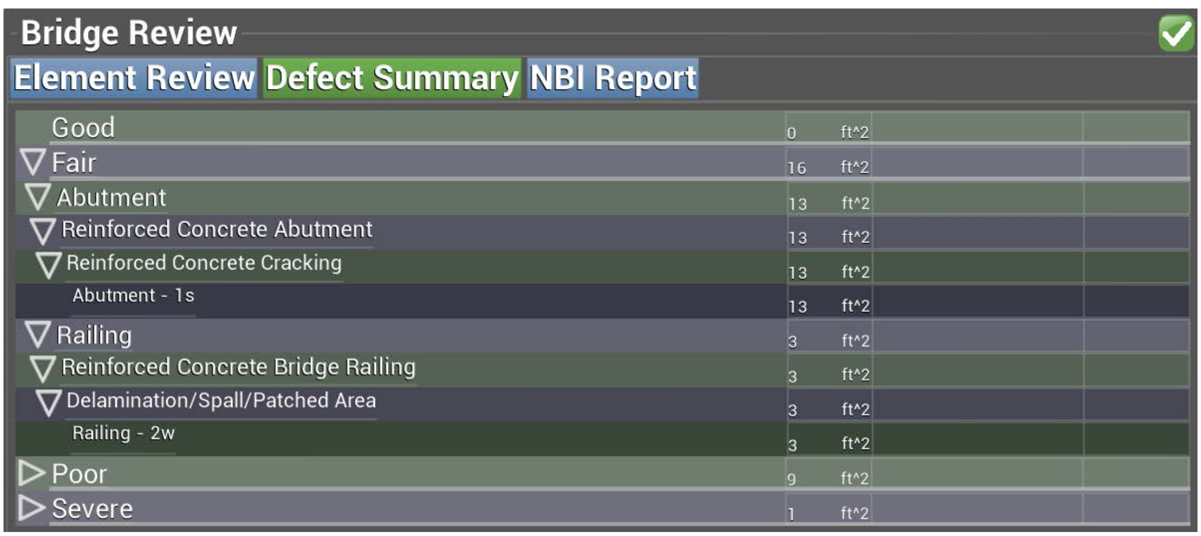
The Bridge Review menu offers several choices for reviewing the data collected during the inspection process. The Element Review mimics the format available on the MiBRIDGE website (see Figure 6-7), listing the percentage of condition states for each bridge element but also providing a breakdown of the individual defects contributing to that score. Totals are updated as the inspection continues, relieving inspectors of having to perform the calculations themselves.



Element Number	Element Name	Unit	Quantity	Good	Fair	Poor	Severe
Decks/Slabs	-	Units	Total Quantity	CS1	CS2	CS3	CS4
Superstructure	-	Units	Total Quantity	CS1	CS2	CS3	CS4
Substructure	-	Units	Total Quantity	CS1	CS2	CS3	CS4
215	Reinforced Concrete Abutment	feet	105	83 79%	13 12%	9 9%	0 0%
234	Reinforced Concrete Pier Cap	feet	157	156 99%	0 0%	1 1%	0 0%
Bearings	-	Units	Total Quantity	CS1	CS2	CS3	CS4
Joints	-	Units	Total Quantity	CS1	CS2	CS3	CS4
Other Elements	-	Units	Total Quantity	CS1	CS2	CS3	CS4
Culvert	-	Units	Total Quantity	CS1	CS2	CS3	CS4

Figure 6-7: Element Review mimics MiBRIDGE format.

The Defect Summary menu offers an alternative breakdown of the defects (see Figure 6-8). The top level of the drill-down shows the condition rating, while subsequent levels show the category, then element type, defect type, and finally individual defects. Quantities are automatically summed for each level of the drill-down, and comment boxes and icons for photographs are available.



Good	0	ft^2
Fair	16	ft^2
Abutment	13	ft^2
Reinforced Concrete Abutment	13	ft^2
Reinforced Concrete Cracking	13	ft^2
Abutment - 1s	13	ft^2
Railing	3	ft^2
Reinforced Concrete Bridge Railing	3	ft^2
Delamination/Spall/Patched Area	3	ft^2
Railing - 2w	3	ft^2
Poor	9	ft^2
Severe	1	ft^2

Figure 6-8: Defect Summary drill-down to individual element-level defects.

6.6 NBI Safety Inspection Report

As part of the supplemental Phase II work plan, the project team added the capability of entering and reviewing NBI safety inspection report information to create a more integrated solution to bridge inspections. The full NBI safety inspection report information can be accessed through the Bridge Review menu. The display mimics the paper form but includes a few appropriate upgrades for a digital format (see Figure 6-9).

The top section of the display is identical to the paper form, listing bridge information such as location, dimensions, materials, last inspection date, and current inspector, and providing an entry box for general inspection notes. Below that, the NBI rating entries are found, divided into structural categories (such as deck, superstructure, etc.) The categories are collapsible, facilitating navigation between sections on limited screen space. The final section, Miscellaneous, contains data entry fields for all applicable items including guardrail ratings, water adequacy, approach alignment, high-load hits, and underwater inspection method.

Figure 6-9: Digital NBI Report form.

General NBI sections pertaining to the bridge structure and approach all follow the same entry format and can be accessed from the full report form or by clicking the NBI Ratings shortcut button in any bridge defect menu (see Figure 6-10). The shortcut menu option will infer which NBI category the inspector is interested in reviewing based on the current defect context, but any category can be selected from the drop-down menu. This context-sensitive shortcut system allows inspectors to move quickly between entering detailed element-level information and entering information in the broad NBI categories, facilitating an enter-as-you-go approach.

At the top of the shortcut form, the previous three ratings are listed along with a button to enter in the current rating. Below that, the previous three comments are listed, each one accompanied by a button that will copy that comment into the current comment box at the bottom of the entry form. Once copied, the comments can be edited, freeing the inspector from having to entirely rewrite the comments each time. When selecting numeric NBI ratings, the inspector sees a ratings wheel displayed which depicts a pie graphic with the ratings N and 0-9 (see Figure 6-11). This format allows the inspector to quickly select the desired rating on a mobile device with or without the use of a stylus. The N rating was included at MDOT's recommendation to allow for a "not applicable" option when a bridge does not contain that particular component.

Figure 6-10: NBI Rating shortcut entry form, accessible from any bridge defect menu.

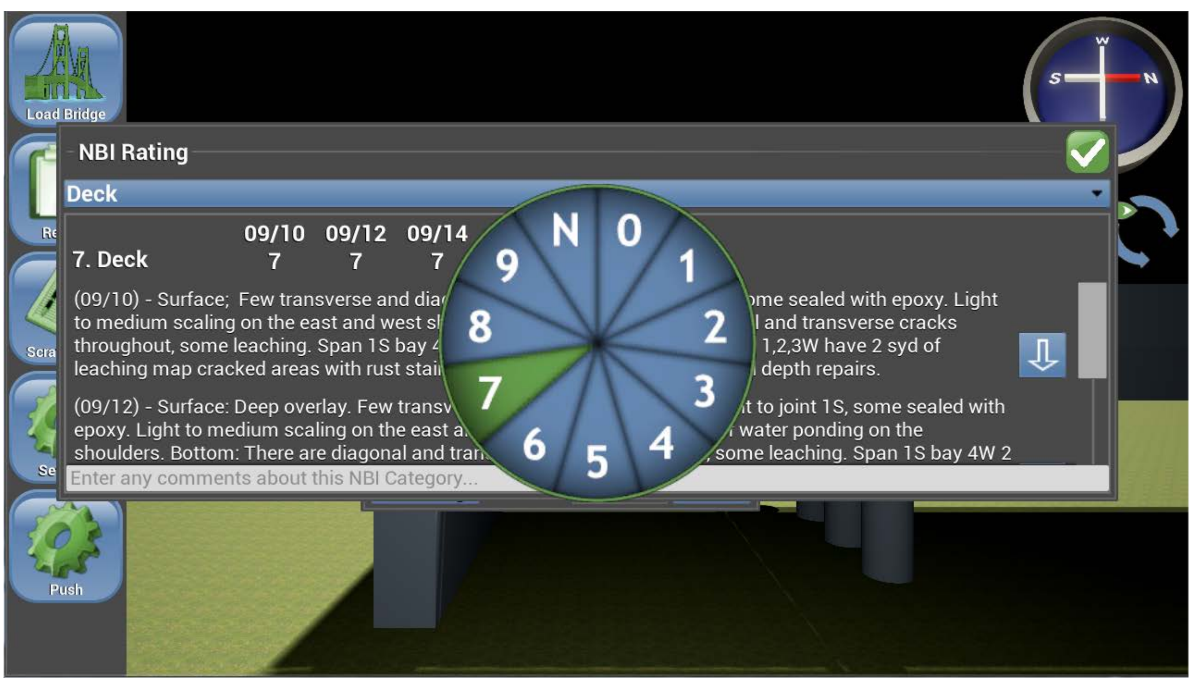


Figure 6-11: The ratings wheel is a touch-friendly interface for quickly selecting NBI ratings.

6.7 Scratch Pad

At the request of inspectors following field demonstration reviews, a scratch pad interface was implemented. The interface consists of a white space upon which the inspector is free to draw or write something of interest (see Figure 6-12). The interface includes several sizes of brushes for drawing and erasing as well as a Clear Screen option. Writing is best done with a stylus since fingers are too large for small text, but drawing can be done easily with either tool. Currently, the scratch pad's content is not recorded within the inspection and is purely for the inspector's personal use. Future development work could include extending the scratch pad tool set to create overlay drawings for pictures associated with bridge defects, allowing inspectors to highlight problem spots or write comments. Such photo overlays could be included with the photo data uploaded to the server to facilitate management review of inspection data.

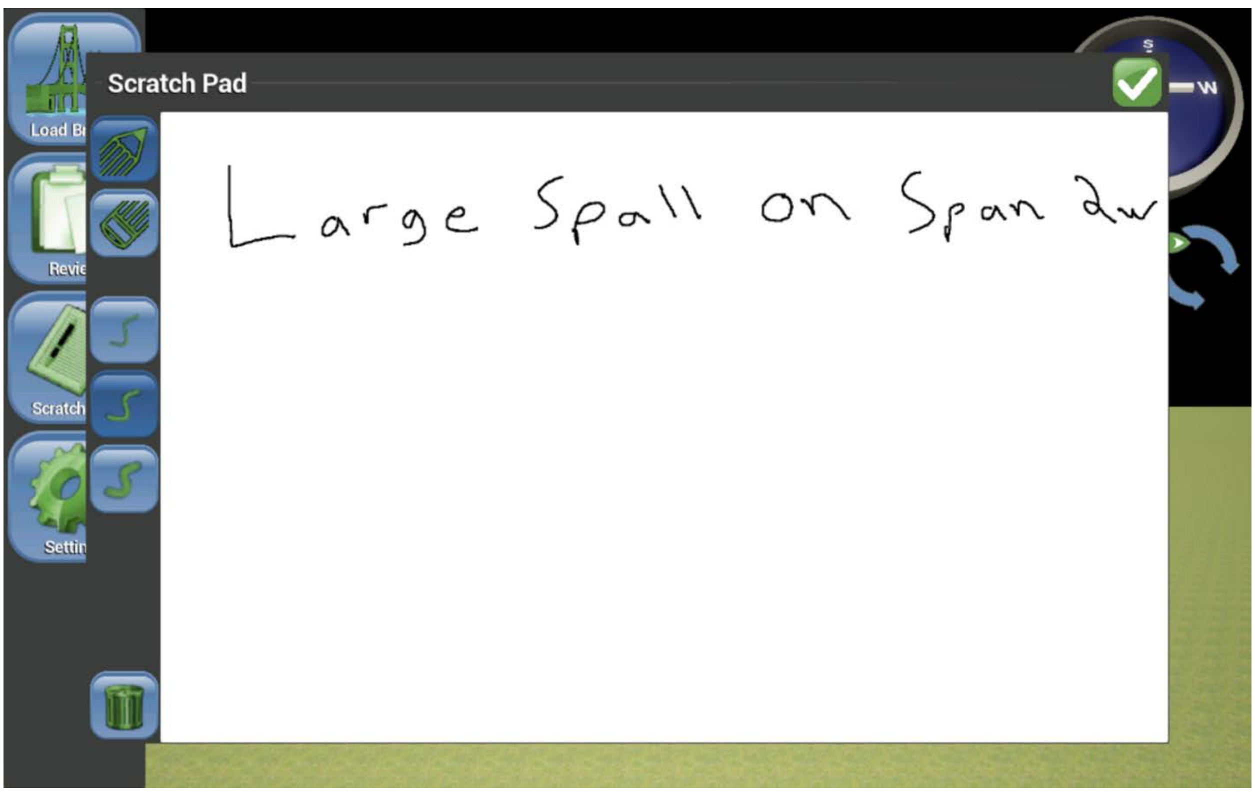


Figure 6-12: The scratch pad gives inspectors a place to write/draw notes that are not included in the report.

6.8 Linear Defects and Defect Aggregation

Certain bridge elements, such as railings and abutments, are measured in linear feet rather than area. Since all defect information is handled by placing area defects on a surface, there must be a method for converting area-based defects to linear quantities. The application handles this by projecting the polygons of the area defects onto a one-dimensional line at the base of the elements. For example, defects on the inside, outside, or top of the railing will be projected onto a line parallel with the long dimension of the railing before aggregation, while defects that are placed on the ends of the railing will be excluded from the aggregate value since they do not contribute to the linear quantity. Aggregation proceeds from Severe defects to Poor and then to Fair. At each condition-rating stage, when the aggregate quantity for that stage is computed, area that overlaps with regions that have a more severe condition rating are excluded. The result is a total linear quantity for the element in which all area defects are included but in which overlapping quantities are not counted multiple times: A severe defect located spatially below a poor defect will supersede the poor defect in the aggregate quantity. In the defect pop-up menu, the inspector may choose to define a particular defect as linear; however, such a defect will be represented in the application as a quadrilateral polygon with an assumed width of 6 inches. These “linear defects” serve as a quick way to represent cracks, but it is the projection algorithm that truly computes the linear quantity.

The bridge deck is the largest element in any bridge model, and typically will have defects on the top and bottom surface. The bridge deck does use an area-based metric, so area defect aggregation must occur in a 2D plane. All defects are projected into the 2D plane, and then aggregation proceeds analogously to the 1D case (described in the previous paragraph), in which the most severe defects are aggregated first and then the combined region is excluded from overlapping but less severe defects. The application uses a polygon operator library, Clipper Lib, to perform the necessary polygon union and intersection operations.

6.9 Saving/Loading and Importing/Exporting

As with any computer-based application, it is vitally necessary for the users to be able to save and load their work at any time to the local device. Such capability is a hedge against software failure and user error. To this end, the application includes both a named-save file system and an autosave feature. At any time, the user may enter a unique name identifying a particular inspection and then save the current progress of that inspection as a file on the local device bearing the chosen identifier. These save files may be restored at any time, and will return the loaded bridge model, all defects, and NBI report data to the state they were in at the time the save file was created, allowing the user to undo inadvertent changes or to resume the inspection at a later time. The autosave feature activates every time the user modifies a defect on the bridge surface. There is only a single autosave for the entire application, so it is not a reliable way to save data for future us as it is frequently overwritten, but it does provide a way to recover quickly from a software failure such as an application crash, or from a limited hardware failure such as a depleted battery. Once the device is operating properly and the software is running, the autosave may be loaded, restoring the inspection to the state it was in as of the most recent modification to any bridge defect.

The final critical element is importing and exporting inspection information so that it may be integrated into the MDOT BMS database. Exporting an inspection generates an XML file that includes the original bridge model and NBI information, but included are all the NBI values as well as each individual defect and its location on the bridge model surface. As an XML file, this information can be uploaded to MDOT servers and processed into database entries documenting the inspection. When the same bridge is inspected in the future, the same XML format may be used to generate a new inspection that includes the previous inspection data, which can then be imported into the bridge inspection application. This import/export system was implemented as an interim substitute for full integration of the 3D Bridge App with the MDOT BMS database. Full integration is awaiting MDOT approval that fits into its schedule of database upgrades.

7. Conclusions

After reviewing nationwide bridge inspection practices and discussing current practices and needs with bridge inspectors, MTRI staff developed the 3D Bridge App to render 3D bridge models and interactively tag them with AASHTO element-level defect information. Currently, bridge models are generated using information gleaned from MDOT's BMS database and then tuned with user input. The new system will allow bridge inspectors to gather element-level information efficiently while eliminating the manual data entry present in the current state of practice.

While this project had a specific scope, future development of the 3D Bridge App would be a logical and very promising follow-on to the first two phases of development and implementation-focused improvement. Should MDOT develop a more detailed set of bridge models (such as by obtaining the engineering design files used in bridge construction) that have the necessary metadata, such as element type and category (Deck, Substructure, etc.), then the application could be modified to work with those models rather than the generic models derived from database attributes. The digital nature of the application also makes it ripe for integration with other operations such as remote sensing overlays and GPS tracking. The app could be extended to work with larger, more complex bridges. Finally, the app's per-defect approach to bridge markup opens up new possibilities for bridge management decision-making and represents a step beyond the current inspection regulations, since the app captures the location of defects in addition to their quantities.

Altogether, MDOT's 3D Bridge App affords cutting-edge improvements in the bridge inspection process, enhancing the efficiency and quality of data collection and interpretation.

8. Appendix

8.1 List of Acronyms and Abbreviations

3DWBIS	3D Wireless Bridge Inspection System
App	Application
AASHTO	American Association of State Highway and Transportation Officials
BMS	Bridge Management System
CoRe	Commonly Recognized
DoF	Degree of Freedom
DTMB	Department of Technology, Management and Budget
FHWA	Federal Highway Administration
HUD	Heads-Up Display
MBIS	Michigan Bridge Inventory System
MBRS	Michigan Bridge Reporting System
MiBRIDGE	the Michigan Bridge Management and Inspection Systems
MDOT	Michigan Department of Transportation
MTRI	Michigan Tech Research Institute
NBI	National Bridge Inventory
OS	Operating System
PI	Principal Investigator
SIA (SI&A)	Structure Inventory and Appraisal
UE4	Unreal Engine 4
XML	Extensible Markup Language

8.2 State of Practice and Literature Review

MichiganTech
Research Institute

www.mtri.org

State of the Practice and Literature Search Review for Wireless Bridge Inspection Data Collection Technology

Colin Brooks (cnbrooks@mtu.edu), Nate Jessee, David Banach, and Dr. Tess Ahlborn, P.E., FPCI

February 15, 2014

Deliverable for Michigan Department of Transportation
Contract #: 2013-0067, Authorization No. 2,
OR #: 0R10-043
"Wireless Data Collection and Retrieval of Bridge Inspection/Management Information"

Michigan Tech Research Institute (MTRI)
3600 Green Court, Suite 100 • Ann Arbor, Michigan 48105

Table of Contents

Contents

Introduction

Under federal regulations, all state-managed bridges located on public roads, fully or partially contained within a State's boundary, must be inspected with a minimum frequency of 24 months (23 CFR 650.307 and 23 CFR 650.311, 2009). To comply with the rules and regulations set forth in the Code of Federal Regulations (CFR), state transportation departments utilize a variety of bridge inspection tools, databases, and management solutions to comply with the federal regulations. Currently, there are a limited number of commercialized software solutions to help agencies with the task of collecting and recording bridge inspection data. Because of the relative lack of options, State DOTs have designed their own customized systems to meet their particular needs or utilize one of the commercial solutions. This state of the practice document examines the existing solutions and discusses some of the custom tools currently being utilized.

Existing Commercial Solutions

To assist infrastructure monitoring and maintenance, the commercial industry has developed tools and systems that are compatible with federal regulations. The most well known of these organizations is the American Association of State Highway and Transportation Officials (AASHTO). AASHTO is the official source for transportation organization, technical excellence and advocates for transportation related policies, technical services and support for state transportation needs (AASHTO, 2014). AASHTO supports the most popular product for bridge management solutions, including AASHTOWare Bridge Management™ (BrM) Software (formerly Pontis, http://aashtowarebridge.com/). This software integrates the entire bridge management process, from data entry and federal reporting to public safety and risk reduction. Some of the features of BrM include:

- Functional Geographic Information System (GIS) utilizing Google mapping technology
- Add on applications for mobile devices (such as the iPad) which allow for data collection on these mobile devices
- Accurately assess performance and risk
- Cost calculations and budget assistance
- Priority and need assessment

These features of the AASHTOWare BrM are not entirely exclusive to the BrM software. One of the more resent upgrades to the system is the previously mentioned application add on. Through an agreement with Bentley Systems, AASHTOWare BrM can be compatible with mobile data entry systems developed by Bentley and marketed as InspectTech©.

InspectTech©, a division of Bentley Systems, is a commercial firm specializing in mobile inspection and asset management solutions. Their software platforms are compatible with NBI, Pontis/Element level data, and custom databases. There are two core products offered by InspectTech©: BridgeInspect™ Collector and BridgeInspect™ Manager. The Collector software can be utilized in the field on mobile devices such as laptops and tablets as well as in the office settings. This software allows for advanced reporting formats, editing functionality as well as quality assurance and controls. One of the highlighted features of this software is that it can incorporate digital pictures directly into the reports for easy and effective use. The Manager software is also equipped with many efficient tools such as database management, report generation tools and a functional Geographic Information System (GIS). The following groups/agencies currently use this software: the United States Fish and Wildlife Service, the Indiana Department of Transportation, Washington Metropolitan Area Transit Authority (DC Metro), New Jersey Turnpike Authority, Minnesota Department of Transportation, Iowa Department of Transportation, and Montgomery County, Maryland. (http://www.inspecttech.com)

The Minnesota Department of Transportation utilizes the InspectTech© software in their system, Structure Information Management System (SIMS, http://www.dot.state.mn.us/bridge/bridgereports/index.html). The workflow of the SIMS systems consist of collecting data in the field, entering this data into the database once back in the office, review and generation of the report. This report is reviewed, approved, and made available through a search function within the database. *Figure 1* is a flowchart designed by the Minnesota Department of Transportation for SIMS.

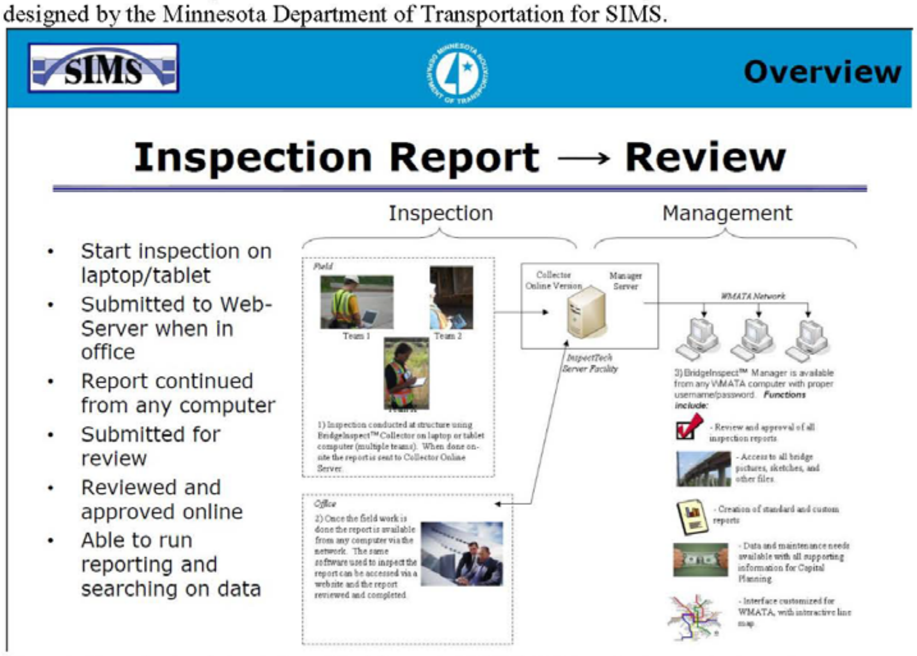


Figure 1: Flow chart and description of the Minnesota DOT SIMS data collection and entry system.

Bentley Systems is also known for their ProjectWise Information Management and Collaboration products (http://www.bentley.com/en-US/Products/projectwise+project+team+collaboration/). The ProjectWise product is a series of software platforms, which are exclusively focused on architecture, engineering, construction, and operations (AECO). Part of this information management system is specifically dedicated to bridge infrastructure. The Bridge Information Modeling (BrIM) system is designed to ease the design and development of a bridge structure as well as maintenance and inspection. *Figure 2* is a diagram of the ProjectWise BrIM information network (http://www.bentley.com/en-US/Solutions/Bridges/brim.htm).

Figure 2: A diagram featuring the information network that can be utilized with the ProjectWise BrIM software.

While the Michigan Department of Transportation (MDOT) utilizes ProjectWise for construction purposes and limited storage for bridge inspections, it is MDOT's overall goal to utilize the MiB^{RIDG}E application to find and retrieve all current information regarding bridge inspection data (Rich Kathrens, Bridge Safety Inspection Engineer, MDOT, January 2014 personal communication).

In addition to AASHTOWare BrM and the InspectTech© systems, another commercial product for bridge management and inspection is the AgileAssets Bridge Analyst and Bridge Inspector software. Similar to the other available software packages, the AgileAssets Bridge Analyst has a built in GIS, offering analysis of bridges to the inventory level with specific algorithms to analyze what-if scenarios and fine tuned to achieve the highest return on investment for management decisions. This software is flexible enough to be configured to meet many management agencies requirements. When the Bridge Analyst package is coupled with the Bridge Inspector software, the user has a system that will allow for compliance to federal regulations through generating National Bridge Inventory reports, National Bridge Inventory System (NBIS) and Structure Inventory and Appraisal (SI&A) criteria. *Figure 3* is an example of the Bridge Analyst GIS interface.

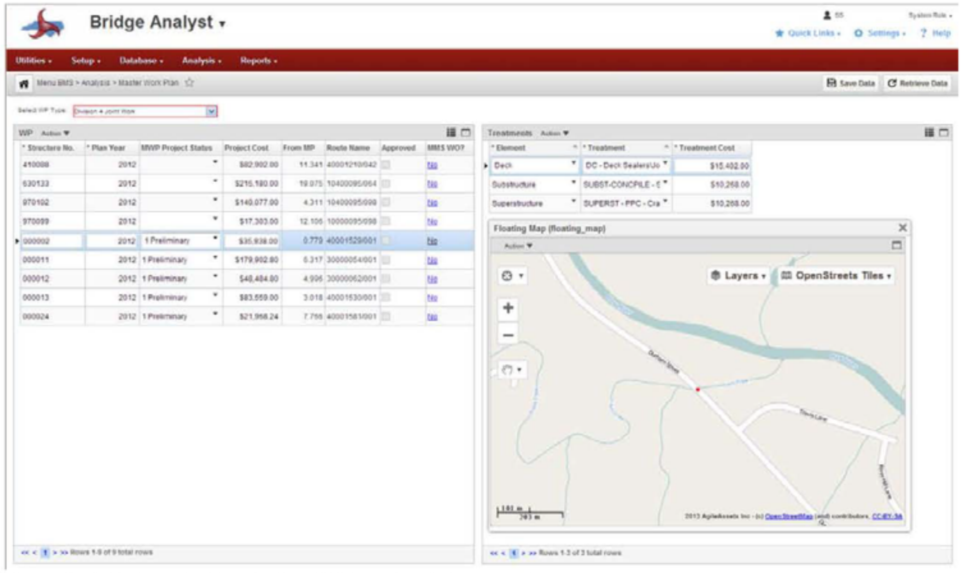


Figure 3: AgileAssets Bridge Analyst GIS interface is featured above.

The AgileAsset software is web-based so mobile devices can have access when an internet connection is available (http://www.agileassets.com/solutions/road-bridge/). In addition to these Bridge Analyst and Bridge Inspector packages, AgileAssets offers AgileAssets mobile. This mobile application is compatible with smartphones, tablets, and laptop PCs, and offers online and offline access for data collection, report generation, and work orders. In addition to the application, AgileAssets Mobile Inventory Manager is a GPS enabled Windows device that works regardless of cell phone network availability. This device allows managers to record work orders, equipment orders, and other cost factors (http://www.agileassets.com/products/mobile-apps/). To operate any of these packages, the user will need to purchase and install the AgileAssets System Foundation package that serves as the root of all the products offered by AgileAssets.

Another management and inspection tool available is the BridgeWeb management system (http://www.bridgeweb.us/). Like many of the other tools available, BridgeWeb is broken in task specific programs. BridgeWeb offers a Bridge Management System (BMS), an Inspection Management System (IMS), and a Data Collection System (DCS). The BMS system is a web-based software designed to provide management insight that prioritizes tasks by assessing bridge condition, deterioration, and cost to effectively manage bridge maintenance. *Figure 4* is a flow chart of the BMS organization.

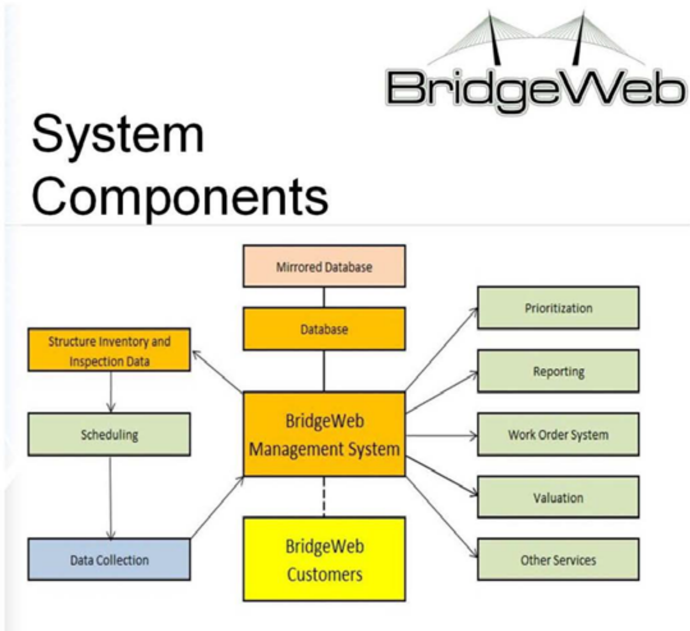


Figure 4: A flow chart describing the components of the BMS system offered by BridgeWeb.

The IMS is a system designed to help oversee large-scale inspection programs. Offering inspection scheduling, quality assurance and control, reporting, and other features, this system can provide managers solutions to large scale inspection programs. The DCS is marketed as an easy to use tool that can incorporate text, photos, and tables using a stylus-based tablet. This tool is designed to provide the necessary information to inspectors, such as past inspection reports and real time quality control, and has an interface that allows inspectors to move from one element to another. This system can upload collection data to the BridgeWeb server through an internet connection.

Customized Bridge Inspection Data Collection and Management Solutions

While commercial data collection and management solutions exist, some state transportation departments utilize multiple software platforms to meet their needs. Alternatively, some cases exist where agencies are developing customized data collection and management systems to meet their needs.

The Utah Department of Transportation (UDOT) is developing a design, management and inspection system that utilizes technology from multiple sources. The program, known as the Utah Transportation Integrated Business System (UTIBS) is a collaboration of multiple sources of expert technology and innovation. Some of the collaborators in this project are Bentley Systems, Agile Assets, Oracle Corporation (database management), and Esri (GIS provider). This system appears to be similar to ProjectWise, but with a custom user-end solutions. The goal of this project is have all data accessible in a single portal with no data duplication, to easily and accurate update and complete projects, and a seamless work environment for all systems. *Figure 5* is a diagram of the desired work flow for the UTIBS program (UTIBS presentation, accessed January 2014, http://www.udot.utah.gov/main/uconowner.gf?n=4915503534970595).

Figure 5: Workflow diagram envisioned by UDOT for the UTIBS program.

The Oregon Department of Transportation (ODOT) currently utilizes AASHTOWare BrM software to help manage their bridge inspections. While this is a satisfactory management solution, ODOT is also looking to create an efficient inspection device technology. With the goal of utilizing touch screen tablet technology, ODOT has experimented with using iPad and Think-Pad Helix (Windows based) devices to collect bridge inspection data. This system is still in the development phase and ODOT is facing some obstacles. The iPad is having trouble getting through IT security measures in place through the ODOT network. Also, the Think-Pad was developed and sold with Windows 8 and the ODOT network is operating on Windows 7. When the Think-Pad was backloaded to Windows 7 many of the features, such as 4G mobile connection and touch screen technology, would not operate correctly and would cause the operating system to crash (Erick Cain, Bridge Inventory Coordinator, ODOT, December 2013 personal communication). We are maintaining communication with the ODOT team to keep track of progress with their system.

Advitam ScanPrint Infrastructure Management System (IMS)

Advitam (www.advitam-group.com), an infrastructure monitoring and management group, is currently using handheld tablets for bridge inspection data management. Advitam's ScanPrint Infrastructure Management System (IMS) software allows users to optimize management, monitoring, and maintenance of infrastructure features. By incorporating ScanPrint IMS on handheld tablets, infrastructure inspectors are able to simultaneously assess the conditions of features and make updates to records while conducting assessments in the field. Any updates are transmitted in real-time, giving insight into incident, alerts, or scheduled future work. Any defects can be drawn directly into the software via the handheld tablet and incorporated into an interactive geographic information system (GIS), which provides a complete inventory of the infrastructures properties and information. An additional component of the software is a computerized maintenance management system (CMMS), containing scheduling and managing interventions of the assessed feature.

ScanPrint IMS is based on an interactive cycle, which is made up of six different components allowing managers, consultants, and users the ability to adjust the program and cycle to their distinctive needs. The six components are as follows: "Know, Track, Evaluate, Decide, Act and Share" (http://advitam-group.com/Home/ComputerizedInspections). The first component "Know" permits users to access data concerning the feature of interest, including historic and current information. The data also incorporates any previous changes that have been made to the record. "Track" will allow inspectors to perform inspections in the field and note any structural defects that need to be further assessed. Next, "Evaluate" allows an inspector to assign structural ratings using the data collected during the previous component. The ratings can include only individual distresses or the entire structure. "Decide" uses the ratings from the "Evaluate" module and generates a report that describes the best way to maintain the structure, which includes budgetary scenarios and calendars to provide a temporal analysis on proposed actions. "Act" provides a variety of management tools for further planned activities. Lastly, "Share" allows the inspection group to provide reports, charts, and other data to the required users.

The ScanPrint IMS program has been used in detailed analysis by transportation departments within the United States. The Michigan Department of Transportation (MDOT) was the first American based organization to use Advitam's software for infrastructure assessment (Finley, 2005). The Zilwaukee Bridge, located on Interstate-75 in Saginaw, Michigan, is regularly inspected every two years to meet NBI standards, and every four years to assess deterioration and long-term health (http://images.autodesk.com/adsk/files/4043445_Michigan_Zilwaukee_Bridge.pdf). Previous to incorporating ScanPrint IMS, each assessment would result in thousands of pieces of paper documents and bookshelves of binders, which would have to be sorted through in order to find a specific document (Finley, 2005). However, once MDOT implemented ScanPrint IMS on handheld tablets, the time it took to share data and gather information concerning the Zilwaukee Bridge dramatically decreased (*Figure 6, http://www.finleyengineeringgroup.com/cfcs/cmsIT/baseComponents/fileManagerProxy.cfc?method=GetFile&fileID=8DE40D2B-F1F6-B13E-8A6745DAEFDF71F4*). Upon the completion of the 2001 project, Advitam updated all of MDOT's previous inspections and graphics for the Zilwaukee dating back to 1993 into the ScanPrint IMS. Additionally, the Zilwaukee Bridge was digitally split into 700 sections with corresponding reports and inspections attached to each within the software. As MDOT continues to inspect the bridge every two years, inspectors will be able to draw any changes in deterioration and fill in forms and reports for any of the 700 elementary sections.

Figure 6. An example of a handheld tablet device that allows inspectors to make on-site updates to bridge inspection records when using Advitam ScanPrint.

Examples of Mobile Inspection Technology in Other Industries

Mobile and tablet technology can also be used for analyses pertaining to environmental health, building, and housing code inspections. One such company that uses this technology is Inspect2GO (www.inspect2go.com). Inspect2GO is primarily used for health inspections within food, school, water and sewage establishments. The software can perform analyses on a variety of platforms including mobile phones, iPad, and Android tablet and does not require an internet connection (*Figure 7*). Inspect2GO inspection solutions is comprised of three components; Inspect, Store, and Manage. The inspect component consists of checklists, photos, notes, etc. that are required for on-site inspections. These forms and data can be formatted to fit each location's needs, and completed on-site and stored for later retrieval within Inspect2GO's cloud database. In addition, the data can also be emailed and formatted into PDF formats. Lastly, the data can be retrieved from Ispect2GO's cloud, managed and analyzed in charts, graphs, documents, and reports. As the database becomes more complex, search queries of customer inspection reports and violations can be completed. All inspections, permits, and violation reports are available to the public.

Figure 7. Inspect2GO's inspection software on a mobile tablet.

Sungard Public Sector's ONESolution software also uses mobile technology for inspections, (www.sungardps.com/solutions/mobile). ONESolution technology is intended to provide tools and forms to public safety and local governments during building inspections and crime related circumstances. Similar to Inspect2GO, ONESolution does not require internet access to conduct an inspection, and results are stored and retrieved upon the restoration of connectivity. Upon completion of a building inspection, fees and penalties can be distributed and assessed while on location. Additionally, ONESolution technology is also used in emergency situations in the Records Management System (RMS) (*Figure 8*). Public safety agencies can collect, store, and analyze data gathered during an investigation or within a correction system.

Figure 8. ONESolution RMS communicate crime statistics and data during public safety investigations

IMEC Technologies Inspection System (http://www.imectechnologies.com/) is a web based inspection system used for the auditing and inspection of equipment within work environments for insurance or monitoring purposes. Each item assessed is given a unique barcode or tag and upon being scanned by a smartphone or handheld device, the inspection system assigns the item a unique identification and location using GPS. Photos and comments can also be assigned to each item. By assigning a GPS location, each item can be accurately monitored as it is moved. This especially proves useful during the transportation of hazardous materials and regulatory requirements. Additionally, this software can be used for safety compliance management, similar to Inspect2GO and ONESolution. However, IMEC Technologies prohibits the use of a scaling system based on an individual inspector's judgment. Instead, the software determines the safety score based on the response given during audit questions and calculated by the system and administrators. *Figure 9* is an image of the IMEC Inspector software supporting mobile device platforms.

Figure 9: IMEC Inspector software dashboard showcasing mobile device usage.

Mobile Technology Hardware Advances

Over the past few years, the capabilities of mobile computing and wireless data connection technology have grown considerably. With more options of size, storage capacity, and operating systems, mobile computing is becoming a practical solution for data collection and recording. Currently, there are three dominant operating systems on the market: Apple iOS, Google Android, and Windows. Because of the nature of the Windows and Android operating systems, multiple companies can develop a suite of hardware designs to fill in niche markets such as rugged tablets and computers. For example, if a desired product is something more lightweight, a Panasonic Toughpad is water and dust resistant to work in many different environments while utilizing the Google Android operating system and is available with 4G LTE connectivity data options (http://www.panasonic.com/business/toughpad/us/secure-tablet-specs.asp). Figure 10 is an example of the Panasonic Toughpad; a typical system cost $1500.

Figure 10: The Panasonic Toughpad featured above is water and dust resistant and utilizes the Google Android Operating System.

Alternative to an operating system designed for mobile application, if a user needed a system to run more sophisticated software such as a Geographic Information System, there are rugged Windows tablets designed by Algiz. The Algiz 10x features a 10 inch screen designed with high visibility screens to be used in any environment and rugged enough to handle light water and dust exposure and impacts. This tablet runs Windows 7 Ultimate and is capable of wireless internet connections through WLAN networks (http://www.ruggedalgiz.com/algiz-10x/). Figure 11 is an image of the Algiz 10x rugged tablet; a typical system costs $2,785.

Figure 11: The Algiz 10x rugged laptop running the Windows 7 Ultimate operating system.

In addition to the Algiz rugged Windows tablets, Xplore Technologies offers a rugged Windows tablet with 4G LTE (http://www.xploretech.com/products/ix104c5_DMSR-LTE/). This tablet runs Genuine Microsoft Windows 7 and Windows 8 (both offered in 32 and 64 bit options). This tablet is designed to resist rain, sand, dust, heat/cold, and forceful impacts. Figure 12 shows the Xplore Technologies iX104C5 DMSR LTE Rugged Tablet.

Figure 12: Xplore Technologies iX104C5 DMSR LTE Rugged Tablet featuring Windows operating systems and a 4G LTE wireless data connection.

While Apple does not currently offer rugged versions of their iPad tablet, there are many weatherproof cases on the market that can be paired with the iPad if the iOS operating system is desired. A typical mid-level iPad Air costs $829, while a rugged OtterBox case costs $99 (http://www.otterbox.com/Defender-Series-Case-for-iPad-Air/apl2-ipad-air-set,default,pd.html?start=2&q=ipad) (see Figure 13). If Apple tablets, and potentially smartphones, can be tough enough to survive practical use in field environments, then iPads and iPhones could be less expensive platforms for field data collection. This platform is under evaluation because of the widespread adoption of iPads and iPhones by transportation agencies on an at least personal basis, so many of these could be available in the field for inspection purposesq. Similar Android tablets and phones, such as the Samsung Galaxy Note

10.1, which include a useful stylus, can also be obtained for less than $800 (see Figure 14, http://www.samsung.com/global/microsite/2014galaxynote10.1/).

Figure 13: iPad Air with rugged Otterbox case

Figure 14: Samsung Galaxy Note 10.1 Android-based tablet with stylus that can be useful for field data entry.

Also under consideration are Windows tablets. Windows 4G LTE devices such as Nokia 2520 run windows RT 8.1 (http://www.nokia.com/us-en/phones/tablet/lumia2520/?cid=ncomprod-fw-src-na-alwaysongenerictablet-na-google-us-en_us-1todtmx57c909) (Figure 15). Windows RT is a tablet-specific version of Windows that does not run legacy applications such as Excel or Word 2010, but runs more efficiently on tablet formats than a full version of Windows. If the bridge data collection inspection tool is primarily web browser based, as planned, then a 4G Windows RT tablet could be a solution. A typical Nokia 2520 Windows RT 8.1 table with 4G costs $500, so the price point is attractive. This format is also under evaluation by the project team. Tablets are offered with screen sizes ranging between 7" – 8" and 10" – 11" but the project team is focused on utilizing the larger screens giving inspectors more room to navigate the software. There are also <$1,000 Windows tablets running the full version of Windows 8.1 such as the Microsoft Surface 2 Pro, but none are yet available with integrated 3G/4G data service. This is a rapidly developing area of tablet capabilities so the project team is monitoring availability of newer systems. A tablet running full Windows 8.1 (with 4G) would be able to run traditional Windows applications such as Word and Excel while also running a browser-based data collection tool, so this could be a promising solution as the market develops.

Figure 15: Windows RT 8.1 Nokia Lumia 2520 tablet, available with 4G for $500.

Concluding Comments

This report describes the state of the practice for bridge inspection data collection solutions. Existing commercial software tools such as AASHTOWare BrM with its InspectTech© addons are reviewed, along with AgileAssets Bridge Inspector software and Advitam ScanPrint. BridgeWeb is also described. Examples of solutions used at the Minnesota Department of Transportation, the Utah Department of Transportation, and Oregon DOT are also reviewed. Solutions from the health inspection, building and crime inspections, and insurance are described. Finally, examples of more expensive fully rugged tablets and less expensive Apple, Android, and Windows tablets are described so that a representative range of hardware data collection platforms appropriate for use by bridge inspectors in the field can be understood. While many software solutions are available, state departments of transportation are also choosing to develop customized solutions to meet their particular needs while still being compatible with new element-level inspection requirements. The project team is focused on making sure that any such solution would meet the needs of the Michigan Department of Transportation.

References

1. 23 CFR 650.307, 23 CFR 650.311, Accessed January 2014. http://www.gpo.gov/fdsys/granule/CFR-2011-title23-vol1/CFR-2011-title23-vol1-sec650-307/content-detail.html

2. The American Association of State Highway and Transportation Officials (AASHTO). Accessed January 2014. http://www.transportation.org/Pages/Organization.aspx

3. AASHTO Bridge Management. Accessed January 2014. http://aashtowarebridge.com/

4. InspectTech©. Accessed January 2014. http://www.inspecttech.com/

5. Minnesota Department of Transportation, Structure Information Management System (SIMS). Accessed January, 2014. http://www.dot.state.mn.us/bridge/bridgereports/

6. ProjectWise Information Management, Accessed December 2013. http://www.bentley.com/en-US/Products/projectwise+project+team+collaboration/

7. ProjectWise Bridge Information Modeling System BrIM. Accessed January 2014. http://www.bentley.com/en-US/Solutions/Bridges/brim.htm

8. Personal Communication, Rich Kathrens, Bridge Safety Inspection Engineer. January 2014.

9. AgileAssets Bridge Analyst. Accessed January 2014. http://www.agileassets.com/solutions/road-bridge/

10. AgileAssets Mobile. Accessed January 2014. http://www.agileassets.com/products/mobile-apps/

11. BridgeWeb. Accessed January 2014. http://www.bridgeweb.us/

12. Utah Transportation Integrated Business Systems (UTIBS). Accessed January 2014. http://www.udot.utah.gov/main/uconowner.gf?n=4915503534970595

13. Personal Communication, Oregon Department of Transportation, Erick Cain, Bridge Inventory Coordinator, December 2013.

14. Advitam. Scanprint IMS The Infrastructure Management Solution. Accessed December 2013. http://advitam-group.com/Content/brochures/EN/Scanprint%20IMS_En_A4_LR.pdf

15. Autodesk, Inc. Michigan DOT builds a technological bridge to improved efficiency Re-engineering a paper standard. Accessed December 2013. http://images.autodesk.com/adsk/files/4043445_Michigan_Zilwaukee_Bridge.pdf

16. *Figure 6*, Finley Engineering. Accessed January 2014. http://www.finleyengineeringgroup.com/cfcs/cmsIT/baseComponents/fileManagerProxy.cfc?method=GetFile&fileID=8DE40D2B-F1F6-B13E-8A6745DAEFDF71F4

17. Finley, C., 2005. Project Case Study: Bringing bridge inspection into the computer age. CE NEWS. http://www.cenews.com/search-zilwaukee_bridge.html

18. Inspect2Go. Accessed January 2014. www.inspect2go.com

19. ONESolution Records Management System (RMS). Accessed January 2014. www.sungardps.com/solutions/mobile

20. IMEC Technologies Inspection System. Accessed January 2014. http://www.imectechnologies.com/

21. Panasonic ToughPad. Accessed January 2014. http://www.panasonic.com/business/toughpad/us/secure-tablet-specs.asp

22. Algiz 10x Rugged Windows Tablet. Accessed January 2014. http://www.ruggedalgiz.com/algiz-10x/

23. Xplore Technologies Rugged Windows Tablet with 4G LTE. Accessed February 2014. http://www.xploretech.com/products/ix104c5_DMSR-LTE/

24. OtterBox Rugged iPad case. Accessed January 2014. http://www.otterbox.com/Defender-Series-Case-for-iPad-Air/apl2-ipad-air-set,default,pd.html?start=2&q=ipad

25. Samsung Galaxy Note 10.1. Accessed January 2014. http://www.samsung.com/global/microsite/2014galaxynote10.1/

26. Nokia Lumina 2520 running Windows RT 8.1. Accessed January 2014. http://www.nokia.com/us-en/phones/tablet/lumia2520/?cid=ncomprod-fw-src-na-alwaysongenerictablet-na-google-us-en_us-1todtmx57c909

8.3 Survey Results

Wireless Data Collection and Retrieval of Bridge Inspection/Management Information: Survey Response Summary as of April 29, 2014

Colin Brooks (cnbrooks@mtu.edu), Tess Ahlborn (tess@mtu.edu), and Nate Jessee (nljessee@mtu.edu)

Michigan Technological University (MTU) and its research center the Michigan Tech Research Institute (MTRI) are developing a data collection system specifically designed to acquire bridge inspection data, with a focus on element level information. This research is being done as a part of the Michigan Department of Transportation project "Wireless data collection and retrieval of bridge inspection/management information" (OR14-021). To help our research staff gain better insight to the current practices and applied technologies currently available, a short survey was designed and distributed to bridge inspection managers across the United States by Matt Chynoweth of MDOT on March 26, 2014 to the AASHTO Subcommittee of bridges and structures e-mail list . This document will summarize the responses that have been submitted to date.

As of April 29, 2014 a total of twenty-one survey responses have been received. The following states have responded: Delaware, Florida, Hawaii, Illinois, Iowa, Maryland, Minnesota, Missouri, Nebraska, New Hampshire, New Jersey, New York, North Carolina, North Dakota, Ohio, Oklahoma, South Dakota, Texas, Wyoming, Utah, and Virginia. The responses represent a wide range of States and the answers vary in detail and have provided a valuable source of information. A portion of the survey responses will be shared at the end of this document, and all responses will be shared as a separate document once the survey deadline of May 2, 2014 has passed.

So far, over 70% (15 of 21) of responding States use some form of electronic hardware to collect and manage bridge inspection data. Of those using electronic devices, over half use laptop computers to collect and manage data. Currently, there is a moderate number of commercial software packages available for transportation infrastructure management and inspections, as described in the State of the Practice Report submitted to MDOT from the MTRI/MTU research team. Despite the available software packages, the survey responses indicate that many of the bridge inspection departments utilize "in-house" or custom software for data collection and management. In addition to the "in-house" systems, many other State departments use existing commercial software, but have customized the standard software package to fit their specific needs. Of the twenty-one responses, only four use commercial products in their native format. The high number of custom/self-made systems indicates that a single software platform may be unlikely to meet the needs of multiple states or infrastructure departments. Those States that did not utilize an electronic data collection and management have indicated that they are likely to adopt such a system, but some concerns were raised, such as the cost of equipping a staff of bridge inspectors with mobile electronic devices. Also, over 80% of those that responded indicated that they are willing to be contacted for additional surveys/questions.

These survey responses will be useful to our research and software design. For instance, understanding that many of these agencies are interested in obtaining or improving their bridge data collection and management systems, such as stated by Mike Brokaw of the Ohio DOT, that "Ohio is transition to a web-based system (and that) DOT inspectors collect inspection and maintenance data on an in-house accessed based interface, (which is) decoupled from state servers" is of importance (Mike Brokaw, survey respondent, 2014). Specific phrases, such as "decouples from state servers" helps remind that security is often a priority for database managers. Other information, such as flexibility in personnel usage of software is also an insight that may help determine how modern hardware and software can accommodate users who may be hesitant to adopt new practices. Such an instance was stated by the Minnesota DOT, "Some inspectors prefer to print out copies of the last inspection form the database to take out in the field" (Jennifer Zink, survey respondent, 2014).

Part of our current research vision involves the use of mobile internet (4G LTE connection specifically). Being able to access inspection databases while in the field seems to be a valuable feature for a future inspection and management system. To understand how frequently inspectors would have cellular coverage, we asked the participants to rate how often 4G connections would be available to their inspectors and the answers were rather divergent. Approximate 24% of the responses indicated that cellular coverage/availability in their districts was less than 50%. The number of responses indicating that cellular coverage was greater than 75% was near the same as it was for those with less than 50% coverage. Unfortunately, this question was misunderstood by approximately 20% of the participants, with one responder stating that cellular coverage was never available; we believe that the responder meant to say that they are not equipped with cellular devices while in the field. The following figures breakdown some of the responses that could be quantified. Lastly, the Virginia DOT listed that they use existing commercial software that allows them to include sketches into their work. We intend to follow up with them about this feature as it is a high priority for our current software system.

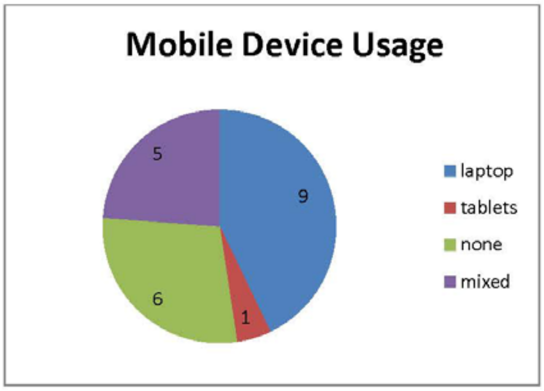


Figure 1: These statistics detail the mobile device usage of the responding agencies. The mixed category includes agencies that use both tablets and laptops.

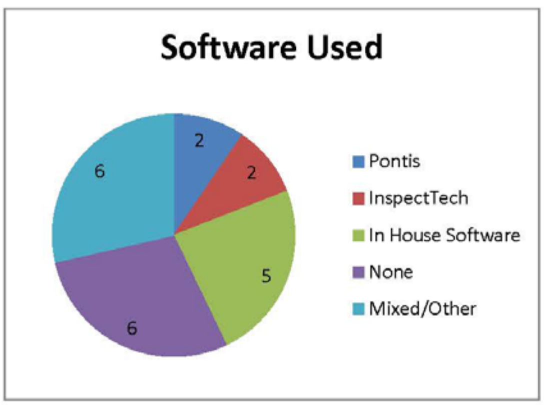


Figure 2: Statistic here represents the types of inspection and management software currently being used. Notice that many agencies use custom software. The mixed/other category represents modified or customized commercial solutions.

Agency	Mobile Devices Use	Software Use	Cellular Coverage
Delaware DOT	Laptops	We use Pontis 4.5 and in house software for photos and supporting documents.	75 – 100%
Florida DOT	None	N/A	< 10%
Hawaii DOT	None	None	75 – 100%
Illinois DOT	None	N/A	10 – 50%
Iowa DOT	20% of NBI inspectors use laptops to review previous records and to enter quantities and photos. Maintenance shops have started using iPads for non-bridge size culvert inspections.	NBI inspectors use InspectTech and culvert inspectors use Fulcrum app	10 – 50%, rural areas are limited, but urban areas are stronger
Maryland State Highway Administration	Laptops (HP ProBook 6560B) Air Cards	InspectTech	75 – 100%
Minnesota DOT	Laptops, iPads, Smartphones	Customized InspectTech	75 – 100%
Missouri DOT	None	None	Estimate in the 50 – 75% range
Nebraska DOT	Laptop Computers	Pontis 4.4.3.BrM 5.2.1	Unknown at this time, but coverage should be good (>75%) with Verizon. We intend to use this technology starting with this current inspection cycle, Using BrM 5.2.1
New Hampshire DOT	Laptops	Pontis	Areas of NH are very remote; each inspection team has "dead zones". Our northern inspection team has coverage in the 10 - 50% range. Our three southern teams have coverage in the 75 - 100 range.

Table 1: Quoted and paraphrased answers from responding State Agencies.

Agency	Mobile Device Use	Software Use	Cellular Coverage
New Jersey DOT	Laptops (Optional) Only for portion of inspections	Commercial (Customized for NJ) Bentley Inspect Tech	10 – 50% (Used to take photos of significant defects to send to office for review and action)
New York State DOT	Laptops	Existing in house software	About 75% cases
North Carolina DOT	Windows Touch Tablets	In House Software (WIGINS)	Never
North Dakota DOT	Laptops may be used but most inputting is done at the office	In house created application	10 – 50%
Ohio DOT	Currently: Ohio is transitioning to a web based system. Future: Ohio will deploy a web based system for all users that can also be used offline in May 2014.	Currently: Inspectors, State and Local, primarily use one of two offline versions that can interface with the mainframe DOS BMS.	We anticipate 50-75% of the state having useable coverage however we expect only 10-50% of the bridges within that coverage
Oklahoma DOT	We do not collect data in the field with electronic devices	None	Unsure
South Dakota DOT	None	None	Unsure
Texas DOT	Laptops	Custom	50 – 75%
Wyoming DOT	WYDOT is currently using laptops for field data collection.	We are currently using software that was developed in-house. However, WYDOT has recently signed a contract with Bentley / InspectTech and will be using commercial software in the near future.	50 – 75%
Utah DOT	Currently Laptops – hope to transition to tablets this summer	Pontis, customized for Utah DOT	50 – 75%
Virginia DOT	Tablets (Motion Tablets, Laptops with rotating screens - old)	Commercial software (write and turn into text, voice recognition, sketching)	0%, not allowed by IT

Table 2: Quoted and paraphrased answers from the responding State Agencies. Note in the fourth column that the Virginia DOT responder stated the cellular coverage was not allowed by their IT department. This example shows some of the ambiguity with this question.

8.4 Requirements Document

Wireless Bridge Inspection System Software Requirements Specification

Michigan Tech Research Institute (MTRI)
3600 Green Court, Suite 100
Ann Arbor, MI, 48105

Prepared for MDOT (Michigan Department of Transportation)
Project MDOT OR14-021 Contract Number 2013-0067, Authorization 2
Project Manager: Richard Kathryns, MDOT

VERSION 1.0 JUNE 2014

Table of Contents

1 Purpose

This Software Requirements Specification (SRS) Document shall contain all of the information needed by a software developer to adequately design the Wireless Bridge Inspection System described in this document. In addition, it delimits the scope of the current project phase and communicates the detailed work plan to MDOT.

2 Product Scope

The 3D MDOT Wireless Bridge Inspection System (3dWBIS), or BridgeView, is a mobile-device application framework enabling the collection of MDOT bridge inspection data for the purpose of generating inspection reports, replacing the current paper-form method. The application serves these core purposes:

1) To allow bridge inspectors to indicate bridge deterioration on the AASHTO (see Definitions) element level;
2) To allow bridge inspectors to precisely locate bridge deterioration within an AASHTO element;
3) To eliminate paper forms and streamline the collection and collation of digital inspection data including field photographs;
4) To provide the bridge inspector with access to previous inspection data; specifically, previous inspection reports.

The application will enable bridge inspectors to enter CoRE/AASHTO element-level inspection data more quickly and consistently than currently allowed. The application will also enable bridge inspectors to access reference data including previous inspection reports and field photographs while in the field.

The goal is to introduce a novel bridge inspection data collection method that streamlines and enhances the element-level inspection process through Inspector interaction with a 3D engineering representation of the bridge under inspection.

3 Current State of the Practice

Currently MDOT bridge inspections are carried out by trained specialist Inspectors who examine all aspects of a bridge, mark up past versions of inspection reports with updated condition state data for a variety of bridge elements and functions, and finalize the current report with data entry at the home office. This is considered the current nominal "use case" for the application. The additional 3D model interactivity will be discussed in section XX.

Field inspection assignments are made via MiBridge (see Definitions) through the Inspection Assignment Dashboard. Candidate bridges for inspection are sorted by their due-inspection dates, nearest due dates first. A team lead is assigned a number of bridges to be inspected that season. Draft bridge inspections can be saved to the BMS but only the originating team lead can edit them.

Routine bridge inspections are conducted in the field by two-member teams. The inspector(s) walk the bridge inspecting and rating the condition of predefined bridge components (elements) while making comparisons to the previous condition of the same as indicated by past inspection reports. Two types of ratings are given, depending on the component: condition ratings, which describe the existing condition compared to the original, as-built condition of the bridge, and appraisal ratings, which describe components in comparison to a new structure built to current standards. During a routine inspection, which is conducted at least once very two years, ratings are given according to criteria specified on three different inspection forms, listed below.

- The Bridge Safety Inspection Report (BSIR), which includes the National Bridge Inventory (NBI) condition ratings
- The Structure Inventory and Appraisal (SI&A)
- The CoRE/AASHTO Elements Inspection (the focus of this 3dWBIS)

On the BSIR form, inspector(s) can see the condition ratings given to each component during past inspections. They also see the inspector's comments on each component from previous inspections. The form is organized by the location of the components into the following groups, in order: Deck, Superstructure, Substructure, Approach, and Miscellaneous. The SI&A form is a complete list of both inventory information about a bridge, which is generally unchanging (e.g. the year a bridge was built), and the appraisal ratings for each element.

The CoRE/AASHTO Elements form lists, for each CoRE/AASHTO element, the quantity of that element (an area or volume) that fall into each of the four condition states. For each of the four condition states, the last recorded quantity (the "old" quantity) can be seen.

An MDOT bridge inspector tends to first fill out the NBI ratings and the CoRE/AASHTO elements condition states; the comments are entered into the digitized inspection form within MiBridge at a later date. In general, if the comments are missing from an inspection form, it is assumed to be unfinished. Notes are annotated in the margins of the printed inspection form while in the field. Notes may be related to particular item/element on the form but general notes about the structure are also written.

4 Concept of Operations for Tablet-Based System

Inspectors will use the tablet to download past inspection data for bridges in their Inspection List. At the bridge site, the inspectors will walk the bridge as before, but enter relevant element-level data into appropriate fields in the tablet application's user interface. The Inspector can navigate to specific instantiation of an aggregate bridge element (eg, a single column of the "piers" class) by interacting with a 3D representation of the bridge under inspection. This 3D model will be rendered in sufficient detail to enable navigation and localization of all inspected elements, though not considered an engineering-drawing level of representation.

The inspector can enter data in any order and navigate to desired elements through menus or by interacting with the 3D bridge representation. Photos may be captured by the tablet device itself and will be tagged to the bridge and element under inspection.

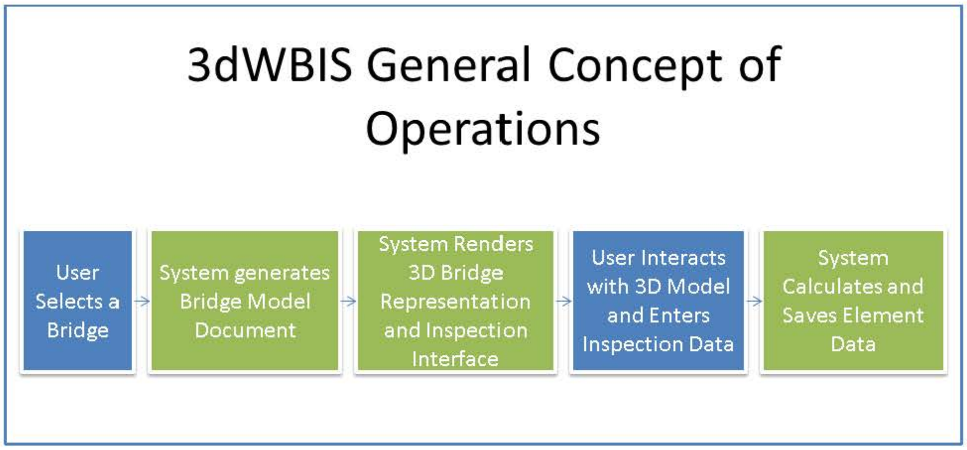


Figure 1: Notional high level workflow for tablet-based bridge inspection

5 General Description of Software

We are developing a mobile application for bridge inspection that will be called the 3D Wireless Bridge Inspection System (also referred to as the "application", "mobile application", "system", or "BridgeView" in this document).

This mobile application will interface with a remote server hosting the surrogate Bridge Management System (sBMS), a relational database management system (DBMS) that will stand in for MDOT's Bridge Management System (BMS) during software development and testing. The interface between the mobile application and the sBMS is a web application programming interface (API) and consists of HTTP requests. API requests will be received by a server on which the database middleware and database management system (DBMS) is running.

The bridge under inspection will be represented as a 3D rendered object that the Inspector can interact with in order to indicate the location of, and tag inspection data to, condition states of individual bridge elements.

The software components that will be developed by the Michigan Tech Research Institute include and are limited to:

- The Wireless Bridge Inspection System tablet application, including the 3D Bridge Rendering Engine
- The surrogate Bridge Management System (sBMS) database
- The Server application that mediates between the sBMS and the Tablet application, including the Bridge Model Schema Generator
- The web-based API
- The Bridge Model Schema

The software components that are already developed and which will be selected from among a number of appropriate alternatives are:

- The database management system (DBMS)
- The database middleware

The Wireless Bridge Inspection System application will be used in the field by bridge inspectors for the entry of MDOT CoRE Elements/AASHTO Elements inspection data. It will also allow inspectors to view the same inspection data for previous inspections. The inspection data as a whole from one of a bridge's previous inspections can be used to populate the form for a new inspection of the same bridge, where appropriate.

A new functionality that we believe unique to the industry is the rendering of a 3D representation of the bridge for the Inspector to interact with from a query to the bridge database combined with bridge design rules.

This model is the basis for interaction with Element-level inspection components, indicating localization of conditions on individual instances of an element (for example, a specific pier column or slab). The Inspector enters quantitative values for the individual element condition states, and the Application performs the aggregation over element to provide the final percentage or ratio per element as currently provided in the Element level forms.

However, the underlying granular state of each instance is saved for future reference and Inspections.

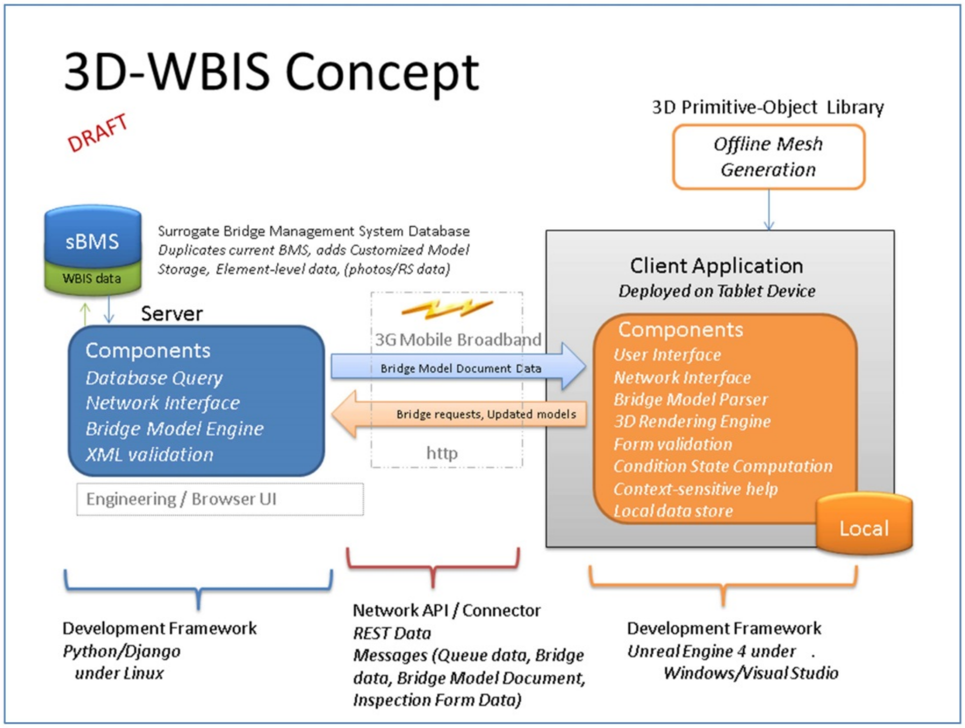


Figure 2: Wireless Bridge Inpsection System Concept

The data entry process will be very similar to the established process in that the entry of inspection data will follow the typical order of MDOT inspections (i.e. as the inspector "walks the bridge"). However, the user will also be able to enter inspection data "out of order," if desired. Unlike existing inspections, where the inspector must total the deterioration in each condition state for every AASHTO element, the application will do this automatically for the inspector as new areas of deterioration are found and entered into a condition state.

The application will be run on tablet computers, specifically on Android or iOS tablet computing platforms. For the Phase I (first year beta version), the Android operating system is the deployment platform for initial development and testing. Users will connect to a remote server for uploading completed inspection reports and for downloading past inspection reports and bridge models. The application will be available whether the mobile device is connected to the internet and able to reach the remote server or not. In the latter case ("offline mode"), inspection data collected by the user can still be entered in the application. Inspection reports will be saved to the device until they are completed and ready to be uploaded. Completed inspection reports that were previously downloaded for reference will also be available offline. While offline, the completed inspection reports will not, of course, reflect any changes made to them on the remote server by another user.

Field photographs and images from remote sensing technologies will be also available through the application. However, metrics from remote sensing surveys and projected remote sensing imagery will not be available. Images can be annotated by the user with comments that will be saved and synced with remote storage.

Users will be able to create and edit new inspection reports saved on their device until they mark the report as formally completed and upload it to the remote server (sBMS). Once the completed inspection report is uploaded, changes can no longer be made through the Wireless Bridge Inspection System. Users can view completed inspection reports including those they have just uploaded as completed. However, these completed inspection reports are available for viewing only (i.e. they are "read-only").

The application will also provide quick-reference materials for bridge inspectors including the AASHTO standard Rating Guides.

The Wireless Bridge Inspection System will annotate bridge inspection reports with automatic metadata as the reports are created, edited, and ultimately submitted to the remote server (sBMS). These metadata will be limited to:

- When a field in an inspection form was changed
- The last value of a field before it was changed
- The date, time, and submitting user identification for submitted inspection reports
- The location and orientation of the mobile device for each field photograph taken

5.1 Product Functions

The WBIS will fulfill the functions listed below.

- Emulate or improve the entry of CoRE/AASHTO element-level inspection data (via User Interface)
- Require users to authenticate themselves before submitting data to a remote server
- Provide data from previous inspection reports, including photos and element ratings
- Provide access to inspection manual information
- Provide limited quality assurance (QA), such as ensuring all fields are filled out, prior to finalizing a report
- Provide the ability to attach photos to the report

The 3dWBIS will make data entry easier by providing:

- An intuitive graphical way to navigate to Bridge elements

- A systematic way to visualize past condition state data local to a particular element instance Lookup data such as substitutions for enumerations, abbreviations, or codes (e.g. "5" means "steel")
- Automatically tallying quantities of condition states for element level data
- Drop-down selectors for fields with a defined set of valid inputs

5.2 What the WBIS will not do

Data from scopings (see Definitions), waterway surveys, and other non-routine inspections will not be entered into the WBIS. The WBIS will not provide for the entry of "routine BSIR" inspection data such as the routine National Bridge Inventory (NBI) inspection data or Structure Inventory and Appraisal (SI&A) data. The WBIS will not allow for work recommendations to be filed. The WBIS will not provide any level of decision support such as deterioration rate estimates, bridge life-cycle or asset management products. The application will not enable mobile devices to take any direct contact or non-contact measurements of the bridge (other than photographs). The application will not track the location or behavior of the inspector in the field beyond metadata attached to inspection reports as defined above. The application will not provide any means of communicating between inspectors in the field or between inspectors and office personnel.

5.3 User Characteristics

The Users of the WBIS are expected to be qualified MDOT bridge inspectors (or contract assignees) with a proficient level of experience with current (2013) MDOT routine bridge inspection practices, or in training for such, or other supervisory personnel with experience with current (paper form) practices.

In addition, the User is expected to possess a basic understanding and familiarity with the operating system (OS) of the target hardware platform (eg, Apple iOS, Android operating system), including but not limited to opening a native application, opening stored or downloaded documents, native navigation actions (home, menus, control panels), and data entry using a touchscreen and/or virtual keyboard.

5.4 General Constraints

The application is constrained by the input methods of the device. Without external hardware, such as a Bluetooth keyboard, all of the candidate tablets use a touchscreen for input. While a stylus may be used with a touchscreen, the application should not assume its presence. The application cannot assume the presence of any input methods other than a touchscreen.

The dimensions and resolutions of the candidate tablets vary, but they are, in all cases, limited. Even on devices with very high resolution displays, such as an iPad with a Retina display, the limited size of the screen will limit the amount of information being displayed at one time. In this regard, standard approaches to graphic design and typography must be used to ensure readability in all conditions by all users.

Each tablet contains different hardware features (cameras, GPS, etc.) with varying specific capabilities. The application cannot assume the presence of every hardware feature, and the behavior and performance of these cannot be guaranteed across platforms.

The application requires a network connection to access the bridge inspection database, but users will frequently be without such a connection. The application cannot assume the presence of a connection to function properly. Similarly, the application cannot assume that the availability of a network connection implies the availability of the database. If the database is unavailable, the application will only be able to store reports locally.

5.5 Assumptions and Dependencies

5.5.1 Application Platform Assumptions

The Wireless Bridge Inspection System (WBIS) must be portable and accessible in the field under varying weather and traffic conditions. The intent is to provide an application which, when paired with suitable hardware, will provide little to no encumbrance to the user. A potential standard for comparison is any leading mobile application that has achieved great popularity and adoption in the mobile market. The platforms for today's mobile applications generally offer similar hardware support (e.g. 5-20 megapixel camera) but are significantly different in terms of the operating system. There are a small number of operating systems in wide use on so-called "smartphones" within the United States; in descending order of 2014 Q4 market share (according to Kantar Worldpanel[1]), they are: Android (50.6%), iOS (43.9%), Windows Phone (4.3%), and Blackberry (0.4%). Market share estimates for tablets are harder to come by, but PC Magazine found that, in a 2012 survey (http://www.pcmag.com/article2/0,2817,2405972,00.asp), 52% of tablet owners possessed an iPad (iOS) while 51% of tablet owners possessed an Android model (these estimates do not add up to 100% as they include users who own more than one device type).

The primary considerations in selecting platform(s) for the WBIS software specifications are: 1) Maximizing platform availability for the application as measured by market share; 2) Ensuring that specific end-user devices are supported; and 3) Providing a consistent user experience across platforms. With regards to these considerations, the development team proposed to support the Android and iOS platforms, which combined represent 94.5% of the smartphone devices and each roughly half of the tablet devices currently on the market (2013 Q4 and 2012 estimates, respectively). Developing for these platforms should provide for a consistent user experience and will provide the end-user and customer, the Michigan Department of Transportation, with the fewest constraints on hardware selection and purchase due to the wide variety of choices afforded by these two platforms.

We assume that the target mobile device platforms will have the hardware necessary to support functional requirements, specifically:

- The mobile device is running either the iOS or Android operating system in good working order
- The mobile device will have a camera that can be activated by applications on the device
- The mobile device has a wireless broadband connection (either 3G or 4G) and a cellular data service plan
- The mobile device will only be used within the United States of America
- The mobile device has a touch screen and a virtual keyboard
- A GPS receiver is available on the device and installed applications have access to it

5.5.2 Bridge geometry assumptions

[1] http://www.kantarworldpanel.com/dwl.php?sn=news_downloads&id=399

Regarding the rendered bridge geometry, for the first version of the Application, "skew" is ignored; that is all bridges are rendered "orthonormal" with 90 degree angles between joints and spans, piers perpendicular to the carried roadway, and no diaphragm offset.

The model schema is being designed to accommodate calculations deriving from skew, but we are ignoring skew for simplicity in this iteration.

Also, due to the fact that there are no indicators in BMS for any asymmetries, bridges are rendered as longitudinally and laterally symmetric based on "longest span length." Many variances from symmetry can be achieved through the Bridge Customization step that is available when a user first visits a bridge.

6 Specific Requirements

6.1 System Interfaces

6.1.1 User Interfaces

The WBIS mobile application will facilitate the entry of CoRE/AASHTO inspection data through a virtual model of the bridge under inspection. This 2D or 3D model will allow the user to identify and select elements of the bridge based on touching or tapping the bridge model as it appears on a touchscreen. After selecting a CoRE/AASHTO element in this manner, an "enhanced presentation" (e.g. zoomed-in and/or panned view) of the element will allow for the delineation of spatially-explicit deterioration information by touching or tapping on the affected area of that element. The overall model can be rotated through constrained views using either direct touch manipulation or controls.

Elements can also be located through contextual menus including a search utility where matching results are filtered to a keyword search for the element's name in real time. Selecting an element in this manner will present the user with the "enhanced presentation" of the model. The alternate "enhanced presentation" may include finer detail than the overall model, allowing for the user to precisely locate a certain area of the element, and will likely present a constrained or predefined viewing geometry.

In the "enhanced presentation," users can tap or touch an area of the element to enter information about the dimensions and condition state of that area. This tap or touch interaction will launch a modal or space-filling window that describes the data to be entered, including the approximate shape of the area, the dimensions of the shape (which could be given in units of area or as the percentage of the overall element), and the condition state that area should be associated with. In all of the data entry interfaces, fields that have been changed will be styled in a certain way so as to indicate to the user that the field has changed (marked as "dirty").

The virtual model of the bridge will be initialized from SI&A element-level information about the bridge deck, substructure, superstructure, approach, and railings; such information presented by the Bridge Model Schema. Other data is not available from the BMS database; namely fascie widths, pier/column shapes & diameters, beam shapes, pin and hanger locations (if any), and bearing locations (if pin & hangers) are used. Reasonable defaults based on the domain knowledge we have captured from MDOT and Tess Ahlborn at MTU will be used to render the "Initial Bridge Model."

Additional user input will be required to customize the virtual model, however. User input will be gathered through a series of prompts which initialize to defaults based on best assumptions. After this is done once for any bridge, the user will be prompted only to accept or reject these settings upon viewing the same bridge again; rejecting these settings will require the user to enter the missing information required to render the virtual model of the bridge.

6.1.2 Hardware Interfaces

6.1.2.1 Specific Device Compatibility Assumptions

After the process of selecting UnrealEngine4 as our application deployment framework due to its 3d capabilities and User Interface extensibility, we learned that there is a reduced set of compatible Android devices as specified by the UnrealEngine4 developers. This list is growing weekly as the community tests deployment to various tablet platforms.

As of June 2014 the following table lists the developer-certified compatible chipsets. We also list specific devices which meet this requirement, as well as a cellular data capability and minimum memory and speed requirements.

Table 1: Table of Android device compatibility from UnrealEngine4 (the 3D Rendering Framework for the Tablet Application)

Unreal Engine 4 Documentation > Platform Development > Android Game Development > Android Device Compatibility — Jump To

Android Device Compatibility

Android support is still in its early stages and we have not yet tested on a broad range of devices. We plan to expand and refine this section as new devices are released and we broaden the devices we are testing in house.

These tables reference feature tiers as described on the Performance Guidelines for Mobile Devices page.

- Supported - We have tested the family of devices here and expect them to perform well.
- Expected - We have not tested the family of devices extensively but expect them to perform well.
- Unsupported - We do not expect the device to perform well for the feature tier.

The following table lists common GPU families.

Device	LDR	Basic Lighting	Full HDR	Full HDR w/ Sun
Tegra4	Expected	Expected	Unsupported	Unsupported
Adreno 320	Supported	Supported	Expected	Unsupported
Adreno 330	Supported	Supported	Supported	Supported
Mali 400	Expected	Expected	Expected	Expected

The following table lists individual devices we have tested here:

Device	LDR	Basic Lighting	Full HDR	Full HDR w/ Sun
Galaxy S4 (NA, Adreno 320)	Supported	Supported	Expected	Unsupported
Nexus 5 (Adreno 330)	Supported	Supported	Supported [1]	Supported [1]
Kindle Fire HDX (Adreno 330)	Expected	Expected	Expected [2]	Expected [2]

1: The Nexus 5 with the latest publicly available driver performs poorly when using features from the Full HDR tier. We have been working closely with Qualcomm in this area and they have developed faster drivers that remove the bottlenecks we were running in to. Our HDR features are fully supported on their latest internal drivers which we hope will be available to the public soon!

2: Similar to the Nexus 5, the Kindle Fire HDX runs in to some bottlenecks in the Adreno 330 driver and we expect it to perform well in the future with an updated driver.

Note that there are more Adreno devices than those listed in the second table; those represent the ones UE4 has tested inhouse.

Below is a table which identifies compatible tablet devices, as of June 2014.

Table 2: June 2014 listing of recommended development and "beta" Android devices

Device	GPU	Data	Screen size	RAM/ROM/ Storage
Xplore RangerX Rugged Tab	Integrated PowerVrSGX544 w/ ISP	3G	10.1	1GB (?)/ 32G ROM/ 192GB
Samsung Galaxy Tab SM-t325 4G Lte (Tab Pro 8.4)	Adreno 330	4G	8.4	2GB/ 16GB ROM/ uSD 64GB
Samsung Galaxy Note Pro 12.2 (MSM8974AAv2)	Adreno 330 450MHz	3G/4G	12.2	3GB/ 32/64GB/ uSD 64GB
Samsung Galaxy Tab Pro	Adreno 330 450MHz	3G/4G	10.1	3GB (RDRAM) / 16/32 / 64GB
Fujitsu Arrows	Adreno 330 450MHz	3G/4G	10.1	2GB / 64GB / uSD (?)
Amazon Kindle Fire HDX 8.9	Adreno 330 450MHz	3G/4G	8.9	2GB / 64GB / none(?)
Sony Xperia Z2 MSM8974Ab v3	Adreno 330 578MHz	Futremark says "Varies by model"	10.1	3GB

The two recommended devices are the Xplore RangerX series of ruggedized tablets, with obvious field advantages but higher cost. The second recommendation is the Samsung Galaxy Tab Pro line, a top consumer product with superior screen brightness and required graphics chipset.

6.1.3 Communication Interfaces

The mobile application will communicate with the remote server, nominally the surrogate Bridge Management System (sBMS) or canonical Bridge Management System (BMS), over the web through normal HTTP requests. Each request for retrieving from or submitting data to the server will therefore take the form of a Uniform Resource Identifiers (URI). The practice of implementing and using this type of communication interface is sometimes referred to as Representative State Transfer (REST). When requests originate in the Secure Socket Layer (SSL) over HTTPS, these requests and responses are securely encrypted between the user's device and the remote server.

The REST API will describe completely the available data operations that can be conducted between the mobile application on a tablet in the field and the sBMS/BMS on a remote server. Operations such as retrieving past inspection reports or uploading completed inspection reports will each have a unique URI through which a connection is made between the mobile application on a tablet device and the remote server. Documents or data objects available through the REST API are generally called resources. Some of the resources will be read-only whereas others will be resources that can be created on (uploaded to) the remote server.

6.2 Functional Requirements Summary Table

The following table lists the major capabilities of the 3dWBIS System.

Subsequent versions of this requirements document will expand each requirement

REQ.	Requirement	Description
3.2.1	System presents user with 3D representation of bridge under inspection	Based on a bridge model schema encoding a plurality of expected highway bridge types, a 3D version of a specific bridge model is displayed for interaction, with sufficient detail to enable inspection and markup of all AASHTO bridge elements
3.2.2	User can interact with model using native (touch) controls as well as intuitive navigation controls	Model views can be manipulated (in a set of constrained views) such that all AASHTO elements are accessible
3.2.3	User can view major bridge element groupings	Bridge elements are grouped into Superstructure, Substructure, Deck,
3.2.3	User can select individual AASHTO elements for markup	AASHTO element will be isolated or 'zoomed' in an 'enhance view' in order to expose representations of physical attributes that are rated by inspector
3.2.4	User can indicate spatially localized data within a single instance of an AASHTO element	Graphical 'indication' of inspected locale on AASHTO element via native interaction with 3D model element by touching approximate location on representation
3.2.5	User can enter text data and associate it with an AASHTO element or spatial subset of an AASHTO element	AASHTO element fields can be accessed from either a listing drop-down type input field or the graphical-object navigation described in 3.2.3
3.2.6	System aggregates individual ratings across multiple instantiations of elements for the reported percentage condition state rating	In cases where AASHTO elements represent a collective entity with a single reported rating (eg, piers), the user enters individual quantities on a per-instance basis, and the system adds the quantities for each instance to a collective value. The user must enter the quantitative "amount" value for the condition state (eg, square feet).
3.2.7	System generates a AASHTO CoRE Element Report	System aggregates and summarizes element-level condition state data to populate the familiar Element Inspection Form
3.2.8	User can take photos and attach to an inspection event	Native tablet camera hardware can be used to take photos at the bridge site during inspections. These photos are "linked" to the Bridge / Location / Element and Date of the Inspection Event, and can be retrieved and viewed by Inspectors at any time. The

		updated BMS database will store bridge-tagged photos.
3.2.9	Users can view historical data	Element level data and comments from previous reports are accessible in the Application

6.3 Use Cases

6.3.1 Prototype Use Case

Structure number 10922, S13-81003: Curtis Road over M-14

1. Inspector selects bridge by structure number
2. System requests data from server
3. Server generates Bridge Model Document
4. Server sends Model.xml and (additional bms data) to System
5. System validates document
6. System renders 3D Bridge Representation
7. User customizes model
8. User inspects bridge in "walkthrough" mode
9. User enters condition states on a per-element-instance basis
10. System calculates overall (aggregate) condition states
11. System generates Report
12. System posts AASHTO form data to sBMS
13. System saves 3D element data and updated XML model

6.4 Quality Requirements

6.4.1 Performance

Transactions between the application and the remote server will be performed in the background (asynchronously) whenever possible. In such cases, the wait time experienced by the user will be zero. Actual wait times will vary with aspects of the wireless broadband connection (e.g. 3G versus 4G, signal strength).t

6.4.2 Reliability

- Recovery from software or hardware failure must be robust to protect the time investment of inspectors
- Inspection reports will be automatically saved ("autosaved") to the device
- Additionally, inspection reports will be automatically backed up to the remote server (sBMS or BMS)
- The application software should reasonably reduce CPU usage to promote long battery life in a mobile environment.

6.4.3 Availability

It is expected that the tablet application be available regardless of the presence of WiFi or Data network. With the exception of downloading "new" bridge data, all functions of the 3DWBIS will be available for the user, including performing and saving a Bridge Inspection and taking photos. These data will just not be saved to the remote server until connectivity is reestablished.

6.4.4 Security

The WBIS will require users to authenticate themselves through a unique username-password combination. These login credentials will be cached on the device for a fixed period of time (e.g. 24 hours) and used to authenticate requests to submit or receive data from the remote server. The login credentials will be encrypted before being sent in any request. The remote server will compare the encrypted credentials against those stored in its secure database and, if they do not match, will reject the associated request. While the WBIS will not be demonstrated over HTTPS, should the customer later decide to implement HTTPS, the software design as elaborated in these requirements will not preclude its implementation.

6.4.5 Maintainability

This pilot software system as designed and implemented by MTRI is considered in beta test after the initial year of development. MTRI will be the sole point of support for the system front end (Tablet Application) and back end (Server / Database Interface) until the close of the current project phase. MTRI will work closely with MDOT IT/Database personnel to propagate the database extensions and middleware functionality to an MDOT in-house capability by close of project.

6.4.6 Portability

The final version of the application will look and feel the same, to the greatest extent possible, across devices. Specifically, the application's user interface, user interaction, and behavior will not vary between the Android and iOS implementations.

The beta (or first year version) will only be supported on Android devices, due to the resource constraints imposed by developing a native 3D application that were not foreseen at the start of the project.

6.5 Design Constraints

The development team at the Michigan Tech Research Institute does not have access to the Oracle license required to host and manage an Oracle enterprise database such as the existing Bridge Management System (BMS). As the development team desires to work with a representative database during active development and for testing purposes, the development team will need to create a surrogate Bridge Management System (sBMS). The sBMS is intended to mirror the BMS database schema but also to extend it where necessary to support new functionality (e.g. the storage and retrieval of field photographs). The sBMS will be contained by a PostgreSQL database management system instance. In designing the database middleware, the development team will select software that is known to work or will be made to work with both PostgreSQL and Oracle databases.

6.6 Logical Database Requirements

As the Wireless Bridge Inspection System is intended to read from and write data to the existing Bridge Management System (BMS), the back-end database required to support the application is a relational enterprise database. The surrogate Bridge Management System (sBMS) will be a relational database modeled after the existing BMS database schema. The database management system (DBMS) selected for this purpose, PostgreSQL, is an enterprise-level DBMS that along with the existing BMS supports the following features thought to be necessary to support the software requirements:

- ACID compliance (atomicity, consistency, isolation, durability)
- User and role permissions at the object level
- Triggers
- Procedural languages
- Spatial object support

7 APPENDIX A

7.1 Definitions

Term	Definition as used in this requirements document
AASHTO	The American Association of State Highway and Transportation Officials (AASHTO) is a standards setting body which publishes specifications, test protocols and guidelines which are used in highway design and construction throughout the United States.
Appraisal rating	During a routine inspection, a rating that describes the condition of bridge components compared to a new structure built to current standards
Bridge Model Schema	A project-defined data representation of a bridge that is specific, complete, human-readable, and allows the Renderer to "draw" a 3D representation of the bridge. Implemented as an XML document.
Condition rating	During a routine inspection, a rating that describes the existing condition of in-place bridge components compared to their original, as-built condition; this includes the 4 NBI Condition Ratings
CoRE Elements	"Commonly Recognized" Bridge Elements as defined by AASHTO
Extended inspection forms	All other forms currently in the Bridge File, namely Fracture Critical, Fatigue Sensitive, Underwater, Other Special, Damage, Scour Action Plan (which may be supported in future)
MiBridge	A web based structure management application allowing Bridge Owners, Engineers, Inspectors, Consultants, and Managers to view and enter information for bridge and culvert assets across the State of Michigan.
Previously downloaded report	A PDF read-only version of a previously submitted Report, as from the MiBridge "Bridge File" document
Routine inspection	The inspection performed per-bridge at least once very two years (but sometimes more frequently) that includes the CORE Element Inspection, the Structure Inventory and Appraisal, and the NBI
Scoping Survey	Evaluating a bridge for various repair alternatives and recommending the most economical rehabilitation or treatment, then developing a scope of work and cost estimate for the selected alternative
Standard inspection forms	The three MDOT forms whose inputs are supported by this WBIS; the three "main" forms known as CORE Element, Bridge Safety Inspection Report, and Structure Inventory and Appraisal Forms
3d Bridge Model	A 3D representation of the bridge to serve as a method of

	displaying and navigating to individual CoRE bridge elements. It is not meant to be a solid model of the bridge, nor an engineering facsimile, but a manipulatable analog of the bridge that the Inspector can mark up, localize conditions on, and attach photos to.

7.2 Coordinate System for Bridge Rendering

The Application maintains an internal coordinate system for rendering the bridge elements (parts) in the correct relative position and relation to each other. The origin of this local reference frame is one of the (symmetrical) bottom-right corners of the deck surface, with the X direction following the span-direction or on-bridge traffic direction, and the Y direction in the lateral, with positive Y values starting on the bridge. These coordinates are not typically used by the inspector, but provides the Application with a consistent rendering of bridges in a local reference frame.

If the default labels are incorrect, the Inspector will have the opportunity to update them during an inspection action.

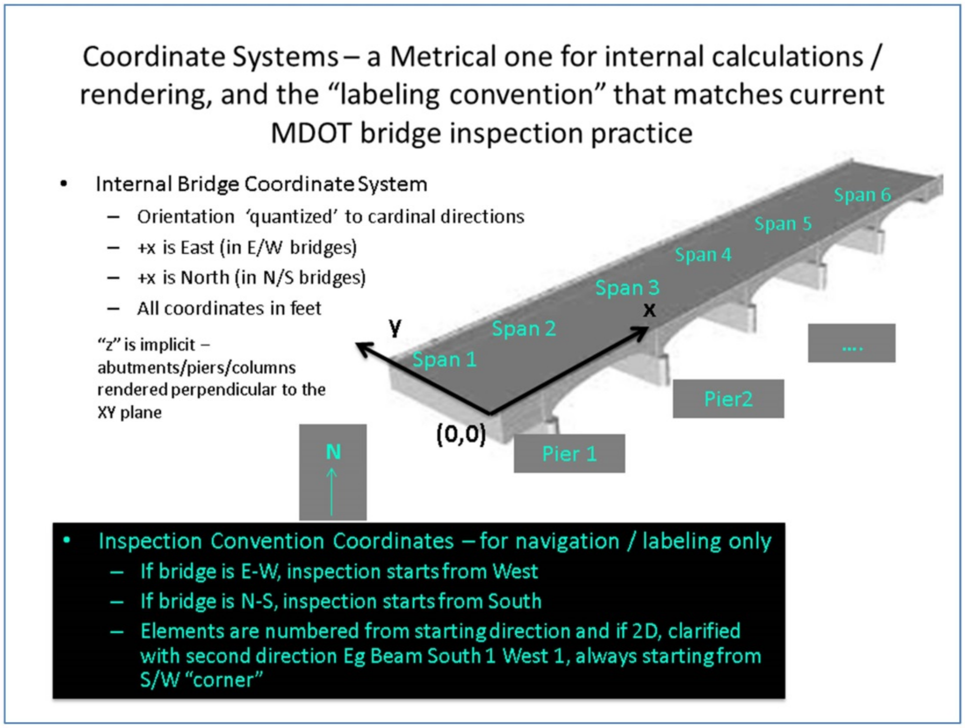


Figure 3: Coordinate System for Bridge Rendering

8.5 Implementation Action Plan

Project Title: Wireless Data Collection Retrievals of Bridge Inspection/Management Information

Project Number: 2013-0067, Auth. No. 2 (R1, R2)

Principal Investigator: Colin N. Brooks

Description of the Problem:

Currently, MDOT is faced with the task of inspecting its entire bridge inventory using a paper form process. Considering that every bridge must examined every two years, or more frequently as condition demands, this is a time-consuming process. The size of the task has only increased in recent years given the FHWA's demands for compliance with the National Bridge Inspection Standards by checking the completeness and accuracy of bridge data. The collection of AASHTO element-level data further increases the demand on inspectors.

To meet and exceed the requirements of the new regulations, MDOT is interested in incorporating mobile digital technology into the bridge inspection process to increase its efficiency and reliability. By switching to a digital inspection process, MDOT will eliminate the need for manual data transcription from paper to digital. The fact that most mobile platforms come equipped with a built-in camera means that MDOT can streamline the process of associating images with defect information, which otherwise must be done manually.

To facilitate the collection of element-level data, MDOT wishes to use interactive 3D bridge models that the inspectors can mark up with defect information. The inspection program will then automatically perform tallying to generate quantity information for element-level reporting. Additionally, an interactive 3D format for bridge inspections lends itself to bridge management decision support, since it provides detailed information about defects and their location on a bridge that is not captured by element-level reporting alone. Tracking individual defects also provides information on how those defects deteriorate over time, further aiding in management decision-making.

Major Discovery:

During this project, MTRI developed a mobile application (the 3D Bridge App) for displaying and interactively marking 3D bridge models with element-level defects. The application was built using Unreal Engine 4, a cross-platform game engine that allows the application to be deployed to a variety of operating environments including Windows, iOS, and Android. The application automatically tallies condition state and defect quantity information, freeing the inspector from that burden. The captured bridge inspection data can be transmitted wirelessly to MDOT to be stored for bridge management purposes.

Additionally, using information from MDOT's BMS database, MTRI developed a system for generating representative 3D bridge models for common concrete bridge construction styles. While these models may not be perfect, in concert with user tuning they are designed to be more than sufficiently useful for enabling inspectors to recognize the bridge structure and intuitively interact with it.

How the Application Will Be Used by MDOT

MDOT has the option of using the application to revise the current state of practice for bridge inspections. Initially, this use should consist of an implementation-focused trial period in which a few interested inspectors attempt to utilize the application in day-to-day operations. MDOT should use the trial period to identify challenges that will be faced in deploying the application statewide, and to formulate improvements for meeting those challenges. Potential problems could include but would not be limited to hardware issues (battery life), errors in the application itself that were not apparent during development, integration issues with the MDOT BMS, and deficiencies in user training. Discovering and correcting these issues during a trial phase would be critical for long-term success in revising MDOT practices, as a rocky deployment could burden an otherwise useful tool with a poor reputation. Conversely, a smooth deployment could improve adoption rate and support further development. The MTRI team would work with MDOT to address issues discovered throughout the implementation trial period.

During the trial phase, problems that actively disrupt the inspector's workflow should be addressed immediately. Continuing the trial without correcting disruptive problems could mask other issues, rendering the ongoing trial usage ineffective. On the other hand, smaller obstacles such as unintuitive interactions and cosmetic flaws should be documented for later evaluation and correction as time and funding permit. Trial participants should be informed of this differentiation so they do not become fixated on perceived flaws that do not reduce the overall functionality of the application. This is especially critical since different inspectors may have different opinions as to what the "best" approach is. Continually revising noncritical functionality will squander the time and effort spent on the trial implementation, distracting from the identification of more serious issues that could be barriers to successful deployment.

At the conclusion of the trial period, MDOT should evaluate the feedback generated by the trial participants. Ideally, any key issues will already have been solved, but if they have not, this is the time to evaluate the readiness level of the application. If the key issues identified cannot be fixed quickly, the application may need further development and another trial implementation before it is ready for wider distribution and full introduction into day-to-day usage. If the application completes the trial period without such setbacks, MDOT can evaluate the trial participants' feedback and make a final decision about deployment.

Value Added to MDOT Operations

As a digital application, the 3D Bridge App eliminates the need for the paper forms currently used by bridge inspectors. This allows them to perform any number of inspections without access to a printer. Since inspection information is ultimately stored in the BMS database, the application also eliminates the manual data transcription process, which is costly in terms of time consumed and is an additional source of potential error in the transcribed data. Additionally, the application has great potential to streamline the inspection process, improving inspector efficiency and accuracy in the field. This is especially the case since the application allows inspectors to capture AASHTO element-data in an intuitive manner (by marking the bridge model with defects as the inspector observes them in real time). Further, the application automatically aggregates the information for reporting, freeing inspectors from that burden. The application also is able to display context-sensitive information concerning the inspection process, such as the condition-state guidance tables from the Michigan Bridge Element Inspection Manual. Integrating those tables into the application allows the inspectors to quickly verify their choices without flipping through the physical manual.

As a digital platform for bridge inspections, the application offers a wealth of new opportunities in the future. Integrating more-detailed bridge models, such as those used during the construction of the bridge, could better facilitate lifetime management of infrastructure. Additionally, the rendering capabilities of Unreal Engine 4 could be leveraged to display remote sensing data as overlays on the 3D models, aiding both inspections and management decision-making.

To enable MDOT to take advantage of the full value of these mobile app technologies and the investment made in the 3D Bridge App, the Michigan Tech team has recommended four tasks for a potential third phase of the project. These tasks would focus on implementing and deploying the application into day-to-day usage at MDOT. They are as follows:

1. Integrate the 3D Bridge App with MDOT's database using the current version of BrM. Currently, the application will save inspections as XML files that are output to specific locations on the tablets. To integrate the application with MDOT's database, all of the information within the XML would need to be uploaded to MDOT's database.

2. Update key features identified by MDOT to make the application more user-friendly and the bridge inspector's job even easier. MTRI has recorded all of the suggestions

made throughout Phase II of the project, and suggests MDOT look at some of these for potential incorporation (see Appendix 8.6).

3. Use the 3D Bridge App for a wider set of bridges. Current implementation focuses mostly on generic highway overpass bridges in Michigan. More detailed models could be created by rendering non-generic bridge elements and improving material and mesh fidelity (material is how the model is ‘painted’, the mesh refers to the geometry itself). to help mimic reality. MTRI would ensure that the application creates models for the majority of bridges that are accurate enough for use.

4. Perform alpha and beta tests to bring the 3D Bridge App to the point of deployment into day-to-day usage.

Implementation Plan Checklist:

Results achieved through this research (check all that apply)		Actions needed to implement results (check all that apply)	
X	Knowledge to assist MDOT	X	Management decision
X	Manual change	X	Funding
	Policy development or change	X	Training
X	Development of software/computer application	X	Information technology deployment
X	Development of new process	X	Information-sharing
X	Additional research needed	X	Other (specify) implementation-focused trails and database integration.
	Project produced no usable results		
	Other (describe)		

8.6 List of Possible Future Developments for the 3D Bridge App

(The development times noted in parentheses are estimates; some tasks may take more or less time than expected.)

Features to Integrate (Short Amount of Time)

- Develop a “Home View” button that when clicked would reset the camera to a set location so that users know where they are. Would help if the user gets “lost” within the 3D environment.
- Develop common views in addition to just navigating using the pinch/slide method. Common views include looking down at deck, looking up at deck soffit, right and left elevations, and front and back face of substructure.

- Develop square and round columns through user tuning.
- Render an approach slab for every bridge. (Dimensions are not in the database, but MTRI could hard-code them, so inspectors could record approach slab information.)
- Show length and width of a defect instead of total area and aspect ratio.
- Develop a button that would give directions to the bridge through Google Maps.
- Show bridge name somewhere on the screen.
- Show time stamp in the corner.
- Develop an exit-without-saving button.
- Develop zoom capabilities in defect editor mode.

Features to Integrate (Medium Amount of Time)

- Develop an Orientation Viewer—Have a side button named "Viewer" to assist in orienting the inspector to the bridge. Once the user clicked this button, a list of every individual member would be displayed and organized by element, span, bay, etc. If a user clicked the individual member, the camera would be placed in a position for viewing that member. A rough example is shown below.
    - Beams
        - Span1
        - Span2
        - Span3
            - Beam 3S 1W
            - Beam 3S 2W
            - Beam 3S 3W
            - Beam 3S 4W
            - Beam 3S 5W
        - Span4
        - Span5
- Develop a label schema that can be toggled on and off with a button to clearly label every individual member.
- Add the option to enter defects according to the strict unit reported in the MBE rather than surface area only. For instance, select all of an element when reported by "each", For instance with bearings, users would not want to highlight one face of a bearing; the whole bearing should be bad. Or, when a beam end is bad, since beams are reported by linear feet, then it would highlight the entire surface area of the beam for the length of the beam that is bad. For most elements it is useful that there is the option (columns for instance), but many users would want their inspectors to match.

- Have a view-only mode, especially on desktop computers, where inspectors could view inspection models but could not edit any information.
- Develop a copy-and-paste functionality for defects.
- Limit defect dimensions to the dimensions of the element it is attached to
- Limit the defect total quantity so that it cannot exceed the total quantity of the element. Currently, users can make the defect as big as they want.
- Require inspector to take a minimum number of photos before pushing the data (i.e., sending the data to the application's back-end for storage).
- Implement different materials to simulate concrete, steel, etc.
- Develop capability to draw over pictures taken with camera.

Features to Integrate (Long Amount of Time)

- Have the application use GPS to allow for the inspector to be better oriented. (What happens if GPS gets disconnected? How reliable will GPS be under a bridge?)
- Pinch-to-rotate view as in Sim City game. Camera could be confined to a region, as opposed to a rail. (Camera rails are easy and can be made to help the user avoid getting lost when navigating around a bridge in the app.)
- Draw a defect in defect editor mode.
- For cracks, draw the crack by setting a series of points, which the application would then connect to draw a line.
- Implement customized elements—splayed spans, curved girders, beam shapes, box beams, T-beams, straight beam curved decks, etc.
- Integrate model into Google Maps to overlap with Google Maps' version of the bridge based on the latitude and longitude coordinates, as some CAD models are able to do.
- Add voice-to-text for comments, or ability to write in comments with stylus; this is especially useful when inspectors are on the deck and have to watch out for traffic.
- Have the compass reflect the actual direction the inspector is facing.
- Render pin-and-hanger assemblies as well as diaphragms (need user input).
- Create decals for every individual defect to better reflect what each defect looks like instead of showing the defect as a rectangle or circle.
- Add spell-checker feature for comments.
- Toggle protective system or coating from whatever element the coating is on.
- Make the deck transparent to see defects on the top or bottom surface of the deck.
- Integrate CAD models into the app, and also integrate map metadata into the CAD model.
- For reviewing past inspections, highlight old defects that have not been reviewed by the inspector.

- Develop 3D models focused on design, not operations and inspection/maintenance (add in areas on operations and inspection/maintenance).
- Need top/bottom layers to show corrosion above bottom spalls.
- Use camera as recording device to take video, or use unmanned aerial vehicles to take video or pictures.
- Add skew to the bridge model.
- Include settlement units—deflection between elements.
- Add button to export all photos to a photo log and organize the folders correctly. This would be useful since normally, one inspector inspects the bridge while another inspector is taking photos.