12/09/09

REVISED

INTEGRATED REMOTE SENSING AND VISUALIZATION (IRSV) SYSTEM FOR TRANSPORTATION INFRASTRUCTURE OPERATIONS AND MANAGEMENT

-PHASE ONE-

VOLUME 2

Knowledge Modeling and Database Development

By

Dr. Seok-Won Lee
Dr. William J. Tolone
Rashna Vatcha
Mahalakshmi Vishnampettai Raghuraman

University of North Carolina at Charlotte
Charlotte, North Carolina

Center for Transportation Policy Studies &
Software and Information Systems Department,
University of North Carolina at Charlotte

USDOT Report Number 01221
December 2009

REVISED

Technical Report Documentation Page

1. **Report No** **USDOT/RITA CRSV-DT0S59-07-H-005**	2. **Government Accession No.**	3. **Recipient Catalog No.**
4. **Title and Subtitle** INTEGRATED REMOTE SENSING AND VISUALIZATION (IRSV) SYSTEM FOR TRANSPORTATION INFRASTRUCTURE OPERATIONS AND MANAGEMENT: PHASE ONE, VOLUME TWO, KNOWLEDGE MODELING AND DATABASE DEVELOOPMENT	5. **Report Date** December 2009 6. **Performing Organization Code** 551356/501356	
7. **Authors** Seok-Won Lee, William J. Tolone, Rashna Vatcha, Mahalakshmi V Raghuraman	8. **Performing Organization Report No.** 12-09-2009	
9. **Performing Organization Name and Address** Department of Software and Information Systems UNC Charlotte 9201 University City Blvd. Charlotte, NC 28223-0001 704-687-5953	10. **Work Unit No.** 11. **Contract or Grant No.** **DT0S59-07-H-005**	
12. **Sponsoring Agency Name and Address** Research and Innovative Technology Administration, USDOT 1200 New Jersey Avenue, SE Washington, DC 20590	13. **Type of Report and Period Covered** Final Report; June 1, 2007 through December 31, 2009 14. **Sponsoring Agency Code**	
15. **Supplementary Notes** This is Volume Two of a seven-volume set of Draft Final Reports describing accomplishments of the IRSV project.		

16. Abstract

The Integrated Remote Sensing and Visualization System (IRSV) is being designed to accommodate the needs of today's Bridge Engineers at the state and local level from several aspects that were documented in Volume One, Summary Report. The following provide supplementary descriptions of the input by the Software and Information Systems (SIS) Team in developing the Knowledge Modeling and Database Development of the IRSV Prototype.

- An explicit language reflected in bridge management processes and the relationships among the language attributes and their semantic understanding at different level of abstraction
- Problem Domain Ontology (PDO) that enables bridge managers to solve complex problems where the underlying domain concepts provide a collective understanding of the bridge data based on domain knowledge from multi-dimensional resources
- A model of the domain knowledge of bridge inspection processes by using the ontological engineering toolkit called Generic Object Model (GenOM)
- Support the ability for bridge managers provides to browse, access, query and reason about complex bridge inspection processes
- Provide a method for answering "what-if" queries via GenOM by matching various initial conditions and circumstances based on rules specified in the PDO
- Rules with ontological concepts and properties that describe the problem domain to support inferences about the problem domain
- Temporal knowledge provided through a flexible Service-oriented Architecture (SOA) framework to compose and provide services on-demand to other system modules
- A software system that is interoperable, scalable, and adaptable that facilitates heterogeneous data requirements, operational requirements, and overlapping functionalities
- Functional requirements primarily based on input and feedback provided by North Carolina Department of Transportation (NCDOT) and Charlotte Department of Transportation (CDOT), which will be expanded to other states and localities in Phase Two
- And finally, an IRSV prototype user interface that combines bridge inspection data and domain knowledge based on a knowledge representation and goal-driven modeling technique

17. **Key Words** Commercial Remote Sensing, Infrastructure Management, Asset Management, Visual Analytics, Knowledge-based Modeling, Service-oriented Architecture, High-resolution Aerial Surveys, Bridge Management Systems	18. **Distribution Statement** Unlimited distribution
20. **Security Classification (of this page)** **Not classified**	21. **No of Pages** 103
	22. **Price**

Form DOT F 1700.7

Contents

List of Tables

List of Figures

Executive Summary

The Knowledge Modeling and Database Development component of the Integrated Remote Sensing and Visualization System (IRSV) was crafted to support bridge management activities in state and local highway and transportation agencies from a hardware and software system that contains the following properties:

- An explicit definition of language reflected in bridge management processes and the relationships among the language attributes and their semantic understanding at different level of abstraction
- Problem Domain Ontology (PDO) that enables bridge managers to solve complex problems where the underlying domain concepts provide a collective understanding of the bridge data based on domain knowledge from multi-dimensional resources
- A model of the domain knowledge of bridge inspection processes by using the ontological engineering toolkit called Generic Object Model (GenOM)
- Support that provides bridge managers the ability to browse, access, query and reason about complex bridge inspection processes
- A method for answering "what-if" queries by matching various initial conditions and circumstances based on business rules specified in the PDO
- The ability to infer new data and knowledge in support of bridge management activities through business rules described in terms of the concepts, properties and features of the PDO
- Temporal knowledge provided through a flexible Service-oriented Architecture (SOA) framework to compose and provide services on-demand to other system modules
- A software system that is interoperable, scalable, and adaptable that facilitates heterogeneous data requirements, operational requirements, and overlapping functionalities
- Functional requirements primarily based on input and feedback provided by the domain experts at the North Carolina Department of Transportation (NCDOT) and Charlotte Department of Transportation (CDOT), which will be expanded to other states and localities in Phase Two
- And finally, an IRSV prototype user interface that combines bridge inspection data and domain knowledge based on a knowledge representation and goal-driven modeling technique

Acknowledgements

This project is supported by grant number DTOS59-07-H-0005 from the United States Department of Transportation (USDOT), Research and Innovative Technology Administration (RITA). The views, opinions, findings and conclusions reflected in this presentation or publication are the responsibility of the authors or presenters only and do not represent the official policy or position of the USDOT, RITA, or any State or other entity. The authors also would like to acknowledge the guidance and contributions of Mr. Caesar Singh, the Program Manager at USDOT; and the technical assistance of Dr. Moy Biswas of the North Carolina DOT (NCDOT), Mr. Garland Haywood of NCDOT Division 10, and Mr. Jimmy Rhyne of Charlotte DOT

We also acknowledge and appreciate the excellent review and input of our National Advisory Committee: Sreenivas Alampalli, New York State DOT; Mrinmay (Moy) Biswas, North Carolina DOT; Hamad abu-Harash, Iowa DOT; Kelley Rehm, American Association of State Highway and Transportation Officials (AASHTO); K. T. Thirumalai, STI International; Dan Turner, University of Alabama; and Phillip Yen, FHWA Turner-Fairbank Research Center.

2.1. Introduction

2.1.1 Overview: Setting the Stage for Applying Knowledge Modeling to Bridge Management

Infrastructure management and its associated processes are complex to understand, perform and thus, making it hard for infrastructure managers to make efficient, effective, informed decisions. Infrastructure management involves a multi-faceted operation that requires robust data fusion and decision making. In order to protect and build sustainable critical assets, we focus on supporting bridge structure inspection and management. As identified in the Volume One Summary Report on Integrated Remote Sensing and Visualization (IRSV), the system is designed to support bridge management activities from the following perspectives:

- To facilitate a better understanding and stricter enforcement of complex inspection processes that can span the gap between evidence gathering and decision making during bridge management through the implementation of ontological knowledge engineering system;
- To enhance domain knowledge modeling to help build a common understanding among bridge inspectors and managers;
- To integrate the above mentioned needs through a flexible Service-oriented Architecture (SOA) framework in order to compose and provide on-demand services to different IRSV modules (e.g., the Visualization module);
- To demonstrate the integration of a SOA and ontological modeling support within the IRSV prototype;
- To design a database schema for the Charlotte Department of Transportation (CDOT) and North Carolina Department of Transportation (NCDOT) that could serve as a prototype that could be modified for application elsewhere.

The goal of knowledge modeling and database development team is to support bridge inspection processes by developing a system based on a knowledge-based approach that provides a scalable and adaptable platform for all system components to share common knowledge and build a common understanding through knowledge services whose design is based on the well-known service-oriented architecture (SOA). Using the SOA paradigm and an ontological engineering approach, the IRSV system will provide a common platform to integrate heterogeneous system components to share bridge data and domain knowledge that will be flexible, scalable and adaptable.

The database schema was developed in a SQL (Standard Query Language) Server. The IRSV prototype was designed with a user interface that combines bridge inspection data and domain knowledge based on a knowledge representation and goal-driven modeling technique. Also, our team proposed to develop a Problem Domain Ontology (PDO) that enables bridge managers to solve complex problems where the underlying domain concepts provide a collective understanding of the bridge data based on domain knowledge from multi-dimensional resources.

2.1.2. Problem Statement for Research

One of the challenging research issues in the IRSV project is to gather various types of “evidences” that support decision making in bridge inspection process. The IRSV system is an integration of multiple software solutions. Each of these solutions is designed to solve a very specific set of problems. These systems are perceived to have heterogeneous data and operational requirements. The design and development process of these systems are independent of each other, which raises concerns for the feasibility of system integration. Thus, the IRSV system requires an architecture that provides a flexible and scalable solution that enables integration among these software solutions.

2.1.3. Implementation Approach

Our knowledge modeling approach promotes and understanding of the bridge inspection and management domain by not only trying to represent the explicit domain knowledge, but also trying to capture the implicit domain knowledge (inferred from business rules) that bridge inspectors and managers gain from their experience. By representing such knowledge in a machine-understandable form, one can build services on top of the knowledge that can be leveraged by other system modules.

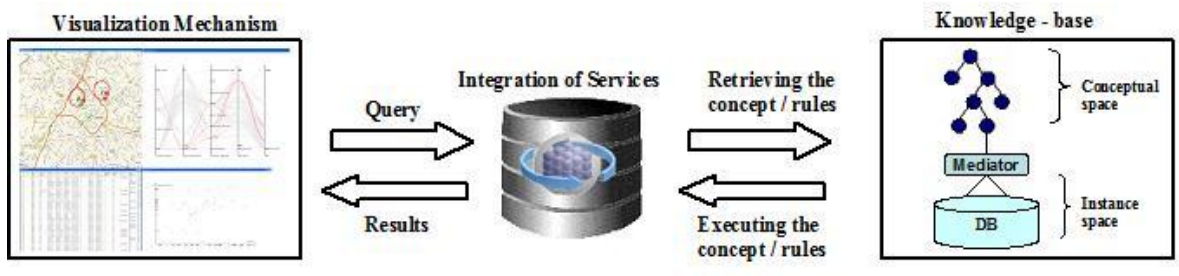
Our approach enables flexible and scalable integration of different modules by using a Service-Oriented Architecture (SOA) framework (Gandhi, et. al., 2006) (Lee et. al., 2009) (Lipyeow, et. al., 2006) (Papazoglou, et. al., 2003), which also promotes the encapsulation of individual software solution functionality as “system services.” A repository of these services is exposed to other system components through a service interface, which functions as a cohesive and coordinated point of integration for all system services.

Another important aspect of our approach is the ability to map business requirements to individual system services, also referred to as process composition. Business requirements can be met by sequences of activities that describe business processes. Incorporating business process representations into the knowledge model (i.e., the Problem Domain Ontology (PDO) (Lee, et. al., 2006)) will enable the IRSV system to respond more effectively to business requirements, e.g., inspection requirements and activities.

The primary objective of knowledge modeling team is to use SOA paradigm and an ontological engineering approach to build the prototype IRSV system that combines the various bridge inspection data and domain knowledge based on the frame-based knowledge representation and a goal-driven modeling technique. In this way, the IRSV system will provide a common platform for heterogeneous system components to share bridge data and knowledge under a base framework that will be flexible, scalable and adaptable.

As large amounts of heterogeneous data are available to bridge inspectors, it is often difficult for bridge managers to relate and reason about the data. Also, the complexity of the data makes it difficult to define the concepts (e.g., data types with embedded semantics) and the relationship among the data in the database. By connecting the concept and the data approach through the knowledge structure, the bridge manager can benefit by creating business rules that can infer implicit knowledge (e.g., concepts) and data to assist managers in decision-making activities.

Our goal is to enable bridge management engineers to retrieve the “right” bridge data more efficiently from the database using a Semantic Matching Operation provided by the PDO (i.e., the knowledge model). This approach will enable the creation of “meaningful and useful database queries” through interactive knowledge acquisition with the subject matter expert (as explained in Chapter 2.2.5). By introducing an ontology-based Semantic Matching Operation, IRSV will enable bridge managers to improve their analyses by leveraging a domain knowledge understanding and its associated representation.



(a) Visualization Mechanism (b) Integration Service (c) Ontological Knowledge Base and Data Base

Figure 2.1: Framework of IRSV for Bridge Management System

Figure 2.1 shows the interaction between the knowledge model ((c) in Figure 2.1) and visualization mechanism ((a) in Figure 2.1) through a Service-oriented Architecture ((b) in Figure 2.1). The IRSV system is built upon an ontological knowledge representation system that interfaces to a Geographical Information System (GIS)-based platform, existing Commercial Remote Sensing – System Integration (CRS-SI) technologies, satellite imaging, visual inspection guidelines and large-scale data visualization. The system provides real-time structural information, structural loss estimation, and post-event damage assessment through an interactive visual interface. It aims to provide the right information at the right time to the bridge managers to help them make the most informed decision regarding bridge maintenance/repair.

2.1.4. Related Enhancements to Bridge Management Processes

There are commercially available bridge management systems that are commonly used for transportation infrastructure asset management. One of them is Hansen system (Hansen information technology), which generates only descriptive reports based on a subset of attributes describing inspection data and information about DOT assets; it does not contain any analytical components for the bridge inspectors to perform analyses. Another bridge management system, PONTIS (bridge management system), develops an optimal bridge preservation policy, simulates conditions and generates work candidates to recommend preservations and improvements of bridges by considering expenditures and predict a broad range of performance measures. However, it also lacks analytical components for the bridge inspectors to perform analyses.

The analytical component will enable inspecting engineers to work with not only rudimentary data-level attributes but also with higher level notions or concepts (i.e., knowledge) of the domain in order to acquire a comprehensive understanding of the complicate critical infrastructure management that requires efficient, effective and informed decision making. Ontological knowledge engineering approach has been successfully used to provide such support to various domains including Certification and Accreditation (C&A) for security documents (Lee, et. al., 2006). Thus, the formulated conceptual space bridges the gap between evidence gathering (complex data) and decision making (conceptual model). However, this bridging requires automated mapping of the complex data space to the easily understandable conceptual space, which is the main motivation of the enhanced domain knowledge modeling.

Souripriya, et. al., (2006), address the problem of supporting ontology based semantic matching in RDBMS by enabling users to reference ontology data directly from SQL using the semantic match operators, thereby opening up the possibility of combining such queries with other operations (e.g., joins) as well as making the ontology-driven application easier to develop and more efficient to operate. In contrast, other approaches use RDBMS only for the storage of the ontology. Querying of ontology data (i.e., the knowledge model) is typically done via APIs. Our approach presents the ontology-related functionality (including inference capability through business rules), overviews how it is implemented on top of Oracle RDBMS, and illustrates the usage with several database applications.

Srinivasan, et. al., (2005), introduces a framework for managing relational data and hierarchical domain knowledge together. The framework persists taxonomies contained in ontological models by leveraging XML support in hybrid relational-XML DBMS (e.g., IBM's DB2 v9) and rewrites ontology-based semantic matching queries using the industry-standard query languages, SQL/XML and XQuery. The approach of semantic data management is to manage the domain knowledge in the same framework as the data are managed in the DBMS and leverage native XML capabilities in a DBMS to support inference operations.

Lipyeow, et. al., (2004) focus on introducing “Extensible Indexing” which is a SQL-based framework that allows users to define domain-specific indexing schemes, and integrate them into the Oracle8i server. Extensible indexing framework enables domain indexes to operate essentially the same way as any other Oracle built-in index, the primary difference being that Oracle server will invoke user supplied code specified as part of the index type to create, drop, truncate, modify, and search a domain index.

2.2 Bridge Management Knowledge Structure

DOT databases contain multitude of heterogeneous data with complex correlations. To effectively process these data, it is necessary to represent the data in a machine understandable form. This can be accomplished using meta-knowledge to represent the semantics of this large data repository. But meta-knowledge representation in databases is complex and inefficient when compared to a standard database schema used to store information (Lim, et. al., 2007). To address this knowledge representation problem, we take an ontology-driven domain knowledge modeling approach. The use of this goal-driven approach is to model the understanding process that underlies the semantics of data and the way the process is implemented in the proto-type system.

The challenge facing bridge managers is to gather various types of “evidences” that support effective decision making during bridge inspection and management processes. These evidences can be textual documents, photo images, sensor images, geospatial notations, etc. Based interactions with the bridge inspectors and managers, and acquired knowledge on the standards of the bridge inspection process, we designed a general knowledge structure for the bridge inspection process. (To replicate a specific bridge inspection process exactly is impractical, as different bridge inspectors and managers take different approaches to bridge inspection and management, which was made very evident during our discussions with the bridge inspectors and managers.)

As such, we had to look at the bridge inspection and management processes from a certain level of abstraction that reduces the subjective variations to a minimum and at the same time preserves purpose and context of the processes. Through repeated interactions with bridge inspectors, managers, and other domain experts, it was determined that the domain of bridge inspection and management is based on a very complex body of knowledge which contains many internal dependencies. In order to make the correct decision, a bridge manager has to understand this knowledge, the dependencies, and all the factors contributing to his/her decision making process. Given the vast number of variables involved, bridge manager can be easily overwhelmed. In addition to the bridge inspection guidelines, bridge manager experience and intensive training were valuable resources. Therefore, it was imperative to capture the inherent knowledge collected by bridge managers in their line of work during the domain modeling phase.

2.2.1 Literature Review on Knowledge Structure

An ontology is a conceptualization of domain knowledge which comprises concepts, properties and their relationships. A Problem Domain Ontology (PDO) is a focus ontology designed around a complex problem where the underlying domain concepts have high interdependencies among each other and scoped by building up a problem scenario based on concepts, properties and features defined in the ontology. After collection of domain information pertaining to the bridge inspection and management processes as well as the knowledge from the domain experts from the State and City DOT, this domain knowledge is represented by an ontology-driven domain knowledge modeling approach. We have used Generic Object Modeling (GenOM) toolkit to capture, represent and model this domain knowledge. GenOM provides functionalities to browse, access, query and reason about complex bridge inspection process.

2.2.2 Problem Domain Ontology for Bridge Management

The bridge management ontology was developed with the help of trained civil engineers and both CDOT and NCDOT bridge managers. The parent node (bridge data profile) as described in Figure 2.2 contains the knowledge of bridge asset data, bridge images, bridge data source, damage classification and bridge operative process guidance. Bridge asset data node briefly explains the notion of the structure or classification of the bridge and what type of service does bridge provide. The bridge asset data have a relationship ‘contains’ with its child node. The bridge images concept describes the type of the image, which further relates with the damage classification and helps bridge inspectors to detect defects in bridges. Bridge data source contains the metrics and measures of the bridge. It describes the rating, geometric data and proposed improvement of the bridge. Since the bridge data source contains the rating and posting knowledge, it has the ‘contains’ relationship between the parent and the child node concept. The damage classification describes the types of damages detected by the LIDAR data, AMPIS data and bridge inspectors. The ontology also contains the meta-data of the bridge operative process guidance which describes the NCDOT process.

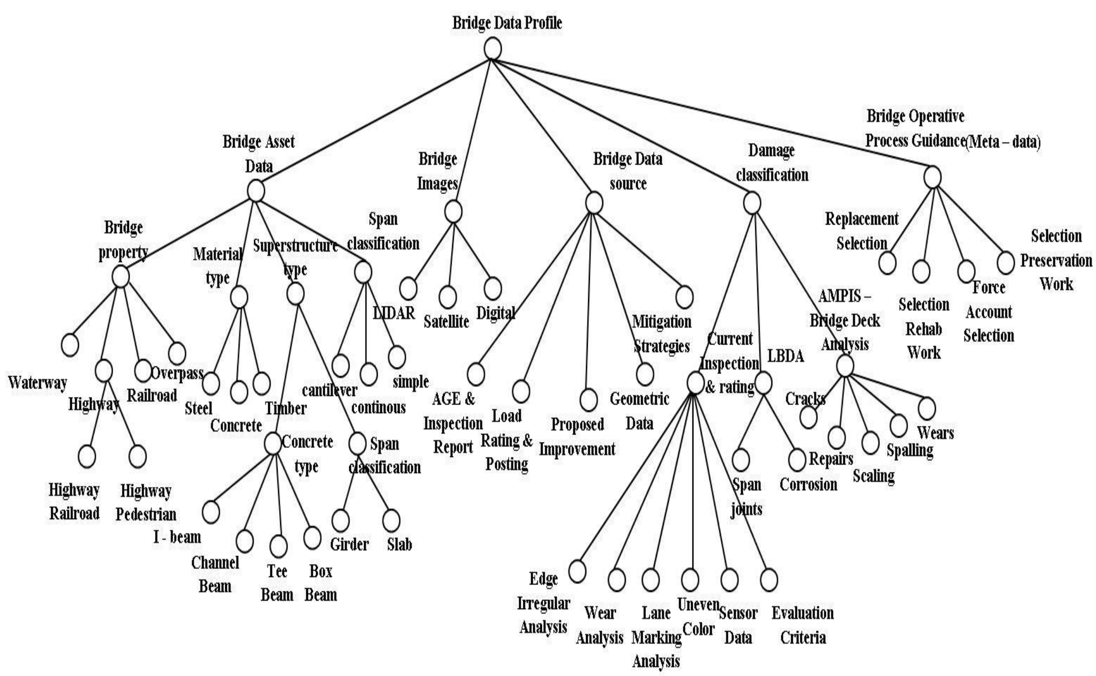


Figure 2.2: Ontology for Bridge Management System

By introducing Ontology-based Semantic Matching, IRSV enables bridge managers to correlate data from previous Inspection Reports and deduce patterns, if any, in these data. For example, the domain knowledge representation will help bridge managers in establishing relationships between the characteristics of a bridge (sufficiency ratings, year of build, condition, etc) over a period of time for which inspection reports exist. These relations will highlight any and all damage the bridge might have suffered over a period of time and the cumulative effect of this damage on the structural integrity of the bridge. Also, with the help of knowledge structure, bridge managers will have the capability of to specific business rules that infer implicit knowledge from the ontology. These rules can provide multiple benefits. First, business rules can expose implicit knowledge to the bridge manager, knowledge that is essential to the reasoning (i.e., decision making) process. Second, business rules can support "what-if" analysis that leverage a set of representative conditions to uncover the prevalence of critical patterns within the knowledge model. We illustrate how business rules can enhance the decision making process and expose critical patterns within bridge inspection data through the following example.

For example, with the help of the proposed knowledge model, we can correlate such defects shown in the Figure 2.2 and further determine other bridges which are affected due to these same conditions. There are more than 200 bridges in North Carolina and sometimes it is difficult for bridge managers to keep track of all the bridges that need attention. Also, the bridge inspection process tends to be very time consuming for all the bridges in states to be inspected regularly. Therefore, IRSV users will benefit from the rules inferred (explained in 2.2.4) from the knowledge structure that will aid decision making based on defects like abutment cap erosion, guardrail cracking, sufficiency deficient and spalling at girder end. In conclusion, we can follow similar pattern to determine the number of bridges which can fall under these factors. By generating this pattern based approach with the help of knowledge structure, bridge managers are provided with the list of bridges that received similar inspections. Also with the help of LIDAR images, satellite images and rating analysis by AMPIS and civil engineers, the knowledge structure can help bridge managers in making decision by associating the defects with the particular bridge. Further, with the help of service oriented architecture framework (described more in detail in chapter 2.3); the knowledge structure will be exposed as services to the visualization module to leverage the results through interactive visualizations.

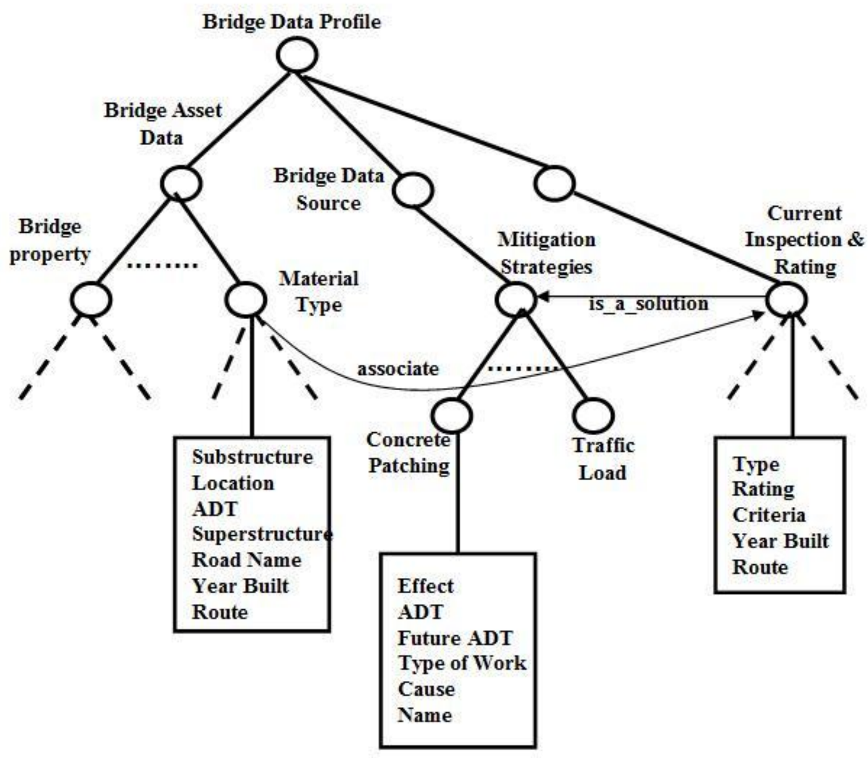


Figure 2.3: A Part of Problem Domain Ontology for Bridge Management

Figure 2.3 depicts a part of PDO for bridge management that describes the concept / object, properties and relationship that is created in knowledge structure using GenOM. Each node of the tree represents a concept (object) having attributes (properties) that are listed in the rectangle box of the particular concept. The feature relationship is shown by an arrow. The knowledge structure of the bridge management consists of (a) bridge asset data that describes the structure of bridge, and (b) bridge images that describe the defect based on the LIDAR, satellite, or digital image. The PDO also contains knowledge like mitigation strategies, geometric data, proposed improvements, rating and postings, types of defects and inspection ratings. It further contains the meta-data of process guidance from NCDOT.

2.2.3 Inference Rules (i.e., Business Rules)

Most scientific texts which describe natural phenomenon are composed of causal relationships (antecedent- consequent), which allow to convey the account of events that result in an informative text. These causal relationships are referred to as inference rules (if-then sentences) in PDO that describe the logical inferences drawn from the assertion(s) in particular form(s). This allows inspection engineers to formulate what-if kind of queries addressing the behavior of a system at a conceptual level. The following showcases an example (as shown in Figure 2.4) which finds all bridges which might have incurred damage due to environmental factors:

IF:

- The type of service of the bride is waterway
- The LIDAR image of the bridge shows severe crack
- The Prompt Action Report is notified

THEN:

- The bridge needs to be rehabilitated or needs repairs.

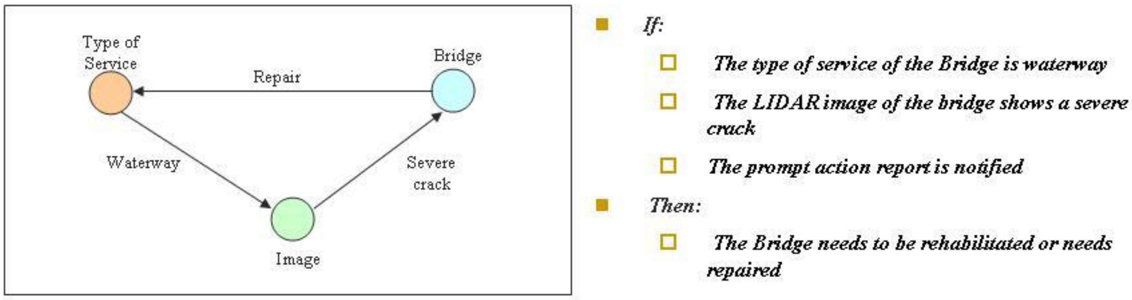


Figure 2.4: Example of Inference Rules

Figure 2.5 describes the rules implemented in rule library which are inferred from the knowledge structure using GenOM toolkit. Inference engine tab is used to develop the rules based on the concepts, properties and features. The rules are created using If-Then logic statements. These are few examples created based on the seven factors which influence the replacement decision of bridges. Rule 1: “Rehabilitation by rating and posting” describes the scenario where all the bridges are affected by scour critical AND waterway adequacy AND status AND present condition AND ADT AND load rating and posting. Variable 1 in 'If' condition contains 'Appraisal' as a concept or object defined in the bridge management knowledge structure which contains scour critical and waterway as properties and then compares to the value that is equal to 'true' which is the GenOM object instance.

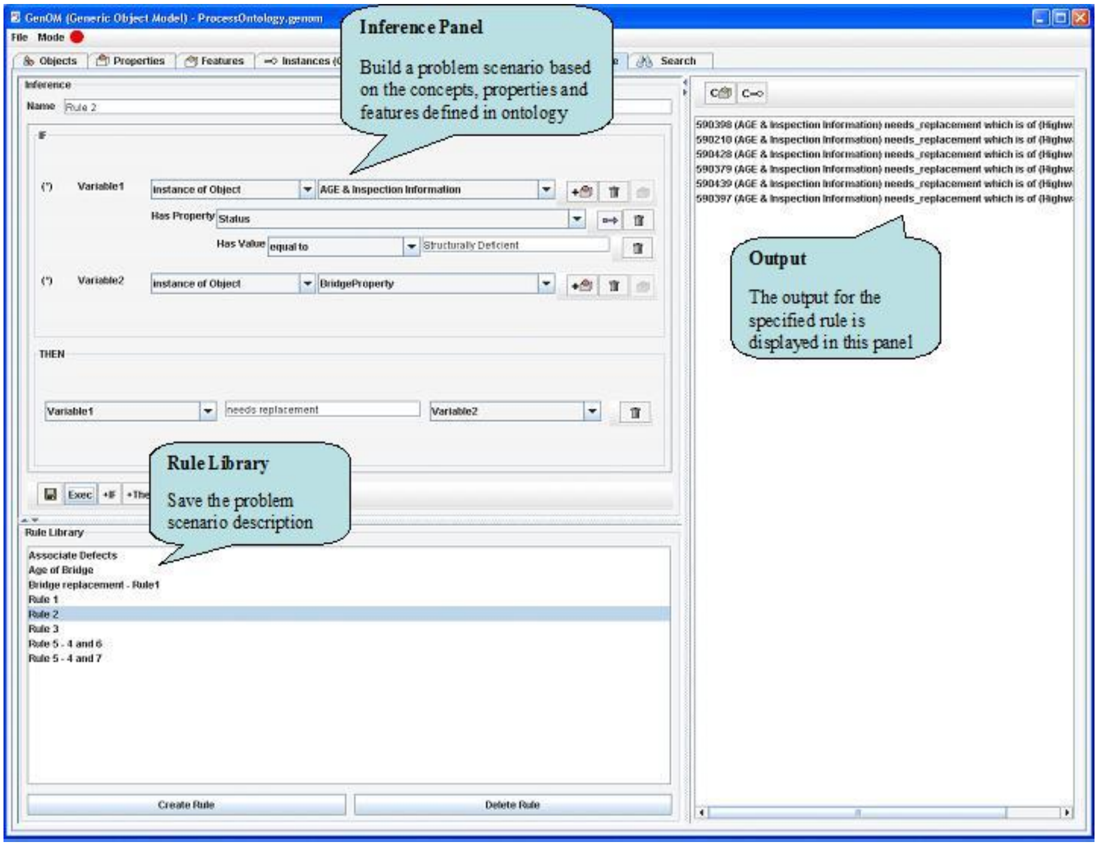


Figure 2.5: Inference Engine using GenOM Tool

Figure 2.5 describes scenario for Rule no. 2 that shows the number of bridges which fall under 'structural deficiency.' As we can see from the Figure 2.5 that in year 2006 there are around 6 bridges falling under 'structural deficiency' status. To study more in detail, the knowledge modeling team has analyzed over 20 bridges that represent the CDOT and NCDOT bridges. Based on expert knowledge, the conditions that are critical to bridge management decisions are listed. Based on these conditions, rules are generated. Following are the rules explained in detail that influence the replacement decision of bridges. First, we present the conditions that are used to design the rules.

Condition 1: Scour Criticality
To makes sure that scour problems do not threaten the structural integrity of bridges, this condition filters all scour critical bridges. The ratings for each bridge are provided in the database.

Condition 2: Water Inadequacy
The bridges falling under the ratings of 2, 3 and 4 which has 'waterway adequacy' as an attribute needs attention. Therefore, this condition filters bridges with inadequate waterway attribute.

Condition 3: Structural Deficiency

This covers all the bridges that fall under the status ‘structurally deficient’
Condition 4: Functional Obsolescence
The condition applies to all functionally obsolete bridges
Condition 5: Present Condition
If the present condition is poor, then the bridge could be a potential candidate for bridge replacement decision based on its applicability to other conditions.
Condition 6: Load Posting
This condition filters bridges posted for load.
Condition 7: High ADT
ADT over 20000 is considered heavy traffic and this information is critical to bridge maintenance decisions if one or some of the other six conditions hold true for a bridge.

Rule based Inference using GenOM

As mentioned before, GenOM offers a rule library as a repository of user defined rules. In this case, based on these seven conditions, the user can define rules and execute them to view results. For the purpose, the bridge data should be loaded into GenOM. The rule should specify a cause (conditions to check) and an effect (suggestions on decision to be made). The rule can hold multiple causes and effects. The following section describes the list of rules created and how the rules can be used to aid in decision making for bridge replacement. Any number of rules can be created in the Rule Library. Upon execution of each rule, the list of bridges for which the rule holds true will be displayed. Screenshots of the eight rules and the results are provided.

Rule Library

Rule 1 - Appraisal Bridge Data Source: If all conditions are met, then bridge needs replacement.

IF:

Scour Criticality == "TRUE" AND
Water Inadequacy == "TRUE" AND
Structural Deficiency == "TRUE" AND
Functional Obsolescence == "TRUE" AND
Present Condition == "Poor" AND
Load Posting == "TRUE" AND
ADT > 20000

THEN:

Bridge needs replacement

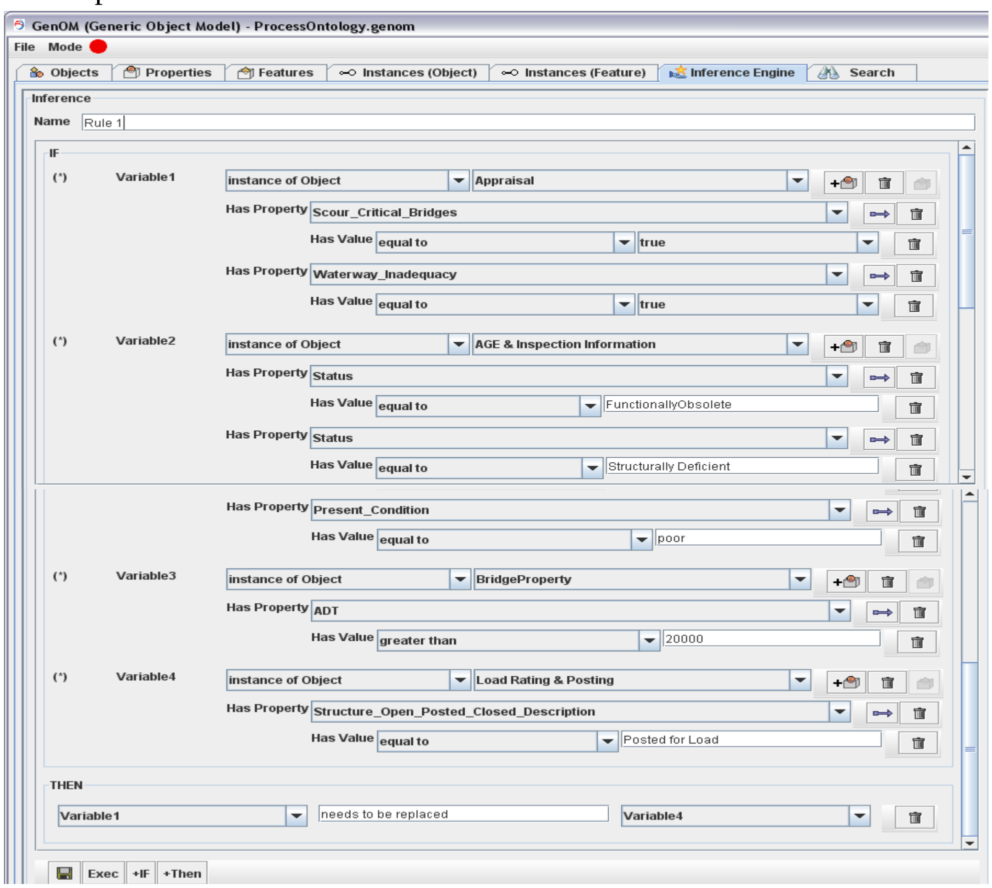


Figure 2.6: Rule 1 created using Inference Engine in GenOM

Inference based on Rule 1 (Inspection year 2006):
None of the bridges fall under this rule since every bridge holds false for at least one of the conditions. GenOM responds with 'No Statements have been inferred'.

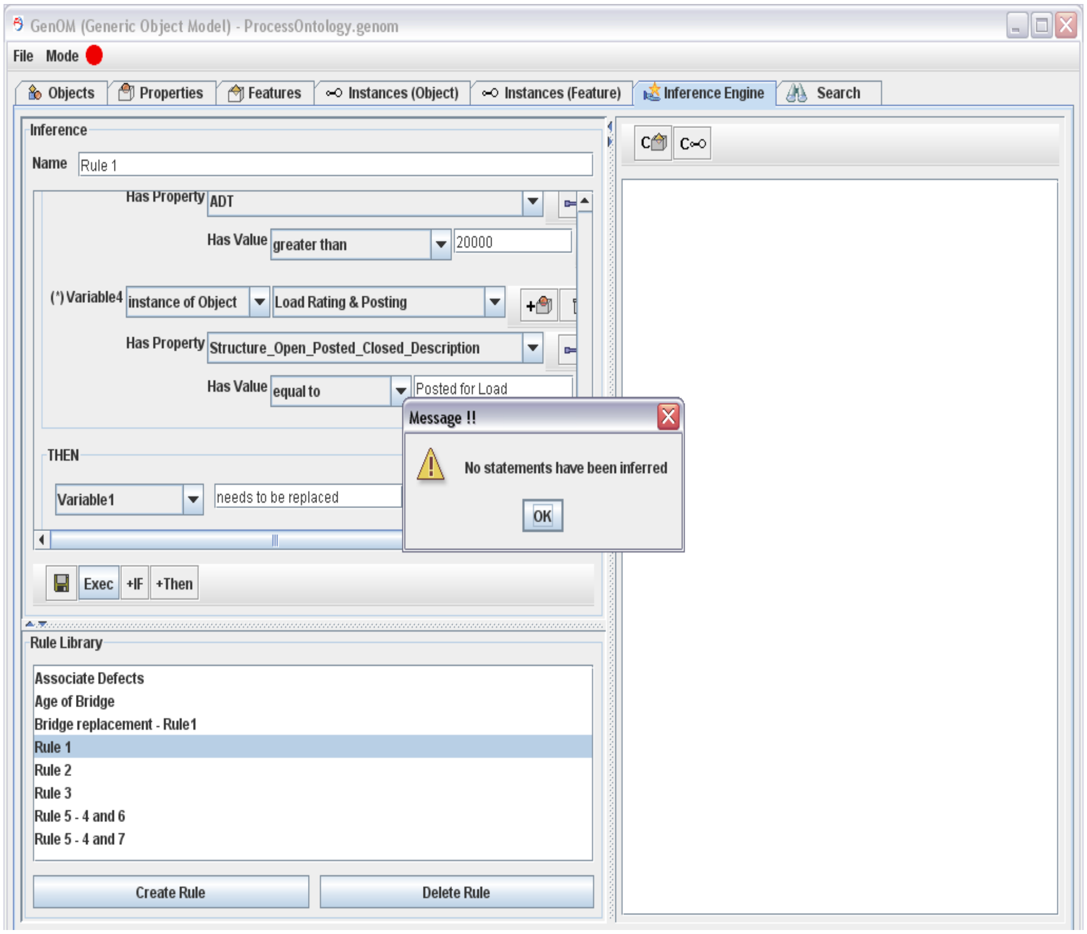


Figure 2.7: The result of Rule 1 created using Inference Engine in GenOM

Rule 2 - Structurally Deficient Bridges: If condition 3 is met, then bridge needs replacement.

IF:
Structural Deficiency == "TRUE"

THEN:
Bridge needs replacement

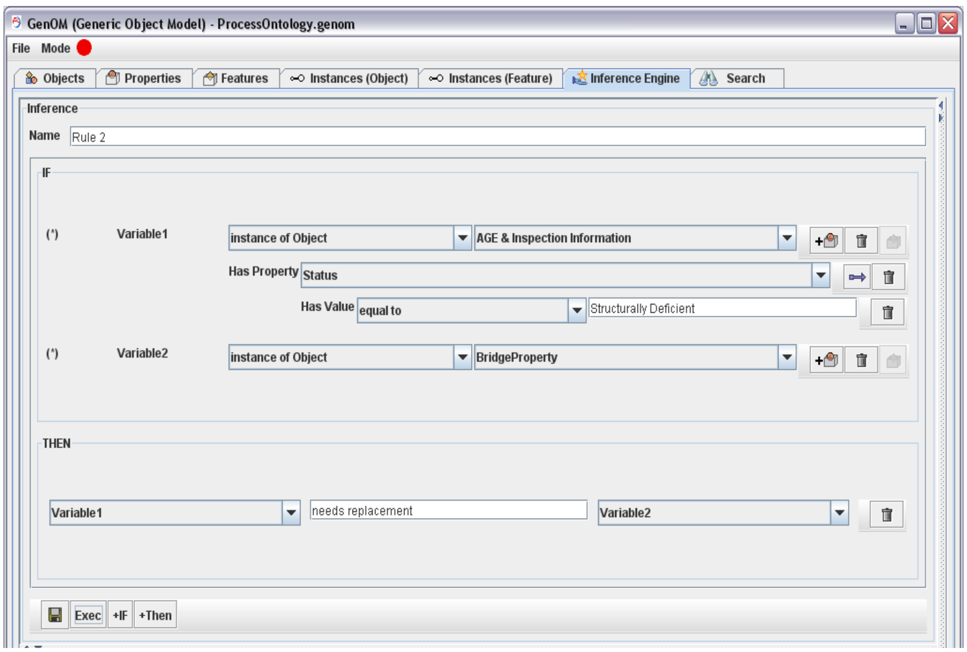


Figure 2.8: Rule 2 created using Inference Engine in GenOM

Inference based on Rule 2 (Inspection year 2006):

Six bridges are structurally deficient. GenOM provides the list of bridge numbers with the suggestion "needs replacement" as specified in the 'then' clause of the rule.

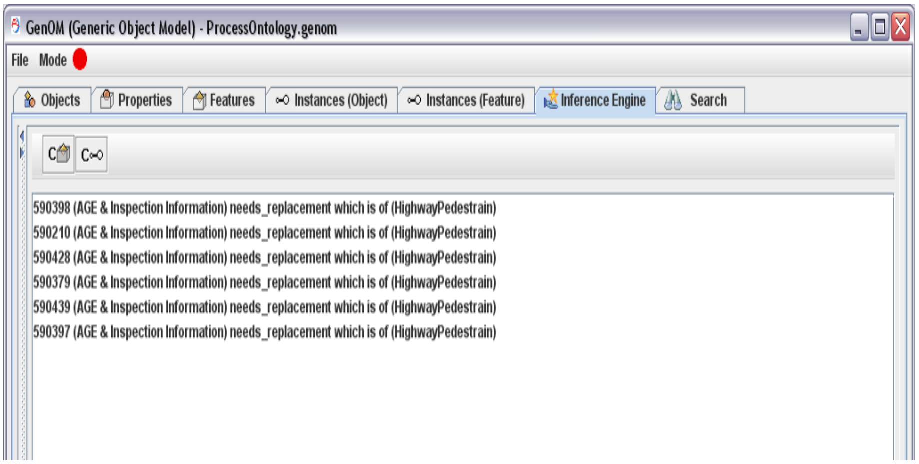


Figure 2.9: The results of Rule 2 created using Inference Engine in GenOM

Rule 3 - Bridge Data Source without Structural Deficient: If conditions 1, 2, 4, 5, 6, 7 are met, then bridge needs additional consideration.

IF:

Scour Criticality == "TRUE" AND
Water Inadequacy == "TRUE" AND
Functional Obsolescence == "TRUE" AND
Present Condition == "Poor" AND
Load Posting == "TRUE" AND
ADT > 20000

THEN:

Bridge needs additional consideration.

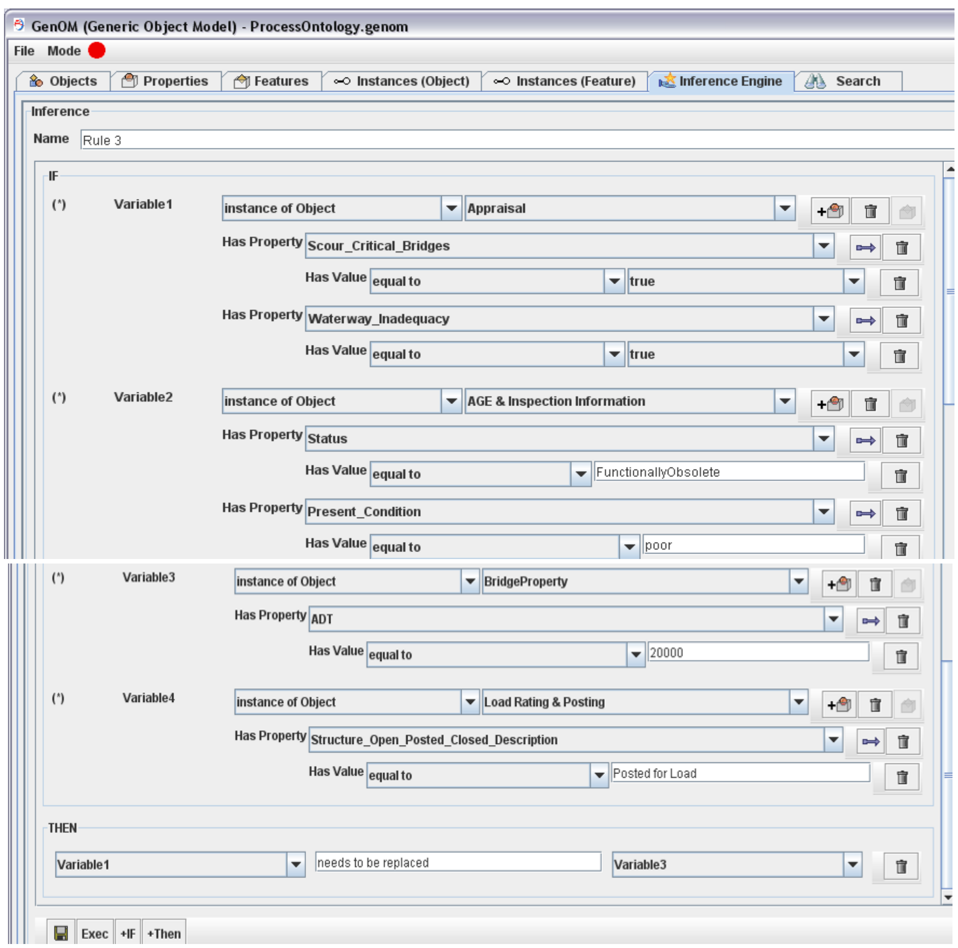


Figure 2.10: Rule 3 created using Inference Engine in GenOM

Inference based on Rule 3 (Inspection year 2006):

None of the bridges exhibit all the conditions for rule 3

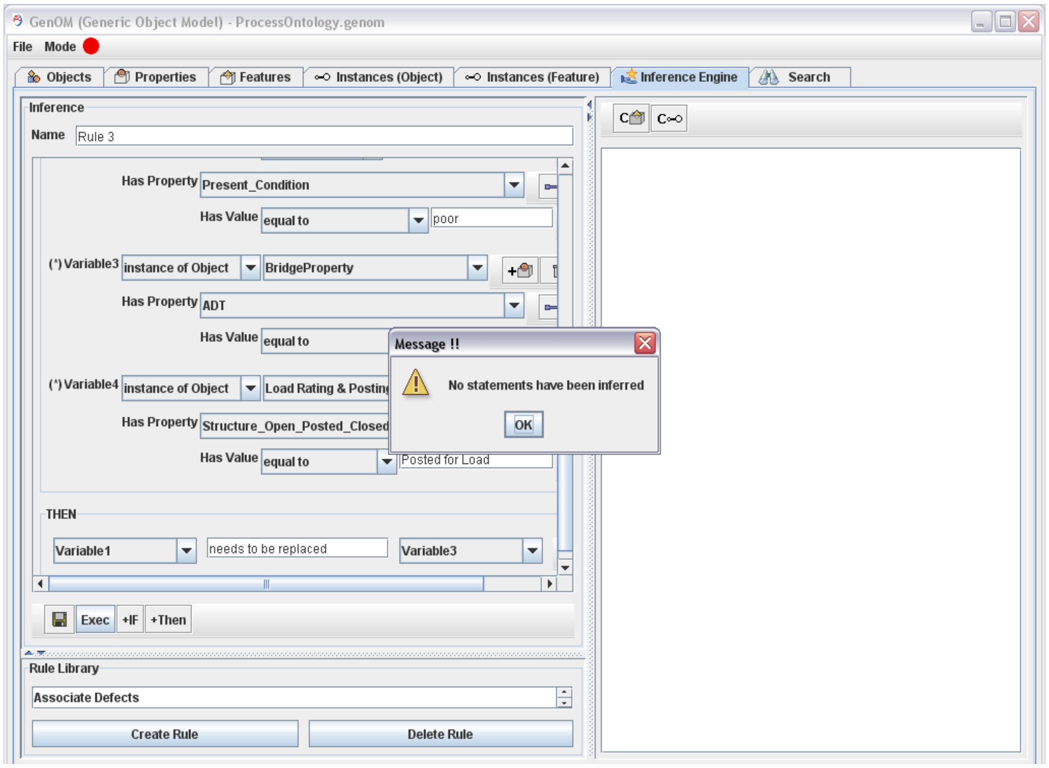


Figure 2.11: The results of Rule 3 created using Inference Engine in GenOM

Rule 4 - Bridges under scour critical and waterway adequacy: If conditions 1 and 2 are met, then bridge needs replacement.

IF:

Scour Criticality == "TRUE" AND
Water Inadequacy == "TRUE"

THEN:

Bridge needs replacement

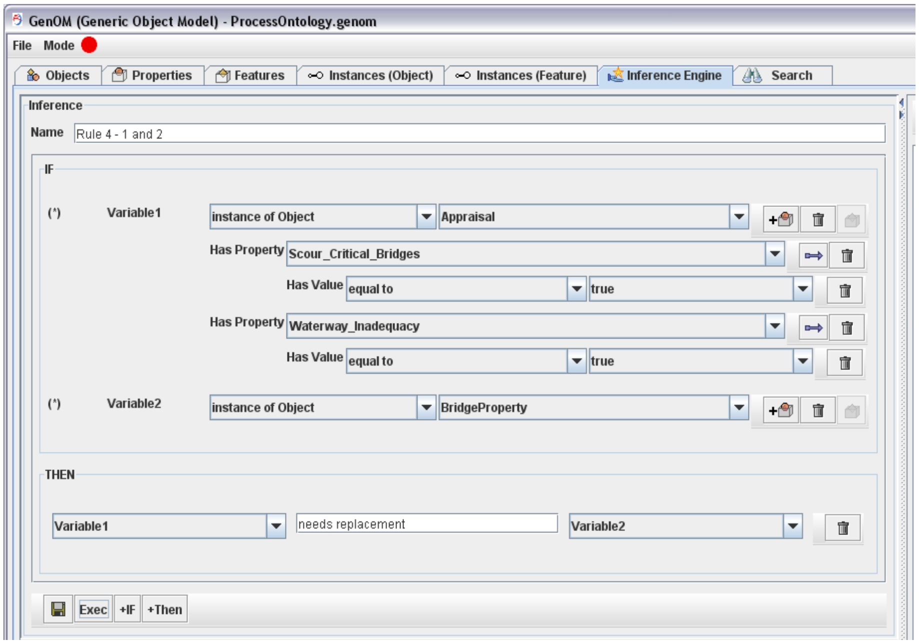


Figure 2.12: Rule 4 created using Inference Engine in GenOM

Inference based on Rule 4 (Inspection year 2006):

None of the bridges are both scour critical and water inadequate. So no statements are inferred.

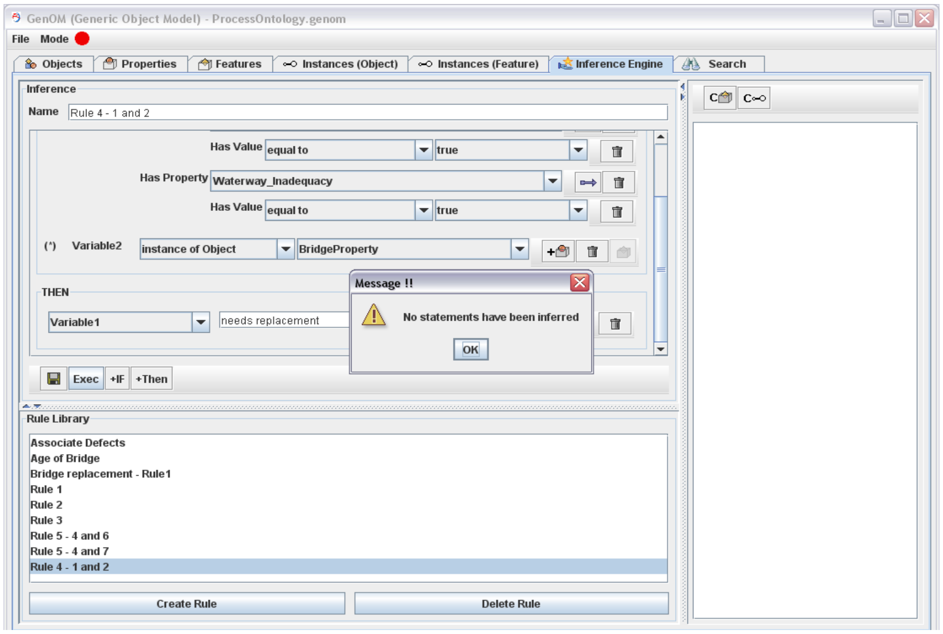


Figure 2.13: The results of Rule 4 created using Inference Engine in GenOM

Rule 5 - Load Rating and Posting Bridges: If condition 4 and 6 are met, then bridge needs replacement

IF:

Functional Obsolescence == “TRUE” AND
Present Condition == “TRUE”

THEN:

Bridge needs replacement

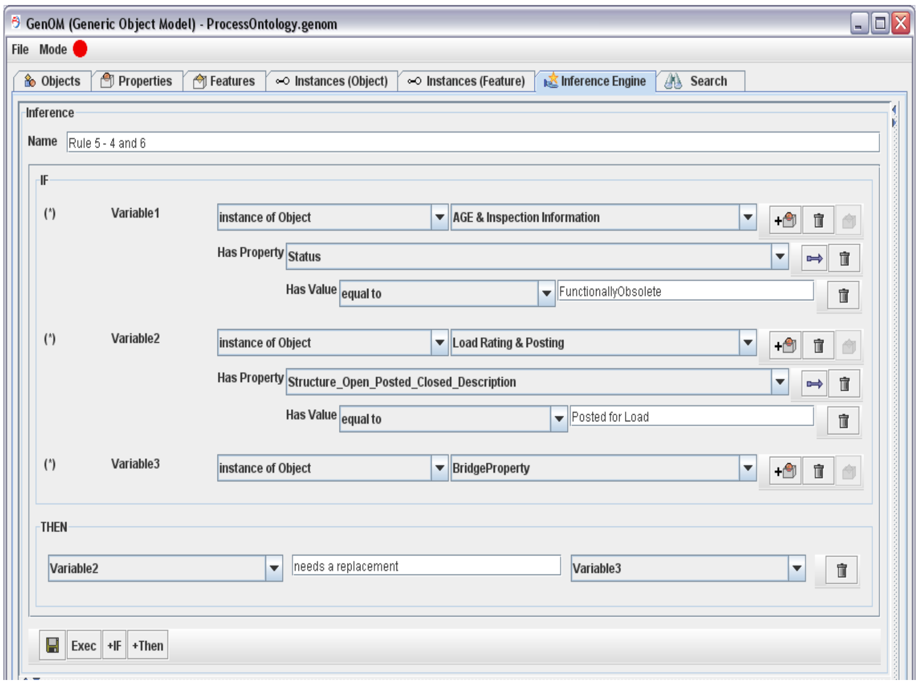


Figure 2.14: Rule 5 created using Inference Engine in GenOM

Inference based on Rule 5 (Inspection year 2006):

Four bridges are both obsolete and posted for load. The bridge numbers along with the suggestion as defined in the rule is displayed.

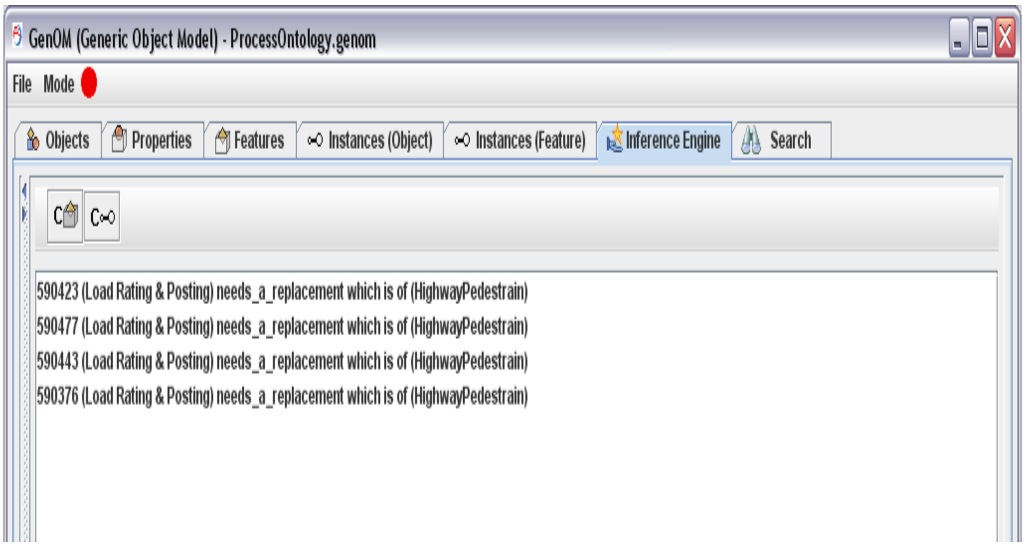


Figure 2.15: The result of Rule 5 created using Inference Engine in GenOM

Rule 6 - Functional Obsolete with high ADT: If conditions 4 and 7 are met, then bridge needs replacement.

IF:

Functional Obsolescence == “TRUE” AND
ADT > 20000

THEN:

Bridge needs replacement

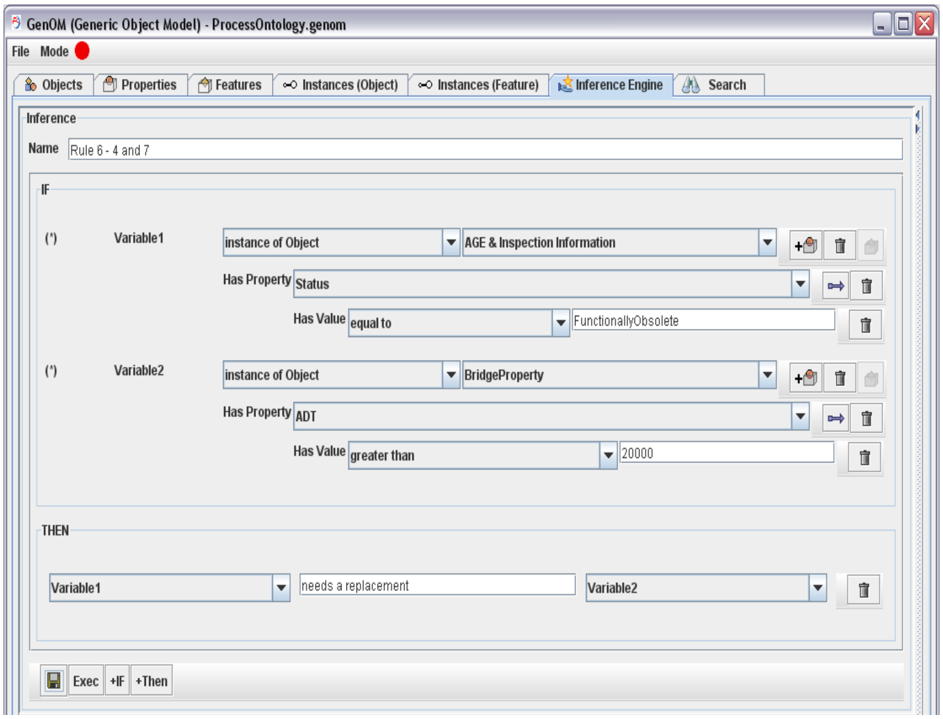


Figure 2.16: Rule 6 created using Inference Engine in GenOM

Inference based on Rule 6 (Inspection year 2006):

Three bridges are both functionally obsolete and carry traffic of more than 20000.

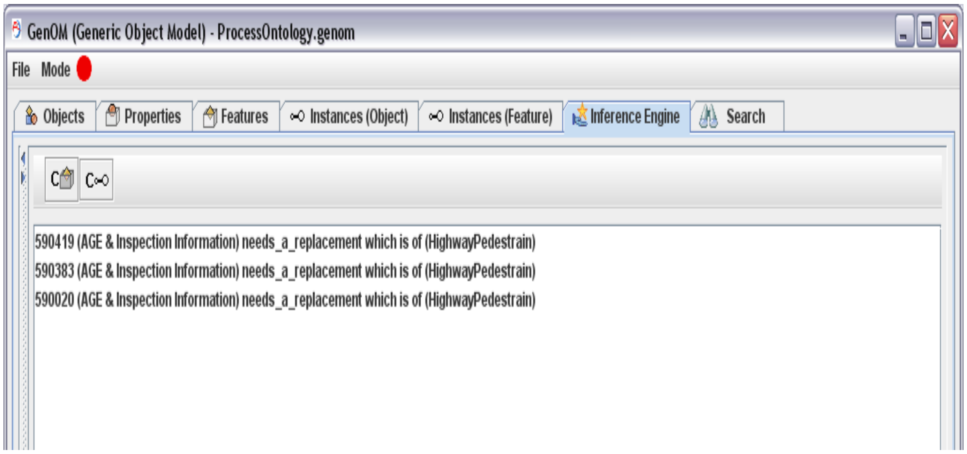


Figure 2.17: The results of Rule 6 created using Inference Engine in GenOM

Rule 7 - Load Posting with High ADT: If condition 6 and 7 are met, then bridge needs replacement

IF:

Load Posting == "TRUE" AND
ADT > 20000

THEN:

Bridge needs replacement

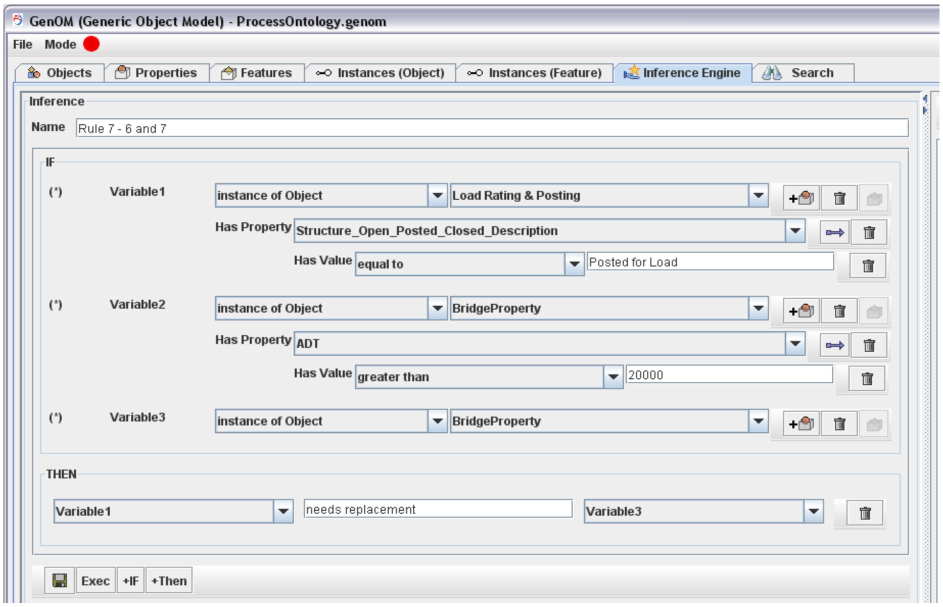


Figure 2.18: Rule 7 created using Inference Engine in GenOM

Inference based on Rule 7(Inspection year 2006):

None of the bridges that are posted for load have high ADT. So no statements are inferred.

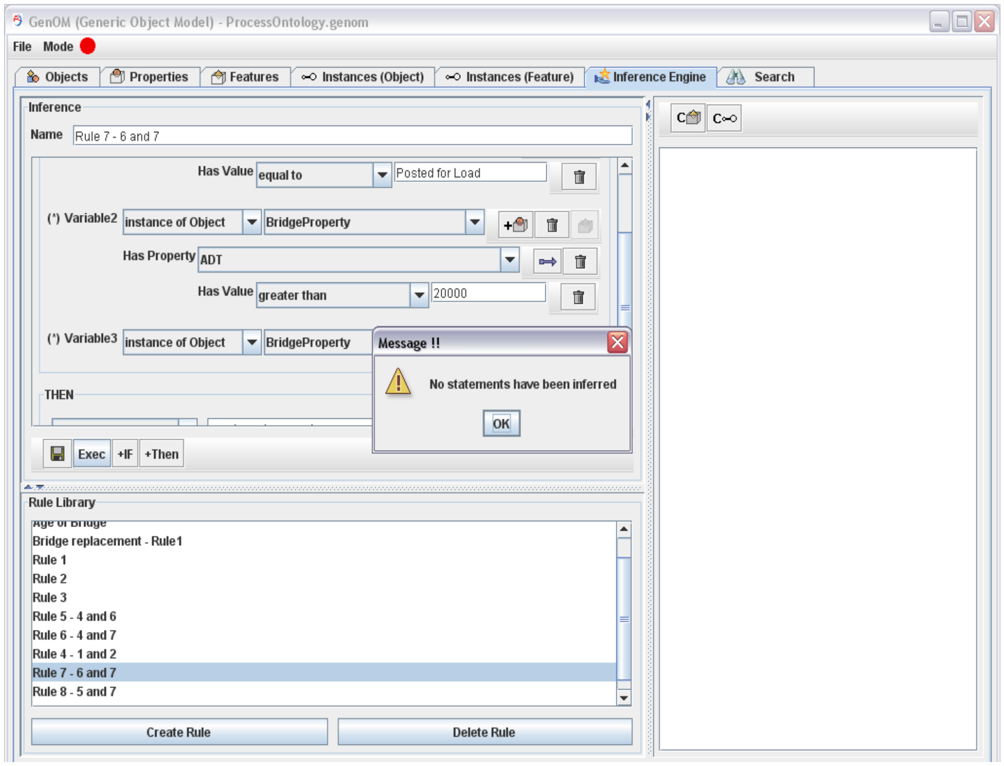


Figure 2.19: The results of Rule 7 created using Inference Engine in GenOM

Rule 8 - Present Condition with high ADT: If condition 5 and 7 are met, then bridge needs no replacement

IF:

Present Condition == "POOR" AND
ADT > 20000

THEN:

Bridge needs no replacement

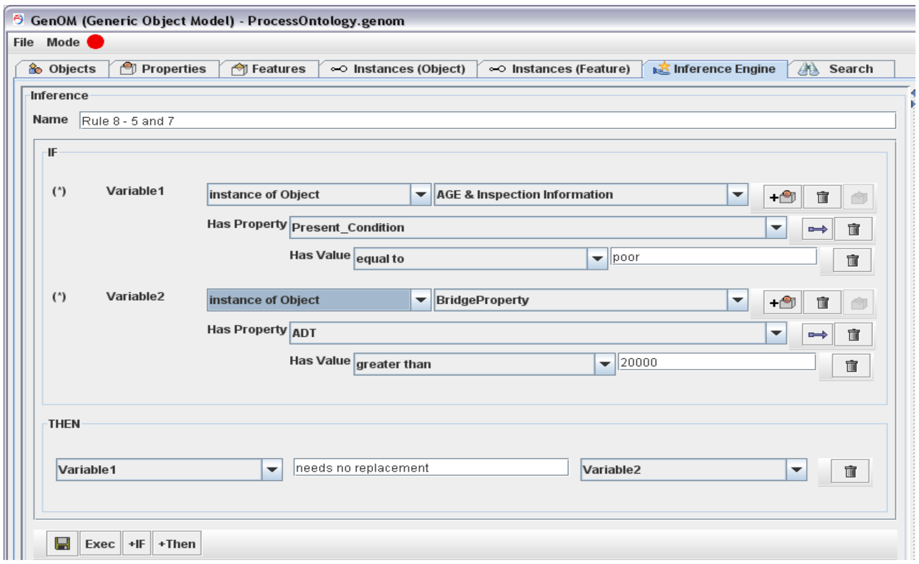


Figure 2.20: Rule 8 created using Inference Engine in GenOM

Inference based on Rule 8 (Inspection year 2006):

Based on the bridge data, the bridge numbered 590700 is in a poor condition with a very high ADT of 30600. Based on the rule and the data provided, the system infers that it needs no replacement.

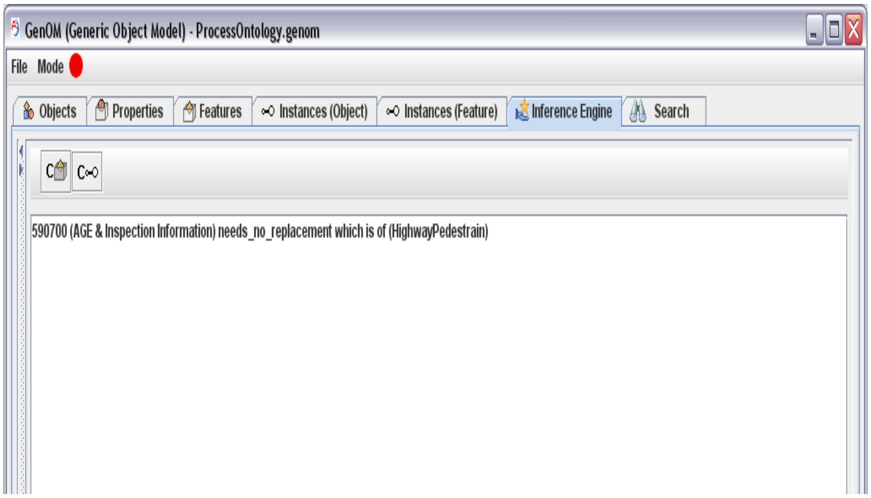


Figure 2.21: The result of Rule 8 created using Inference Engine in GenOM

Bridge Analysis Report

Table 2.1: Overall Bridge Data under Analysis

Total No. of Bridges	20
No. of CDOT Bridges	6
No. of NCDOT Bridges	14

Table 2.2: Bridge Analysis Results Based on Conditions

Condition	Total No. of Bridges	No. of CDOT Bridges	No. of NCDOT Bridges
Scour Criticality	None	None	None
Water Inadequacy	None	None	None
Structural Deficiency	5	1	4
Functional Obsolescence	3	2	1
Poor Present Condition	3	2	1
Load Posting	7	2	5
High ADT	5	1	4

Table 2.3: Rule-based Bridge Analysis Results

Rule	**Decision**	**Total No. of Bridges**	**No. of CDOT Bridges**	**No. of NCDOT Bridges**
Rule 1{Scour Criticality, Water Inadequacy, Structural Deficiency, Functional Obsolescence, Poor Condition, Load Posting, High ADT}	Needs replacement	None	None	None
Rule 2{ Structural Deficiency }	Needs replacement	5	1	4
Rule 3{Scour Criticality, Water Inadequacy, Functional Obsolescence, Poor Condition, Load Posting}	Needs additional consideration	None	None	None
Rule 4{Scour Criticality, Water Inadequacy }	Needs replacement	None	None	None
Rule 5{ Functional Obsolescence, Poor Condition }	Needs replacement	1	1	None
Rule 6{ Functional Obsolescence, High ADT }	Needs replacement	None	None	None
Rule 7{ Load Posting, High ADT }	Needs replacement	None	None	None
Rule 8{ Poor Condition, High ADT }	Needs no replacement	1	1	None

Scenarios: Rule based inference for bridge management decisions

For purpose of illustration, we limit the scope to 20 bridges that represents CDOT and NCDOT bridges.

CDOT Bridges

Scenario for bridge number 590255.

According to the inspection record of year 2006, this bridge is in fair condition. This was built in 1969 and carries 13,000 vehicles per day. It is classified as functionally obsolete bridge but not a scour critical bridge. It has no issues of Waterway adequacy and it is declared open and no restriction.

Figure 2.22: Digital image of bridge no 590255

Inference:

Scour Criticality == “FALSE” AND
Water Inadequacy == “FALSE” AND
Structural Deficiency == “FALSE” AND
Functional Obsolescence == “TRUE” AND
Present Condition == “FAIR” AND
Load Posting == “FALSE” AND
ADT < 20000

By evaluating the bridge for each of the conditions, system infers that none of the conditions hold true for this bridge. So this bridge does not require immediate attention.
Similarly, we have illustrated the inference rules on 20 bridges that represent the CDOT and NCDOT bridges (Please refer Appendix E for more details).

2.2.4. Mapping of Conceptual Space to Data Space

As mentioned earlier, to process the large amount of heterogeneous data effectively, it is necessary to represent it in a machine understandable and process able form. This can be accomplished using the concept of meta-knowledge to represent the semantics of this large data repository. But meta-knowledge representation in a database is complex and not efficient when compared to a standard database schema used to store information. To solve this knowledge representation problem, we take an ontology-driven domain knowledge modeling approach. The use of this goal-driven approach is to model the understanding process that underlies the semantics of the data and the way of the process. However, there are some scalability issues like generating results from the conventional database query process can be a time consuming effort, especially if the database has a large number of instances. Also, the query process does not guarantee a solution to a given problem and it may require multiple queries and manual sense making process. DOT databases contain different types of complex data such as text, spatial, image etc. As a result, the querying process raises performance issues. Therefore, this requires automated mapping of the complex data space to the easily comprehensible conceptual space. The complex data is usually stored in a database, which consist of tables that represent data-level relationship using tables.

By introducing enhanced domain knowledge modeling technique, IRSV will enable bridge inspectors to raise the level of their analyses from a data level to a conceptual level by leveraging a domain knowledge understanding and its associated representation.

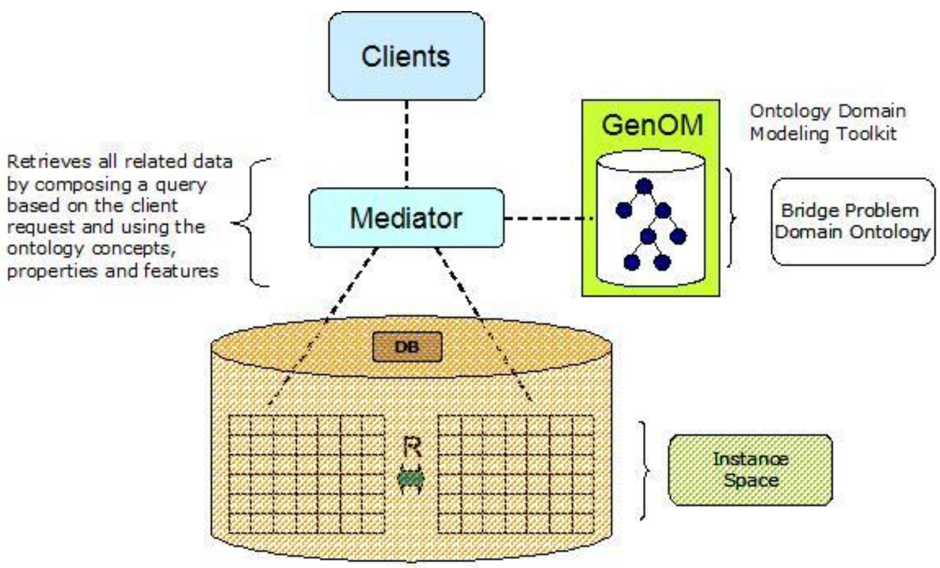


Figure 2.23: Framework for Integration of Conceptual Space and Instance Space

Figure 2.23 describes the framework for integration of conceptual and instance space. A key component of this framework is the knowledge representation that supports the Bridge Problem Domain. Thus, users will retrieve all related data from the database by composing a query that uses the ontology concepts, properties and features through the knowledge mediator. Such queries are different that those constructed by on a database schema. The difference is that database schemas are about the efficient organization of data for storage and retrieval. The problem domain ontology, however, will enable users to construct queries based on a better understanding of the conceptual space. Also, the concepts defined in knowledge structure are not present in the database.

The knowledge mediator will make the IRSV system more scalable and flexible to map and process the large repository of complex data with multiple formats. The knowledge mediator will also enable the plug-and-play of other types of knowledge-based approaches into the IRSV system.

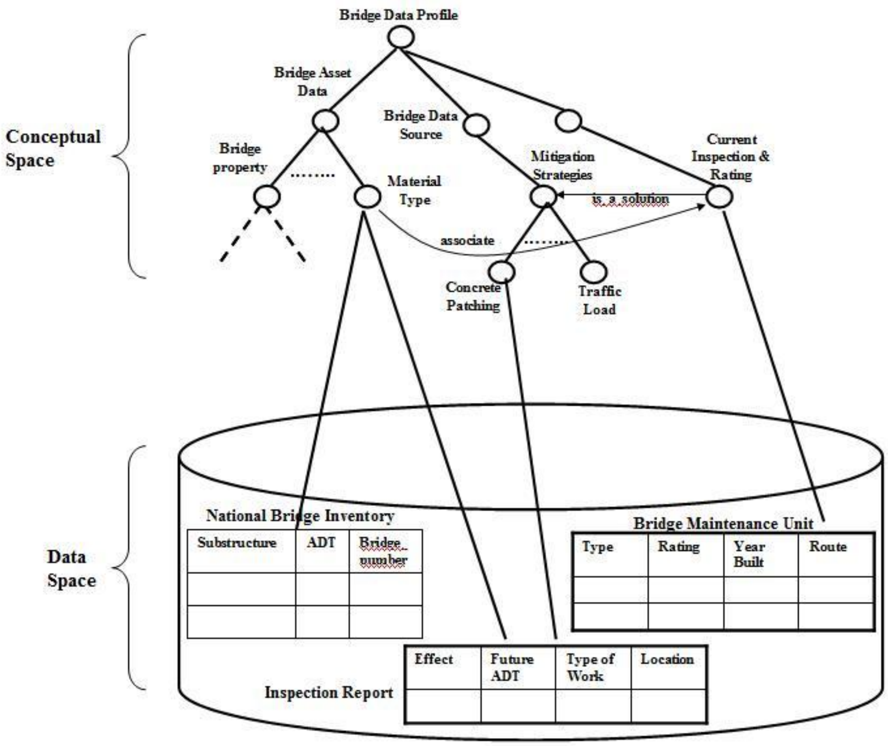


Figure 2.24: Example for Mapping of Concept Space and Data Space

A PDO has finite space of relationships, which can then be used to formulate the mapping to lower-level data space. Each concept maps to a complex data (instance) in the database, whose domain is defined by the possible values existing in the database. There exist concepts $C_1, C_2, \ldots, C_m$ in an ontology that directly map to data attributes. Such concepts, usually, are lower-level concepts that describes about the bridge for this PDO; thus, the instances (data) of a concept can be selected using bridge_id (a number unique to each bridge in the database) attribute of the database. The instances of lower-level concept C_i, which can be directly mapped to some collection of attributes A_i, can be obtained for a particular value V_i of the attribute by the following query:

$$Q_i = \text{Select bridge\_id}, a_1, a_2, \ldots, a_n \text{ from } T_1 \text{ where } a_1 = v_1, a_2 = v_2, a_3 = v_3, \ldots, a_n = v_n$$

Using the above query for lower-level concepts C_1, C_2,…, C_m, we can map the other concepts (say ${}^{m}C_1$) to complex heterogeneous data (instances); they are based upon these lower-level concepts using either of the following relationships:

1. AND relationship: The concept ${}^{i+m}C_i$ is defined based on all the concepts Ci, C_{i+1}, …, C_{i+m}. The instance value taken by ${}^{m}C_i$ is the combination of instances of all the concepts C_i, C_{i+1}, …, C_{i+m}.

   $${}^{m}C_i = Q_i \cap Q_{i+1} \ldots\ldots \cap Q_{i+m}$$

2. OR relationship: The concept ${}^{m}C_i$ exists based on any one of the concepts C_i, C_{i+1}, …, C_{i+m}. The instance value taken by ${}^{m}C_i$ is one of the instances of all the concepts Ci, C_{i+1}, …, C_{i+m}.

   ${}^{m}C_i = Q_{i+k}$, if Q_{i+k} is the instance taken by the concept ${}^{m}C_i$

3. AND / OR relationship: The concept ${}^{m}C_i$ is defined based on all or some of the concepts C_i, C_{i+1}, …, C_{i+m}. The instance value taken by ${}^{m}C_i$ is the combination of some or all the instances of all or some of the concepts C_i, C_{i+1}, …, C_{i+m}.

   $${}^{m}C_i = Q_{i+k} \cap Q_{i+l} \ldots\ldots \cap Q_{i+m}$$

We have associated the complex heterogeneous data to every concept for our PDO using the above equations. However, the engineers trying to analyze will relate one concept to another; thus, we also need the mapping of the relationships to complex heterogeneous data. This is achieved through GenOM inference engine by using rules. The inference engine of GenOM provides rule based inference mechanism to enable the user to query the knowledge base by creating rules. GenOM tool also provides the type of feature that connects the objects or concepts with the appropriate relationship. The type of feature includes regular, symmetric and transitive feature. The regular feature simply relates the two objects in the Knowledge Base. It represents a unidirectional dependency between two objects. For example, bridge property associates with current inspection and rating. A Symmetric feature describes the relationship between two objects such that, object 1 relates to object 2 in the same way as object 2 relates to object 1. A Transitive feature relates two objects in the knowledge base such that, if object 1 is related to object 2 and object 2 to object 3 then relation between object 1 and object 3 is inferred. For example, Bridge property contains damage classification and damage classification 'associated with' mitigation therefore we can infer that the particular bridge of some property 'has_a_solution' for a particular mitigation, say, bridge xyz of type concrete 'has_a_solution' for concrete patching.

We now illustrate the above notions to show the usefulness of enhanced PDO to inspecting engineers through a real-world example. Let us consider a scenario where the bridge inspector wants to know about material types that required mitigation strategies based on damage classification. The concepts in the sentences are 'material types', 'mitigation strategies', and 'damage classification.' The damage classifications of the bridges are identified with the help of LiBE technology and AMPIS technology. Figure 2.25 describes the defects identified by LIDAR image and AMPIS technology of bridge 590140 and bridge 590147. The laser radar system, also called LIDAR, is the optical remote sensing technology developed for range detection. To adapt the range finder for bridge monitoring application, a surface damage detection algorithm, LiBE (LIDAR-based Bridge Evaluation) for material mass loss quantification is developed. LIDAR has the potential for providing high-density, full-field surface static imaging. Hence, it can be used to generate volumetric quantification of concrete corrosion or steel erosion. By recording the surface topology of the object, the laser radar can detect different damages on the bridge structure and differentiate damage types according to the surface flatness and smoothness. These data are further classified and categorized in the form of the concept by using GenOM toolkit.

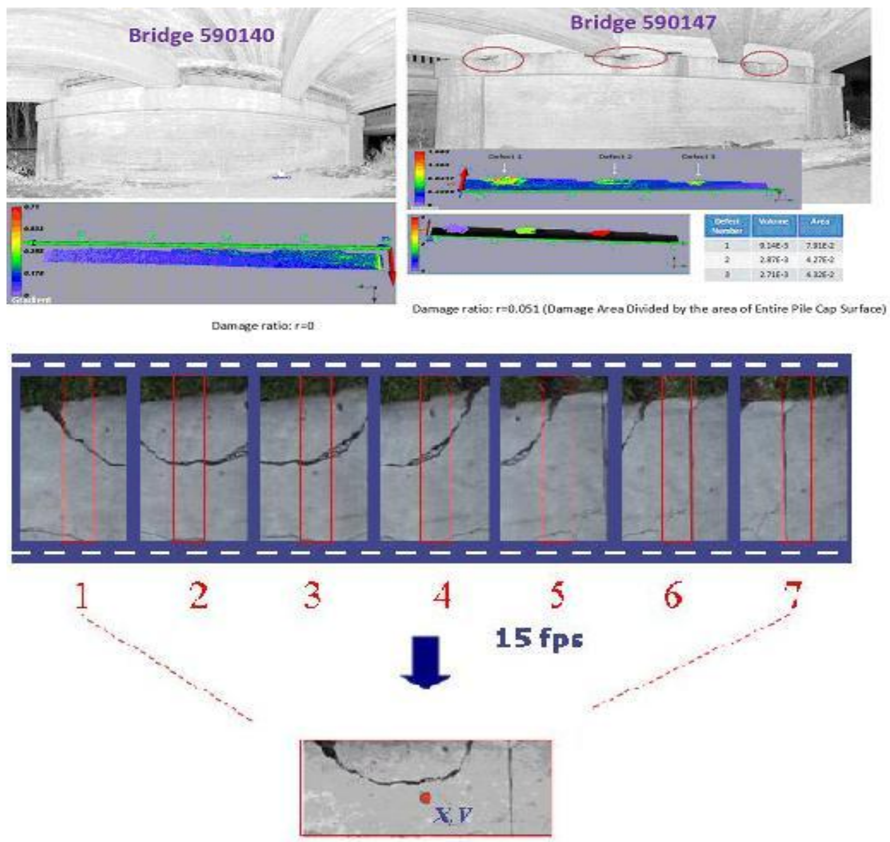


Figure 2.25: LIDAR Image and AMPIS Analysis of Bridge Number 590140 and 590147

The concept 'material types' is mapped to the complex-heterogeneous data using the lower-level concepts through the OR relationship as shown in left-side of the Figure 2.26. The queries for the lower-level concepts are also mentioned. Further, a box indicates the kind of relationship that composes the concept. Similarly for other concepts 'mitigation strategies' and 'damage classification' the corresponding queries are shown.

The concepts 'material types' and 'mitigation strategies' are related through regular relationship; 'mitigation strategies' and 'damage classification' are related through regular relationship. The concept ‘material type’ and ‘current inspection and rating’ share ‘associate’ relationship/feature. The concepts ‘current inspection and rating’ and ‘mitigation’ contain ‘is_a_solution’ relationship /feature. Figure 2.45 shows the mapping of lower-level concepts to the corresponding attributes in the tables. To know about material types that require mitigation strategies based on damage classification, we need to connect the queries using AND operation.

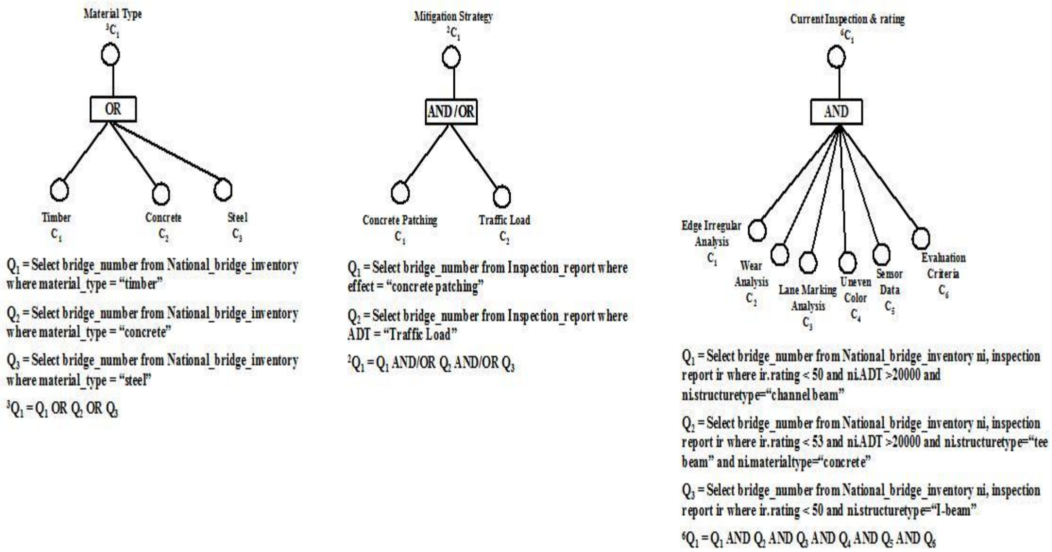


Figure 2.26: Mapping of Concepts to the Corresponding Attributes in the Table

The outcome of this will consist of many bridges where the material type will be associated with damage classification that requires relevant type of mitigation. The bridge number 590140 and 590147 are the examples of this scenario. Figure 2.27 shows the digital image and satellite image of bridge number 590140.

Figure 2.27: Digital Image and Satellite Image of Bridge Number 590140 and 590147

By using this approach the IRSV system will enable the bridge managers to look at and retrieve relevant bridge data from a conceptual perspective by leveraging the domain knowledge understanding and its associated representation. The approach of enhanced domain knowledge modeling will make the IRSV system more scalable and flexible to map and process the large repository of complex data with multiple formats. This will also enable the plug-and-play of other types of knowledge-based approaches into the IRSV system.

2.3 Service–Oriented Architecture Framework

The current software development process employed in realizing the IRSV system has multiple research teams working on component software solutions which include heterogeneous data, operational and functional requirements. For the IRSV system to leverage the needs and functionalities of each software solution effectively, it is very important to build a system architecture that facilitates the interoperability among different software systems, and provide a scalable and adaptable solution to challenging systems integration. This section describes our Service-Oriented Architecture (SOA) framework that can easily integrate tools with different software solutions in a scalable and adaptable manner. We adopt the SOA framework to expose the knowledge service to other tools in the tool-suite. SOA framework (Alonso, et. al., 2004) (Papazoglou, et. al., 2003) (Perrey, et. al., 2003) is responsible for representing and modeling domain knowledge to other tools in the IRSV system in a scalable and adaptable manner.

2.3.1 System Architecture Development

This section will describe the development of the IRSV system architecture as it passed through various stages of development to discover the need for a SOA. We develop the IRSV system architecture that would help the visualization and management of domain data, information and knowledge that are related to the construction and monitoring of the transportation infrastructure, specifically the bridge infrastructure, through efficient and effective data modeling techniques. The schematic representation of the IRSV architecture is provided in Figure 2.28.

The IRSV system has a modular architecture. These system modules are perceived to have heterogeneous data and operational requirements. The design and development process of these system modules are independent of each other, which raises concerns of systems integration. Thus, the IRSV system requires an architecture that provides a flexible and scalable solution that enables integration among these software solutions.

As seen from Figure 2.28, a multitude of data sources feed bridge inspection and inventory data to the IRSV database. The IRSV database is a collection of all the data sources that are used to feed domain information to the various other modules of the IRSV system. Using the Data Management Module, we provide a unified and systematic interface to the IRSV database which can be used by other modules, internal and external to the system, to request relevant data from the database.

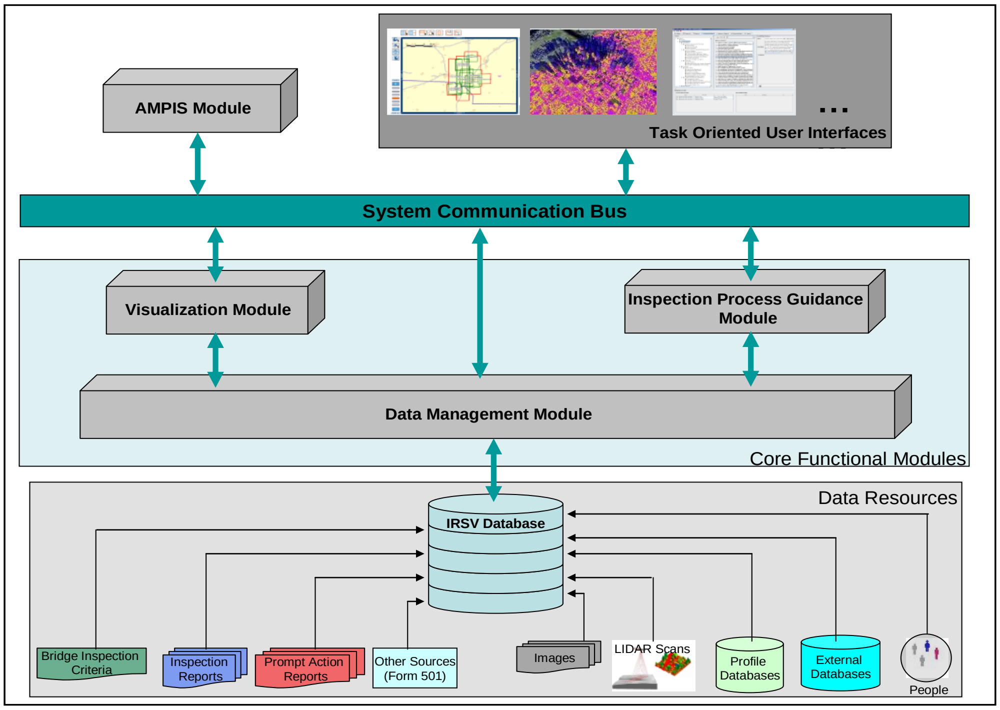


Figure 2.28: IRSV System Architecture Overview

The Visualization Module and the Inspection Process Guidance Module form the core functional modules of the system. The Visualization Module is responsible for providing an interactive, explorative interface to aid in the analysis and decision making process using effective visualization techniques. The Inspection Process Guidance Module is responsible for managing and providing meta-knowledge that facilitates the exploration of the domain knowledge. The Inspection Process Guidance Module is also responsible for representing and managing an understanding of the bridge inspection process within the system. The system communication bus provides a channel to talk to any of the system's core functional modules.

Each of these modules is conceptualized to play a very specific role in helping the user of the system to analyze and visualize information pertaining to the bridge infrastructure. Though each module is unique in its functionality, they all feed off each other's analytical and data processing capabilities. These modules will function together to provide a better understanding of the bridge inspection process and additional guidance to the bridge managers when exploring and interpreting the vast amounts of interrelated data that are collected from multiple data sources, including bridge inspection reports, LIDAR scans, images of the bridge infrastructure, etc.

From the above explanation, we can conclude that IRSV system is also an integration of multiple software solutions. Each of these solutions is designed to solve a very specific set of problems. These systems are perceived to have heterogeneous data and operational requirements. The design and development process of these systems are independent of each other, which raises concerns about the feasibility of system integration. Thus, the IRSV system requires an architecture that provides a flexible and scalable solution that enables integration among these software solutions. The Knowledge Modeling and Database Development team approach will enable flexible and scalable integration by using a SOA, which promotes the encapsulation of individual software solution functionality as "*system services.*" A repository of these services will be exposed to other system components through a service interface, which will function as a cohesive and coordinated point of integration for all system services. As mentioned before, another important aspect of this approach is the ability to map business requirements to individual system services, also referred to as process composition. Business requirements can be met by sequences of activities that describe business processes. Incorporating business process representations into the problem domain ontology will enable the IRSV system to respond more effectively to business requirements, e.g., inspection requirements and activities.

Composing Knowledge Services and a Need for Common Framework

IRSV system is composed of services, which offer a variety of functionalities, including that of providing knowledge. In addition to these services and tools, we have developed an ontology adapter shown in Figure 2.29 that provides very specific functionality of interfacing with and extracting knowledge from external data structures/sources. The architectural framework of the IRSV system has been designed so as to bring all these components together in a way to preserve loose coupling among them, and at the same time exposing all the functionalities of each module to all other modules, without sacrificing on performance or individual functionality. Such an approach to integrating system components makes the system scalable and adaptable to any future modifications/additions to the system.

Knowledge Service via Service Oriented Architecture

Service Oriented Architecture (SOA) would comprise of business services, which is an aggregation of relevant functionalities of each individual software solution. By using SOA framework, IRSV system will provide a common platform to integrate heterogeneous system components to share bridge data and domain knowledge. For example, one of the modules in the IRSV system is the Visualization (VIS) module, which requires knowledge service, a composite of the object, properties and instances from GenOM (Lee, et. al., 2005), the ontology engine. GenOM provides functionalities to browse, access, query and reason about complex bridge inspection process. The ontology adapter provides very specific functionality of interfacing with and extracting knowledge from external data structures/sources.

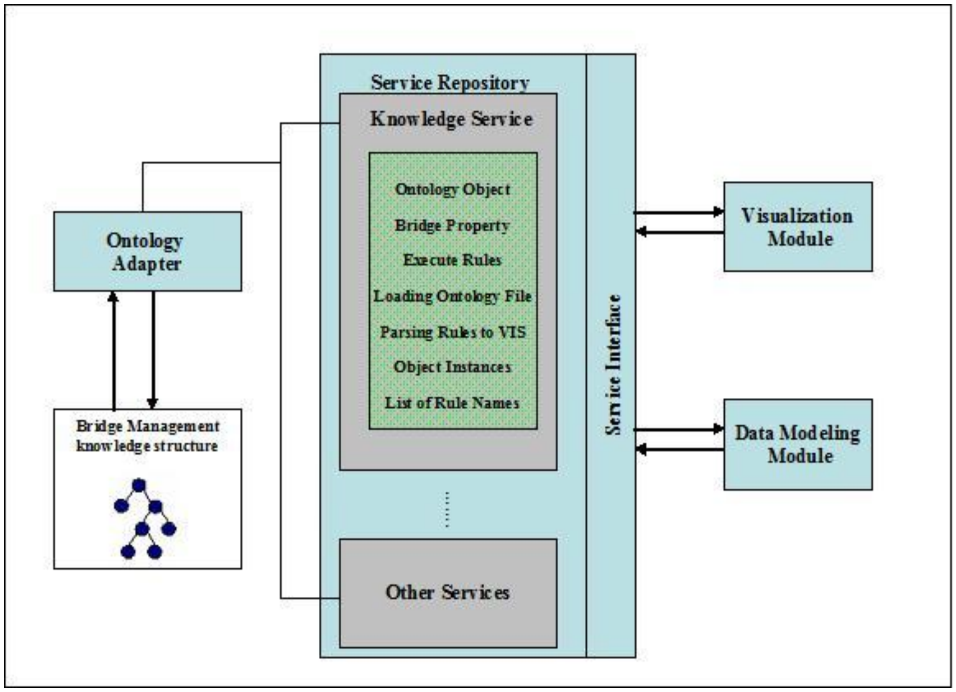


Figure 2.29: Service-Oriented Architecture Framework

Framework and Composition of Services

This section describes a set of system functionalities that can be associated with a particular business goal. This business goal refers to a system or service that will satisfy the need for acquiring knowledge from the required sources based on requests received from other parts of the system, or external systems.

IRSV system comprises of services, which offer a variety of functionalities, including that of providing appropriate piece of knowledge. These services are consumed by other tools or modules within the framework of the IRSV system. In addition to these services and tools or modules, we have other components like the ontology adapter that provides very specific functionality like interfacing with and extracting knowledge from external data structures /sources.

The architectural framework of the IRSV system has been designed so as to bring all these components together in such a way to preserve loose coupling between them, and at the same time exposing all the functionalities of each module to all other modules, without sacrificing on performance or individual functionality. Such an approach to integrating system components makes the system scalable and adaptable to any future modifications/additions to the system.

The system services are defined as web services. The system services can be grouped into a particular business service. From the perspective of development and deployment of code, we are looking at separation of services based on their context. This is very different from an architecture in which modules are grouped based on their source of origin. Modules developed by same development and design team usually get exposed as a single system entity. In our architecture, even though system level modules might be grouped together based on certain pre-determined rules, their functionalities can be referenced as web services and services from different modules can be grouped based on the context of the functionality they provide. This line of thought is especially useful when the system undergoes changes and services are added or removed. Re-wiring the system will be done at the service level, and as web service references are easy to create and modify, this process becomes a lot easier than when one has to write and reorganize explicit references between system modules. Thus, we have modules corresponding to business services, and these modules encompass web services, each of which exposes individual functionalities of the various tools in the IRSV system.

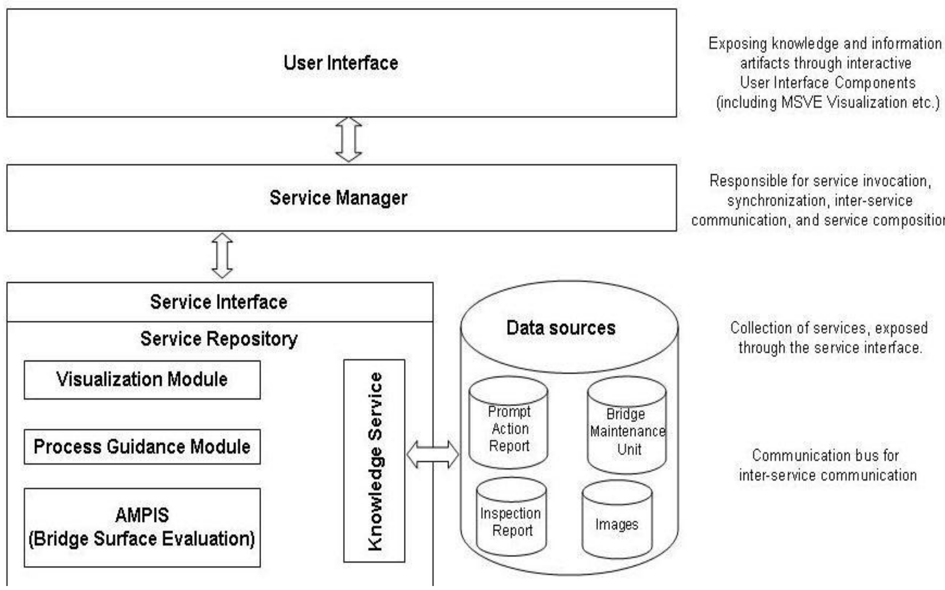


Figure 2.30.Responsibilities of System Modules

To define the business services, it was necessary first to define the processes and sub processes involved in the bridge inspection process. It was also necessary to understand the system services we had at our disposal before we define business services. To define a business service that has no corresponding system services would void the purpose of this process. Also before defining a business service we first have to establish its need and context. This translates into identifying very specific decision points within the bridge inspection process where the bridge inspector can use the IRSV system to provide him/her better understanding of the domain knowledge/data. Based on our interactions with the bridge inspectors and the knowledge acquired by the standard bridge inspection process, we were able to determine the knowledge structure (Lee et. al., 2009) for bridge inspection process. To replicate this exact bridge inspection process would be almost impossible, as different bridge inspectors take a different approach to bridge inspection, which was very much evident from our discussions with the bridge inspectors. To solve this problem, we had to look at the bridge inspection process from a certain level of abstraction that reduces the subjective variations to a minimum, and at the same time preserves purpose and context of the process. Following this approach, it was evident that we could identify certain business goals that were abstract enough to be applicable to any bridge inspection scenario, and there were very specific system services in existence that would answer to these business goals. One such business goal is the need for knowledge, which was satisfied by the knowledge service.

2.3.2 Passing Knowledge Services to VIS Module

The knowledge service, hosted as a web service, contains the services that are described in this section.

As mentioned before, using the SOA paradigm and an ontological engineering framework, the IRSV system will provide a common platform to integrate heterogeneous system components to share bridge data and domain knowledge. For example, one of the modules in the IRSV system is the Visualization (VIS) module which requires knowledge service, a composite of the object, properties and instances from GenOM, the ontology engine. The ontology adapter provides very specific functionality of interfacing with and extracting knowledge from external data structures/sources.

A knowledge service was developed to expose the functionality or the composition of specific functionalities of individual software component as a service. This knowledge service is responsible for representing domain knowledge in a machine understandable manner. The frame-based knowledge representation enables to create meta-knowledge from various types of data such as the sensor data, inspection data, images, and geo-spatial data. A knowledge service provides not only the access to relevant set of data but also the help to understand the nature and correlations among the data set.

The purpose of this knowledge service is to provide a mechanism to other component modules in the IRSV system to infer knowledge from these data sources, which in this case are the ontology and databases, without having to understand the underlying intricacies of the data structures in which they are stored.

For a knowledge service to be truly effective, it has to not only talk in an acceptable language, but also interpret and answer requests for knowledge from other tools in a transparent way. In other words, other modules do not have to understand the ontology to request information from it. The process of transforming the request for knowledge to a query that can be run on the ontology to request information has to be seamless and transparent. Thus, the knowledge services encompass methods calls and underlying logic to perform this transformation to map a request to the actual data in the ontology.

The knowledge service provides information regarding three different areas of the inspection process:

1) The sensor data collected for every bridge
2) Inspection data accumulated for every bridge
3) Information pertaining to bridge inspection processes

Because the ontology encompasses knowledge from different aspects of the bridge inspection domain, the knowledge service comprises of system services that have been composed to query different parts of the ontology. For example, the ontology contains knowledge about types of defects that can possibly affect a bridge. Bridge defect knowledge is a part of the ontology that could be leveraged by bridge managers in analyzing bridge inspection related artifacts and to make recommendations for repair/replacement. Exposing this part of the ontology as a service would hence provide a channel to expose vital bridge defect information to other parts of the tool that might need it. To use the bridge defect service, any tools or module within the IRSV system can subscribe to this service, and use one of its provided methods.

As shown in Figure 2.31, each system service in the knowledge module has a set of functions. Some of these functions are used to test connectivity and availability of service. These functions are essential since we do not support dynamic service discovery in the current version of the IRSV prototype.

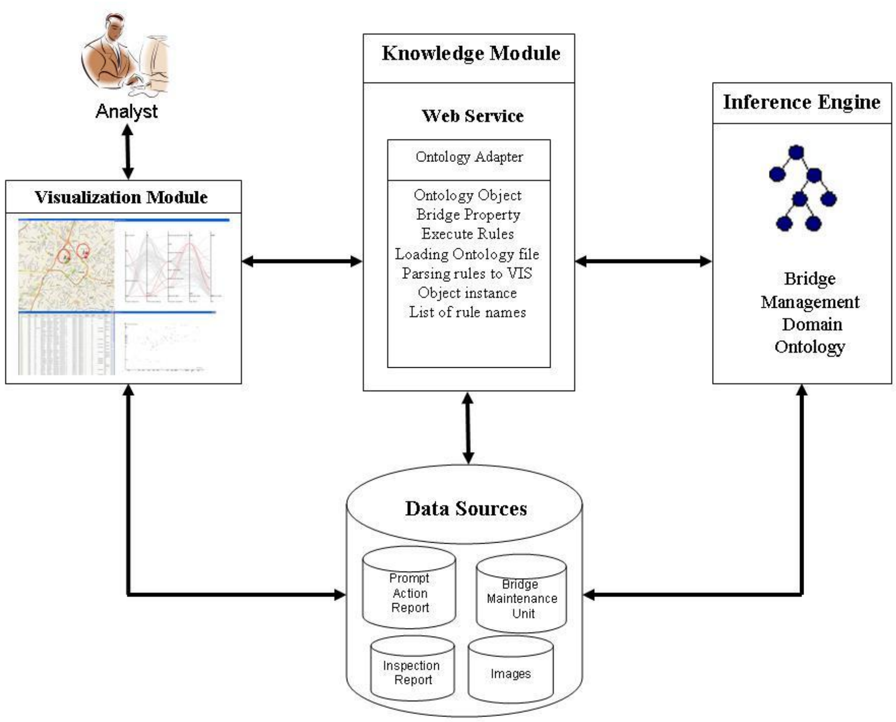


Figure 2.31: Communication Channel among Data Sources, Knowledge Module, and Visualization Module in IRSV

2.3.3 Knowledge Service and Implementation Specification

The implementation and deployment of the services is done in Java API for XML Web Services (JAX-WS) framework (JAX-WS), developed by Sun Microsystems. JAX-WS defines a set of APIs for creating web services, and through the use of annotations, simplifies the development and deployment of web services. NetBeans IDE environment has an extensive support for JAX-WS framework, and hence it was chosen as the preferred tool to create, test and deploy the web services in the JAX-WS framework. NetBeans has an intuitive user interface based service definition console that can be used to create services and define its methods, along with its parameters. Thus, using this user interface based console, one can define the skeleton of a service. The flexibility of a web service driven framework is evident from the fact that these web services are defined irrespective of each others' internal computational logic. The web services can subscribe to each other and pass data between themselves, but this is done using standard web service invocation calls, which can be mapped to any web service based on the requirement and availability.

NetBeans provides an intuitive interface that enables to define a complete web service and all of its bindings. The Web Service Definition Language (WSDL) is generated automatically, and making other web services subscribe to these web services can also be accomplished through the user interface. NetBeans also comes pre-packaged with Glassfish, a web server that can be used to deploy and test the web services. The deployment process is again achieved using the user interface rather than a command line interface which is less intuitive. But one of the most useful features of the JAX-WS framework is the web service interoperability technology (WSIT) APIs, which allow a java developer to create web services and users that can subscribe or talk to other web services created in the .NET framework. This opens up the system architecture of IRSV to a host of opportunities, including interfacing with components that have been developed by third parties and whose functionalities we would like to leverage.

From preliminary analysis, a very urgent and essential need for a source of knowledge for the IRSV system was realized. At various points in the process, tools and the bridge inspector need access to data as well as knowledge inferred from this data. By designing a service that provides access to this data and mechanisms to infer the relevant knowledge, we satisfy the business goal of providing relevant knowledge to the required entities within the system. The purpose of this knowledge service is to provide a mechanism to other tools in the IRSV system to infer knowledge from these data sources without having to understand the underlying intricacies of the data structures in which they are stored. To meet this purpose we provide a bridge problem domain ontology. As mentioned before, GenOM provides APIs to communicate with the ontology. The knowledge service has to interface with this API in order to fetch and parse the required data from the ontology and format it into a data structure that can be understood by other systems. It is necessary to do this, since the API uses data structures that are used internally in GenOM. These object definitions are not understood by other tools in the IRSV system. So transforming the data into a format understandable by all tools is very important.

For a knowledge service to be truly effective, it has to not only talk in an acceptable language, but also interpret and answer requests for knowledge from other tools in a transparent way. Other tools don't have to understand the ontology to request information from it. The process of transforming the tool's request for knowledge to a query that can be run on the ontology to request information has to be seamless and transparent. Thus, the knowledge services encompass method calls and underlying logic to perform this transformation to map a request to the actual data in the ontology. Because the ontology encompasses knowledge from different aspects of the bridge inspection domain, the knowledge service comprises of system services that have been composed to query different parts of the ontology.

As stated before, using the user interface based console, one can define the skeleton of a service. The service needs to be defined in theory before a skeleton of the service could be defined. This includes the type of data it processes, the type of data sources it interacts with, etc. Once this logic is in place and encapsulated into functions, the console can be used to define the service skeleton. After that, one has to write the java code inside each of these functions. The user interface provides functionality to switch between the console-type view and code-view. This helps to quickly review the code and method definitions. Figure 2.50 describes the SOA framework where knowledge service is utilized by different modules, in this case visualization module and data modeling module. Some of the services are used to load the ontology file, test connectivity, and execute inferred rules and knowledge defined in the bridge management knowledge structure. From the figure we can see the functions are exposed to external system components on one side, and on the other side they call ontology adapter. The ontology adapters are a little more complicated in their functionality and design. This complexity stems from the fact that web service communications by default do not support complex user-defined data types. Hence, a web service can return only established data types like strings, array lists, vectors, etc. The ontology adapter acts as an interface between the ontology and the web services. GenOM provides a list of methods to query and modify the ontology.

We implemented a server which is composed of various services and those services are invoked by the client using IRSV prototype user interface. On the server side where the services are implemented, each service contains the business logic that invoked the object, property, features or rules from the GenOM. After the services are created, the WSDL file is generated on the server. With the help of WSDL file, the connection between the server and the client is established. On the client end, the java class `serviceConnector` is implemented where the methods are used to invoke the relevant services from the server and then finally displayed on the user interface with the help of java component.

Deploying a web service requires the deployment of a WSDL file (describes the web service and its contents) and a lot of other procedures, which are all taken care of by the NetBeans IDE. But what is even more significant is that the IDE provides a mechanism by which you can make your current projects subscribe to these web services. You need to create a reference to the web service in your current project. This reference object can then be used to access all the web service methods. The creation of this reference is also very easy and intuitive; you have to know the location of the WSDL file of the web service to which you are subscribing. If that web service was created in the NetBeans, the subscription process is done by pointing the reference to the right service in the projects pane in the IDE.

2.3.4 Knowledge Service Methods Defined

The following is the list of methods offered by knowledge service, hosted as a web service. Distinct software packages utilized in the IRSV development are shown in **`different fonts`** to differentiate COTS software from products developed as part of the IRSV project.

Ontology Object Service

`getOntology`: Object Service defines the concept or object created in the bridge management knowledge structure through GenOM. This service is known as 'getOntology' that enables the consumer to retrieve objects from GenOM file. By using this web service the consumer/client can invoke objects from the Ontology/Knowledge structure of the GenOM file.

Bridge Property Service

`getBridgePropertyNames`: This service enables the consumer/client to invoke properties from GenOM file. With the help of this service we can retrieve properties associated with a particular object from GenOM file. For example if the consumer wants to view the properties of "Bridge Asset Data", he will use `getBridgePropertyNames` service of to achieve this functionality.

Execute Rules Service

`executeRules`: This service allows the user to view the instances of a specific rule that is being selected by him. In the service, a Rule name is passed by the client as a parameter to the service provider (Ontology/Knowledge structure of GenOM). The service provider then looks up for that rule in the GenOM file and return it to the consumer (IRSV Prototype). For instance, when the client passes a rule named "Age of Bridge" as a parameter to the web service via the `executeRules` service, all the instances of that particular rule is displayed to the consumer.

Loading Ontology file Service

`getOntologyfile`: This service reads the Ontology file and loads the GenOM model into memory. The Ontology file resides in the local machine of the server. With the help of this service, the consumer is able to fetch the contents of this file. For example, the list of objects, properties, instances and rules can retrieved from the Ontology file once it has been loaded.

Parsing Rules to VIS Service

`RulesVIS`: This service passes rules to the Visualization module in order to let them view the details of bridges they are exactly looking for. The service that is being offered here by the service provider is to let the consumer view all the rules available in Ontology/Knowledge Structure.

Object Instance Service

`getGenInstances`: This service fetches the instances of a specified object from the GenOM file. The service provider (GenOM file) presents the client the instances of the object he is looking for. For instance, if the consumer wants to view the instances of an object named "Material Type", the service "getGenInstances" will provide the consumer with all the instances associated with the object "Material Type".

List of Rule Names Service

`ListOfRules`: This service lists all the Rules available in the GenOM file. The service being offered here by the Ontology/Knowledge structure is of providing the Visualization module with the list of existing rules from the GenOM file

2.3.5. IRSV User Interface / Prototype

The IRSV User Interface was developed with the business goals to be achieved and the business services to be offered. The user interface and the underlying logic are modeled primarily on the inputs provided from NCDOT and CDOT. The primary focus of IRSV prototype user interface is to combine bridge inspection data and domain knowledge based on the knowledge representation and a goal-driven modeling technique. This prototype will also employ a primitive user interface which helps in exploring and using the functionalities of the services created. Figures 2.32, 2.33 and 2.34 show screenshots of the user interface.

Tabbed Interface and Functions

By using a tabbed interface to depict the workflow of these DOT processes, the prototype tries to associate the system's look and feel to the flow of activities followed by the DOT personnel in inspecting and evaluating bridges.

1. In the first tab *'Data Profile'* as shown in Figure 2.32 (a), the user enters the bridge number (590179) and retrieves the location, source information and aerial images.
2. The user then clicks on the tab named '*Analyze Bridge Data*' as shown in Figure 2.32 (b) where the user views analysis performed by the AMPIS module and LBDA module on the bridge number 590140.
3. The user clicks on the next tab named '*Generate Report'* as shown in Figure 2.33 where the user can view the different ratings for a bridge.
4. In the next tab, '*Inference Engine'* as shown in Figure 2.34 (a) and (b), the user can click on the rules and see the list of bridges that are affected due to the factors mentioned in the established rules.

Tab Data Profile

Search Functionality by bridge number:

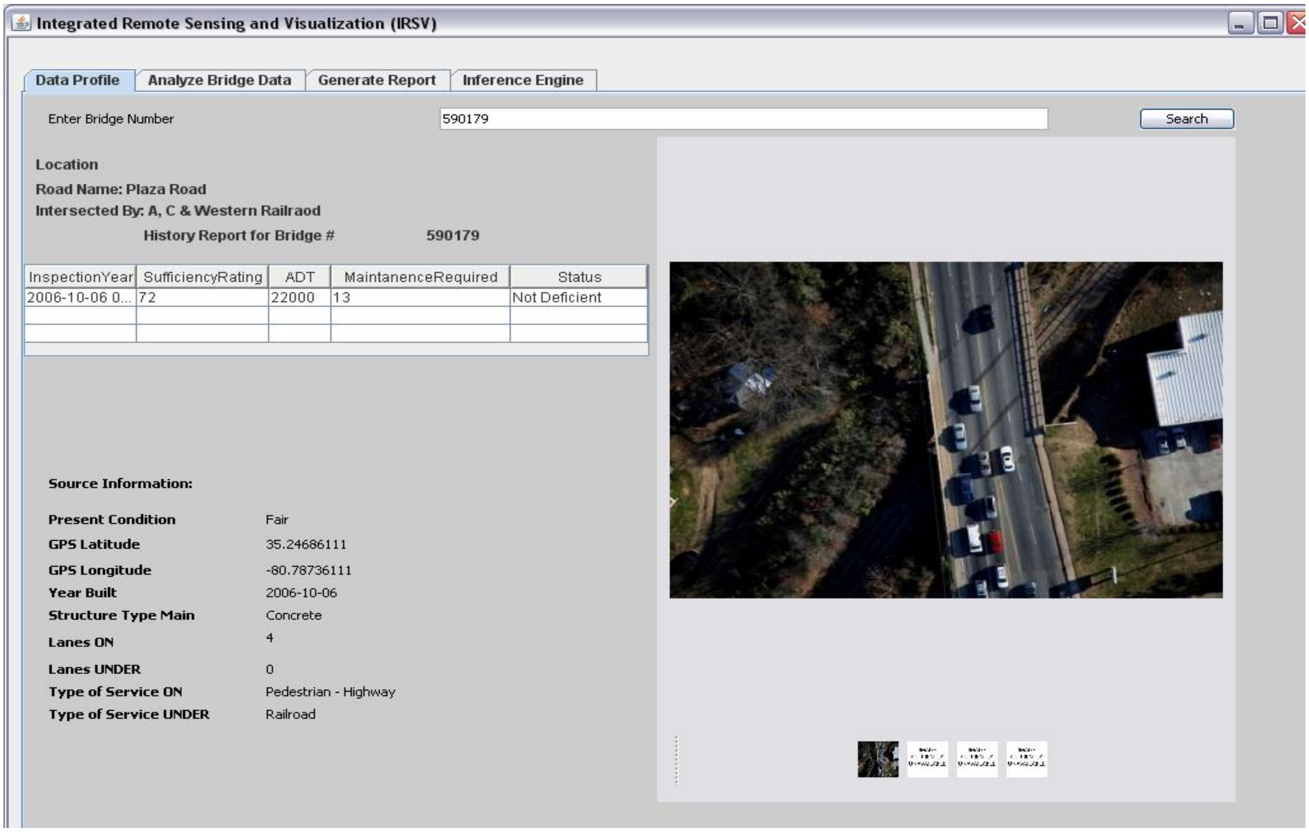


Figure 2.32: (a) Screenshot 1 of IRSV User Interface – Data Profile Tab

Tab Analyze Bridge Data

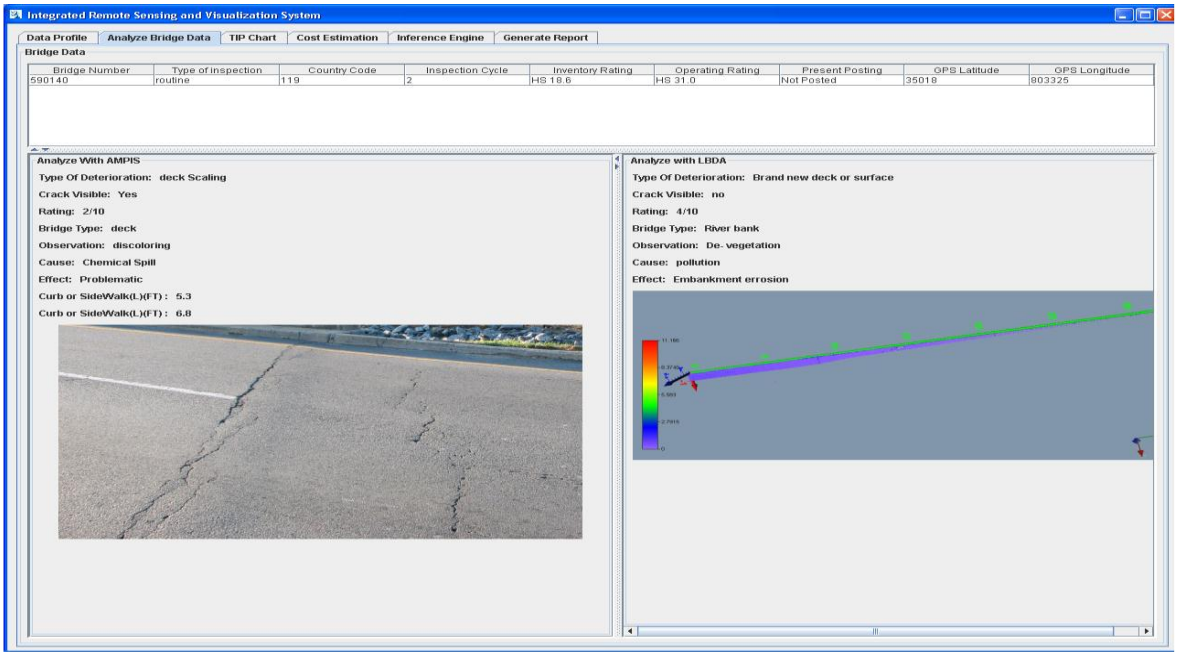


Figure 2.32: (b) Screenshot 2 of IRSV User Interface – Analyze Bridge Data Tab

Tab Generate Report

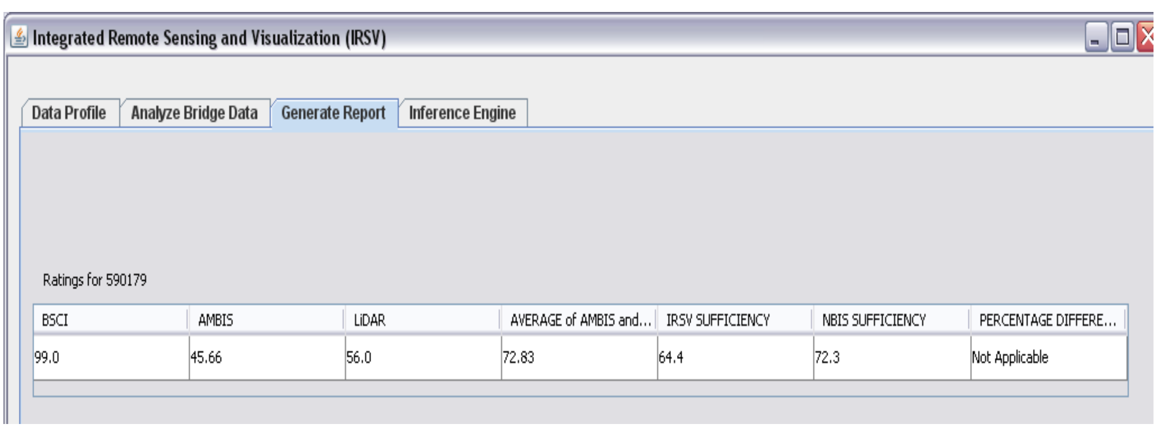


Figure 2.33: Screenshot 3 of IRSV User Interface –Generate Report Tab

Tab Inference Engine

This is the interface to access the bridge knowledge service. The interface is vertically into two areas: (1) Rule library – This provides a listing of rules defined in the ontology by the bridge domain expert. (2) Inference Results – This column displays the bridges identified by bridge numbers that fall under this category.

For example, when the rule structurally deficient bridge is clicked, the corresponding bridge numbers are displayed. In the sample of 20 bridges studied, only 5 bridges are structurally deficient.

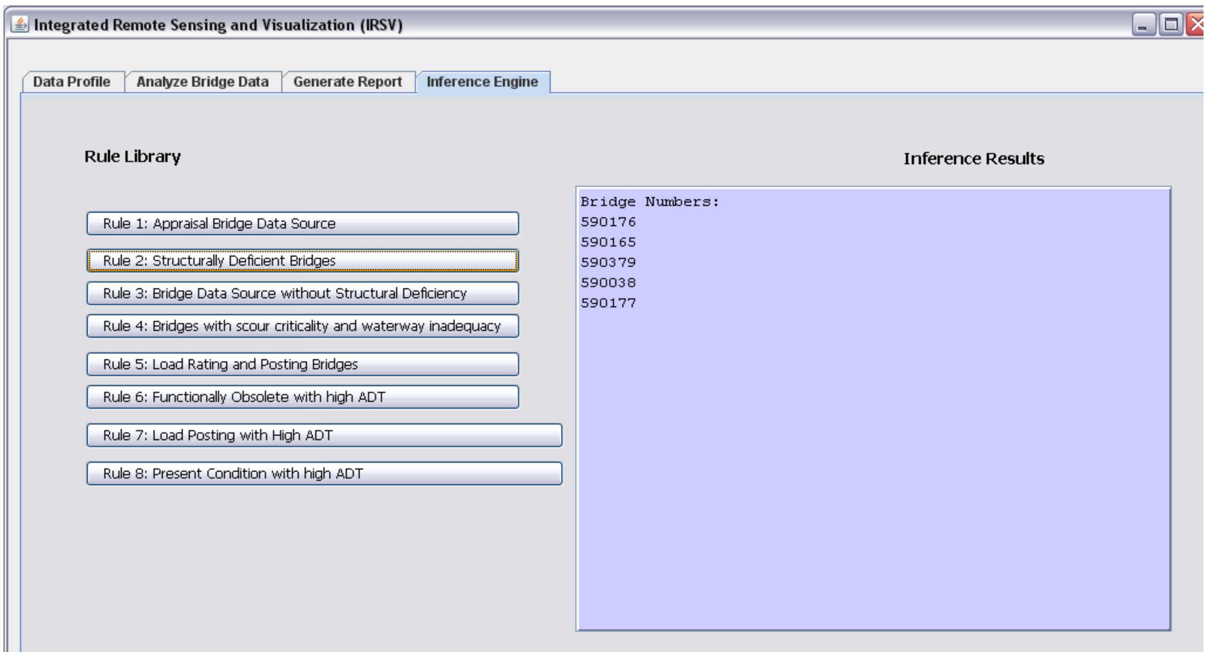


Figure 2.34: (a) Screenshot 4 of IRSV User Interface – Inference Engine Tab

If no bridges fall under a rule, the result displays no bridges found. If specific critical conditions are defined as rules, the user can conclude that no bridges are under the critical condition. For example, in our sample of 20 bridges taken for demonstration purpose, no bridges exhibit both scour criticality and waterway inadequacy.

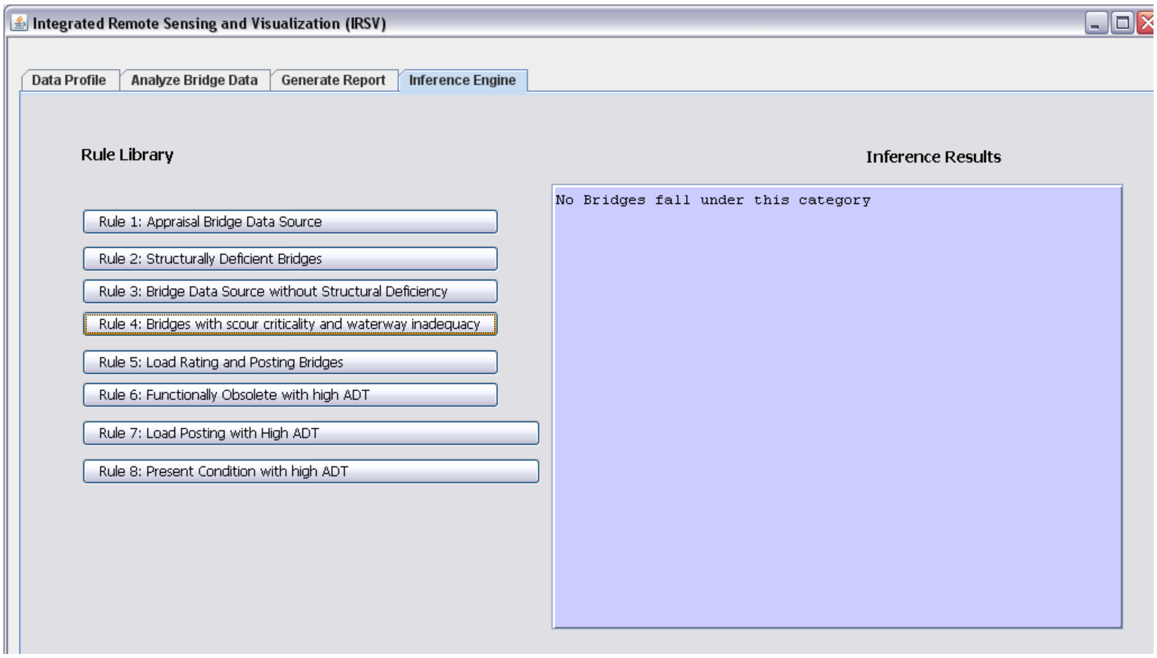


Figure 2.34: (b) Screenshot 5 of IRSV User Interface – Inference Engine Tab

<u>Selected Key Features of IRSV Prototype</u>

- Create data profile for the bridges
- Correlate profiles to existing data sources
- Use the correlation to locate relevant images and other sensor information/inspection reports
- Analyze the available data sources with tools/processes provided by the AMPIS system.
- Based on the evaluation criteria generated by AMPIS module, defects can be tagged with the help of knowledge structure (ontology) and recommend the bridge with the mitigation strategies.
- Logical reasoning can be generated based on what – if conditions through the conceptual space (knowledge structure / Ontology)
- Generate a report that includes the summary of analysis, mitigation strategies with the help of metrics and measures defined in the conceptual space.
- Store this process as a customized process, which can be repeated in the future.

Sample Scenario: By hosting a web service in the knowledge module, other modules can access the services and share the common knowledge and common understanding. For example, the visualization module can successfully invoke the list of rules implemented based on knowledge structure and execute those rules in the platform of visual analytics engine. The following diagram, Figure 2.35, explains how the knowledge service can be used in evaluating the condition of a bridge to determine the appropriate course of action.

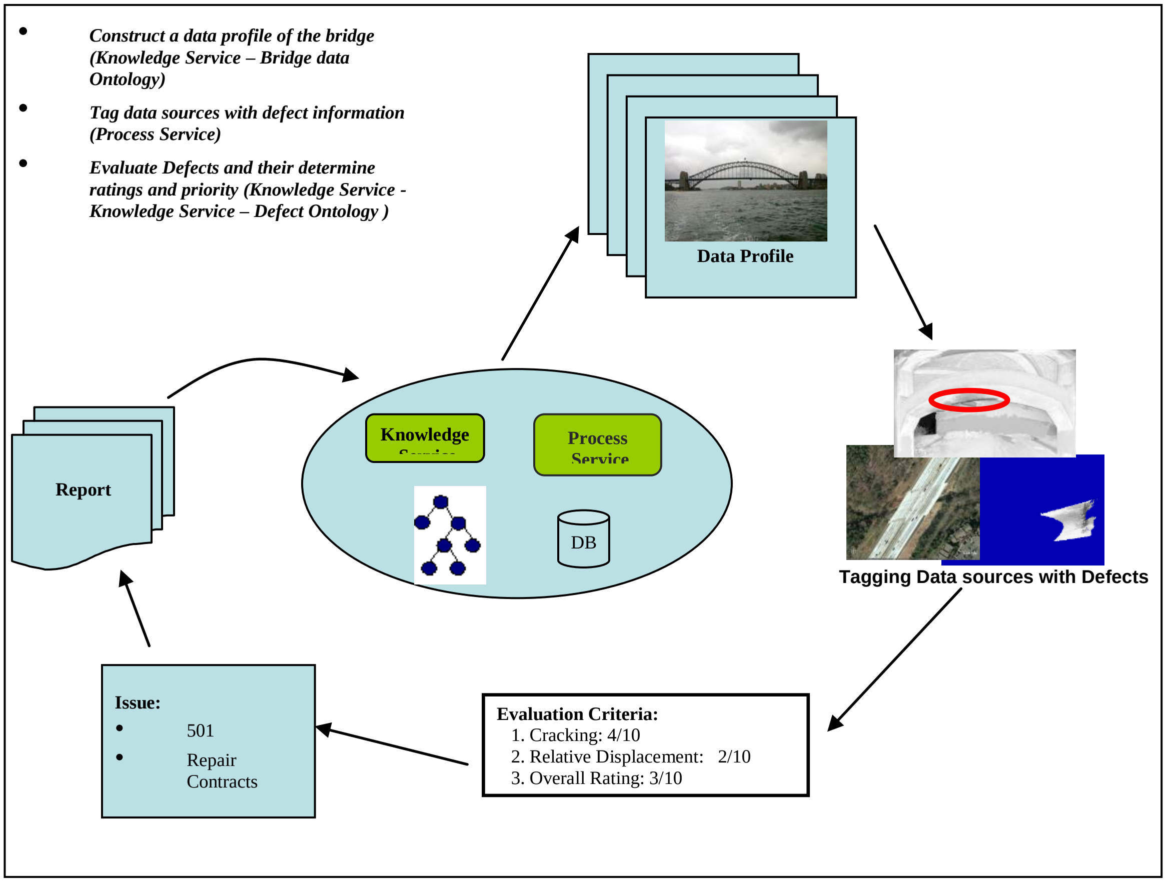


Figure 2.35: Concept of IRSV prototype / user interface

Benefits of IRSV Prototype User Interface

- Exploring IRSV Prototype by using a tabbed interface to depict the workflow of the inspection process by specifying a particular bridge number (e.g., 590140).
- Analyzing the condition of the bridge with the help of historical report and maintain the status of the bridge since the initial inspection cycle. Providing with some relevant source information related to the bridge including LiDAR, satellite, under deck images which describes the defects and condition of the bridge.
- Analyzing the bridge data by invoking the LiDAR bridge defect analysis (LBDA) and AMPIS module. Inspector can view the evaluation criteria performed by

AMPIS and LBDA tools and based on the ratings, measures and metrics the defects can be associated to the bridge with the help of knowledge structure.

- Based on the defects and type of service of the bridge in the knowledge structure (Ontology), cost estimations can be manipulated.
- With the help of defect classifications, material type of bridge, structure type, span classifications of the bridge in the conceptual space (knowledge structure) mitigation strategies will be generated.
- Set of bridges that have similar patterns with respect to defect classification and structure type of the bridge can be concluded through inference engine. For example, Bridge no 590140 indicates bad condition based on evaluation criteria, sufficiency rating, some observation and images. Similarly, list of other bridges can be displayed based on these conditions through knowledge structure. Therefore with the help of inference engine, logical conclusions can be generated based on what-if conditions.
- Generating the TIP Chart and allocate the funding to the bridge that needs to be rehabilitated/ repaired.
- The summary report will be generated based on the analysis performed by LBDA tool, AMPIS module and inference engine.

2.4 Database Structure and Schema

2.4.1 Building a User Interface for NCDOT and Charlotte DOT

This chapter describes the database schema of CDOT and NCDOT which are imported on the SQL Server. Sources of information included Charlotte Dept. of Transportation (CDOT) and North Carolina Dept. of Transportation (NCDOT) representatives (CDOT Rep & NCDOT Rep, respectively). To date, the SIS team has managed to collect and classify data pertaining to the following aspects of the bridge inspection process.

- National Bridge Inventory
- Bridge Maintenance Unit
- Inspection Report
- Prompt Action Report
- Profile Database, which holds customized data that helps inspectors make crucial decisions regarding bridge repairs and bridge ratings.
- Form 501
- Bridge Imagery

CDOT is currently using the Hansen software system, which is an asset management solution, to manage its bridge inspection data. This software system acts as a front-end to an Oracle database management system that stores vast amounts of data relevant to the bridge inspection process. The Integrated Remote Sensing and Visualization (IRSV) system designed and constructed a repository that integrates these data sources with a potential to add more data sources in the near future. Requirement gathering is an ongoing process. Several meetings with CDOT Rep and NCDOT Rep have occurred. Lengthy, focused requirements gathering sessions have occurred and will continued to be scheduled. For example, in the following, we summarize the outcomes of one three hour session with CDOT representative.

On November 27, 2007, the SIS team visited the CDOT facility in Charlotte. The purpose of this meeting was to educate ourselves about the bridge inspection methodologies and processes that CDOT follows. Our aim was to document as much of the domain information as possible and also understand the needs of CDOT with regards to our IRSV system. In summary:

- Tasks handled by CDOT Rep:
  a) After the inspection reports are issued back to CDOT, CDOT Rep feeds the inspection data into the Hansen system.
  b) CDOT Rep participates in the assessment of inspection reports to generate a list of work orders (prioritized based on budgetary and other concerns)
- Preparation tasks for consultant
  a) Consultant has to perform and evaluate the asset (bridge).
  b) Contract for repair work is formulated and issued.
  c) Contractor who is awarded the contract begins work.

- It takes 2 to 3 months of interactions with bridge inspector to finalize the list of repairs and prepare an assessment of the CDOT asset.
- When inspection reports are generated and sent to the CDOT Rep's office, all the reports are analyzed and then based on the collective understanding obtained from this analysis, work orders are generated and prioritized. This collective analysis also helps the CDOT officials look at the bigger picture and take decisions that might help increase efficiency and reduce costs which impacts more than one work order.
- Every year, the to-do list (list of work orders) is prioritized. The activities from the list that are not prioritized for that fiscal year due to budgetary constraints get carried onto the next year's list. Currently, there are no procedures or technology in place to ensure this process is error-proof. In addition, there currently is no traceability for this process.
- CDOT Bridge inspectors need to be NBIS certified. They have a set of coding guidelines to rate bridges. Inspectors can be:
  a) City Staff (working for CDOT)
  b) Consultants approved by the state
  c) State inspectors.
- Artifacts CDOT Rep deals with on a daily basis:
  a) Bridge inspection reports
  b) Work orders for repairs to bridges
  c) Databases containing bridge inventory and inspection data
  d) Architecture plans for bridges
- Figure 2.36 shows the two-year lifecycle of the bridge inspection process for CDOT.

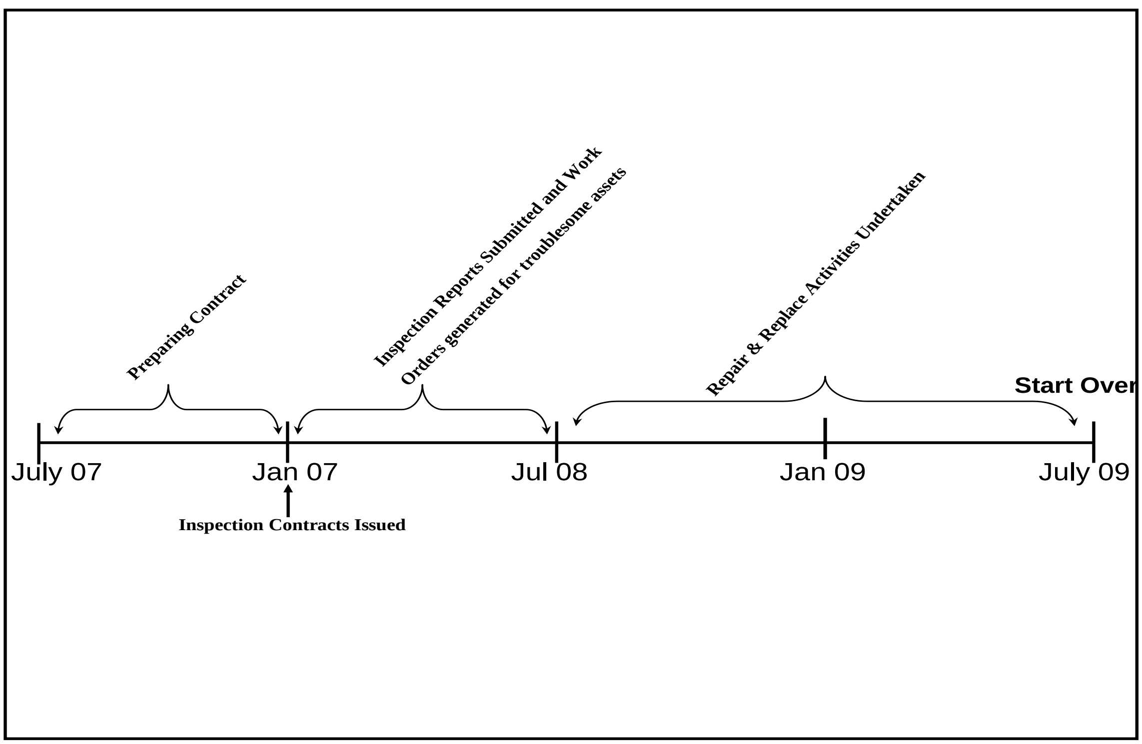


Figure 2.36: Schematic Representation of CDOT Bridge Inspection Lifecycle.

The data collected from the domain experts (CDOT Rep and NCDOT Rep) were organized into tables. The names and structure of these tables were in accordance with the data classification mentioned above.

The two most important and extensive of these tables are the National Bridge Inventory table, which will be used for maintaining bridge inventory and bridge appraisal information, and the Inspection Report table, which will be used for storing all information pertaining to the inspection of a bridge as obtained from the bridge inspection reports.

2.4.2 Using Database Views

From the tables of raw data we have accumulated, we can construct logical and structured data units using the concept of database views. A view is a virtual or logical table composed of data borrowed from one or more of the existing tables.

For example, a bridge manager might want to look at certain set of attributes for a given bridge. To facilitate this, we can compose a view of the required data from the necessary tables. Consider the following example based on the tables that we have built previously where we want to select a certain set of attributes from the Inspection Report, Form 501, and Imagery tables.

We can accomplish this task by creating a View, called ‘BRIDGE_CONDITION,’ which can be represented in the query form as stated below:

```
CREATE VIEW BRIDGE_CONDITION AS SELECT R.PRESENT_CONDITION,
R.MAINTENANCE_REQUIRED, R.SUFFICIENT_RATING, R.STATUS,
R.STRUCTURE_TYPE_MAIN, R.TYPE_INPECTION, I.IMAGE_LOCATION,
F.BRIDGE_NUMBER, F.COUNTY, F.MUNICIPALITY, F.DIVISION_NUMBER,
F.DISTRICT_NUMBER FROM INSPECTION_REPORT R, IMAGES I, FORM_501 F
WHERE R.BRIDGE_NUMBER = F.BRIDGE_NUMBER AND R.IMAGE_ID = I.IMAGE_ID
AND R.STRUCTURE_TYPE_MAIN = ‘PRESTRESSED CONCRETE’;
```

2.4.3 Database Schema

From the tables that have been formulated, the following initial database schema as shown in Figure 2.37 can be conceptualized. During the second and third quarters of FY2008, we continued to extend and refine this schema to account for the data required and acquired.

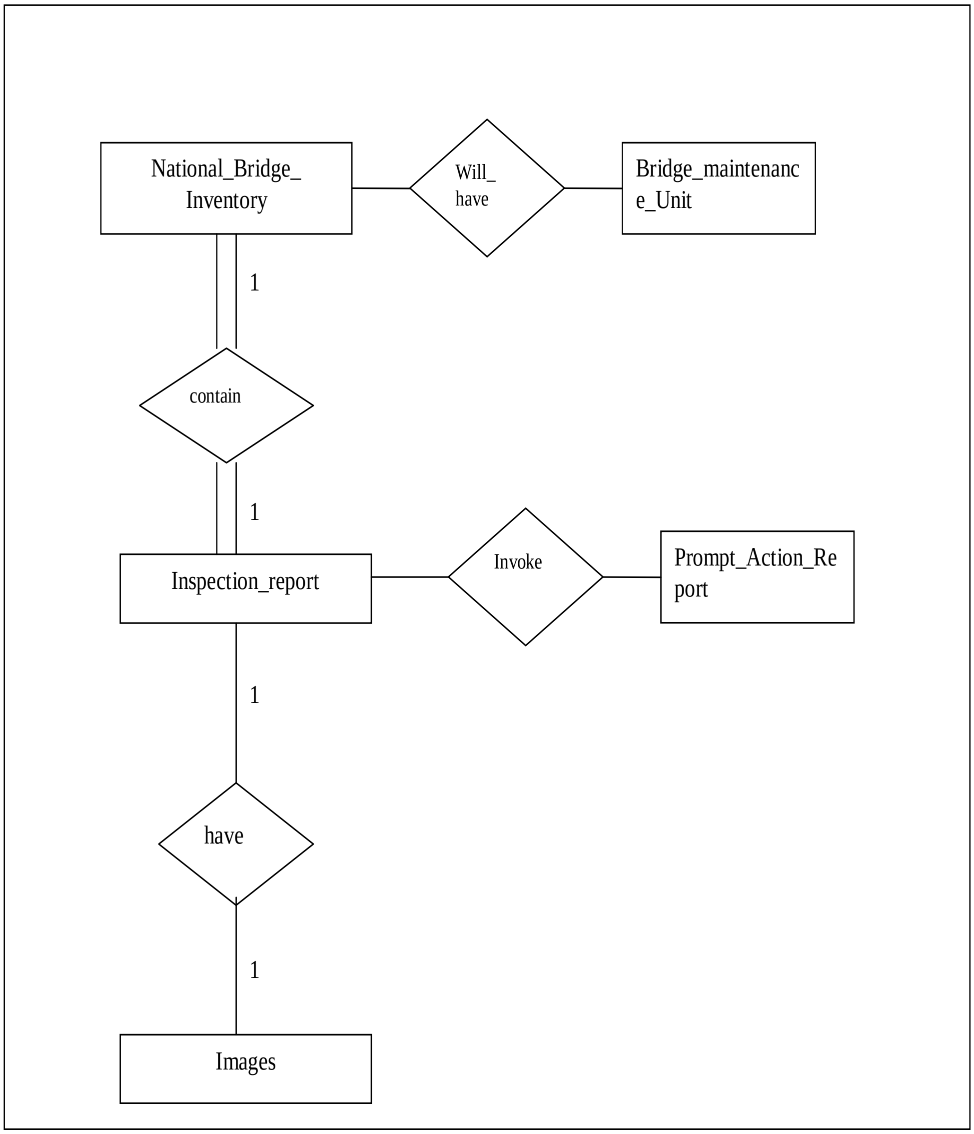


Figure 2.37: Database Schema for the IRSV System.

Following is the database schema representation with the attributes / properties and values of the attributes:

Tables in DOT database

Table Name	Represents:
National Bridge Inventory	Structure inventory and appraisal
Bridge Maintenance Unit	Repair or maintenance
Inspection Report	Inspection details (all states have standardized report of this table)
Prompt Action Report	What needs to be done for the bridge
Profile Database	Personal database
Form 501	Bridge maintenance supervisors report for new and rebuilt bridges (maintenance completed)
Images	Aerial Photo images, LiDAR images, AMBIS interpreted data

Database Tables with Attribute Listing

National Bridge Inventory

Column Name	Data Type
Bridge Number	Int
County	Varchar2
Municipality	Varchar2
Present Condition	Varchar2
Route	Varchar2
Intersected	Varchar2
Location	Varchar2
GPS Latitude	double decimal
GPS Longitude	double decimal
Maintenance Required	Int
Sufficiency Rating	Float
Status	varchar2
Structure Type Main	Varchar2
Wearing surface	Varchar2
Inspection Date	Date
Year built	Date
Year Reconstructed	Date
Type of Service (on)	Varchar2
Type of Service (under)	Varchar2

Lanes (on)	Int
Lanes (under)	Int
ADT	Int
Y of ADT	Int
Truck ADT FCT	Int
Bypass Detour Length (MI)	Int
Length of Maximum Span (FT)	Int
Structure Length (FT)	Int
Curb or sidewalk (L)(FT)	Float
Curb or sidewalk (R)(FT)	Float
Bridge Roadway Width Curb to Curb (FT)	Float
Deck Width Out to Out (FT)	Float
Approach Roadway width (FT)	Int
Bridge Median	Int
Skew (DEG)	Int
Structure Flared	Char
Inventory Route Min Vert Clear	Varchar2
Inventory Route Total Horiz Clear (FT)	Float
Min vert Clear Over Bridge RDWY	Varchar2
Min Vert Underclear REF	Varchar2
Min LAT Underclear RT REF (FT)	Float
Min LAT Underclear LT (FT)	Float
Deck	Int
Superstructure	Int
Substructure	Int
Channel & Channel Protection	Varchar2
Culverts	Char
Design Load	Int
Operating Rating	Varchar2
Inventory Rating	Varchar2
Bridge Posting	Int
Structure Open, Posted, or Closed Description	Varchar2
Structural Evaluation	Int
Deck Geometry	Int
Underclearance, Vertical & Horizontal	Varchar2
Waterway Adequacy	Varchar2`
Approach Roadway Alignment	Int
Traffic Safety Features	Int
Scour Critical Bridges	Char
Type of Work	Varchar2
Length of Structure Improvement (FT)	Int
Bridge Improvement Cost ($)	Int
Roadway Improvement Cost ($)	Int
Total Project Cost ($)	Int

Y of Improvement Cost Estimate	Int
Future ADT	Int
Y Future ADT	Int

Form 501

Column Name	Data Type
Bridge Number	Int
County	Varchar2
Town/City	Varchar2
Division Number	Int
District Number	Int
File Number	Int
Date Completed	Date
Route on the bridge	Int
Feature	Char
Location	Varchar2
Miles	Int
Miles_Of	Int
Miles_towards	Int
Skew Angle	Int
Grade	Char
Overall length	Int
Clear roadway	Char
Roadway between the rails	Char
Sidewalk width left	Int
Sidewalk width right	Int
Deck Width	Int
Roadway width	Int
Shoulder width left	Int
Shoulder width right	Int
Height, top of floor to streambed	Int
Min vertical clearance	Int
Min horizontal clearance	Int
Lateral under clearance	Int
Span length	Varchar2
Type_of_superstructure	Varchar2
Rail Type	Varchar2
Wearing surface description	Varchar2
Floor Description	Varchar2
Stringer Details	Varchar2
Abutment type	Varchar2
Abutment cap	Varchar2
Abutment piles or support	Varchar2

Interior bent types	Varchar2
Bent cap	Varchar2
Bent piles/ support	Varchar2

Bridge Maintenance Unit

Column Name	Data Type
Division Number	Int
District Number	Int
Type of superstructure	Varchar2
Type of substructure	Varchar2
Floor	Varchar2
Span length	Int

Prompt Action Report

Column Name	Data Type
Bridge Number	Int
Type of work	Varchar2
Length of structure improvement	Int
Bridge Improvement Cost	Double Decimal
Roadway importance Cost	Double Decimal
Total Project Cost	Double Decimal
Y of improvement Cost Estimate	Int
Future ADT	Int
Y Future ADT	Int

Profile Database

Column Name	Data Type
Bridge Number	Int
Road Name	Varchar2
Route	Int
Intersected	Varchar2
Location	Varchar2
Highway System	Varchar2
Sufficiency Rating	Float
Status	Varchar2
Type of Superstructure	Varchar2
Year Built	Int
ADT	Int
Structure length	Int
Deck Width out to out	Int

Inventory Route Min Vert Clear	Varchar2
Inventory Route Total Horiz Clear (FT)	float
Superstructure	Int
Substructure	Int
Channel & Channel Protection	Varchar2
Structure Open, Posted, or Closed Description	Varchar2
Crown ht	int

Bridge Inspection Report

Column Name	Data Type
Municipality	Varchar2
Type Inspection	Varchar2
CountyCode	Varchar2
Bridge number	Int
Inspection cycle	Int
Route	Varchar2
Across	Varchar2
MP	Char
Location	Varchar2
Description	Varchar2
Present condition	Varchar2
Inventory rating	Varchar2
Inspection Date	Date
Operating Rating	Varchar2
Present Posting	Varchar2
Proposed posting	Varchar2
Computer update	Varchar2
Analysis date	Varchar2
Posting letter date	Varchar2
GPS Latitude	Double decimal
Other signs present	Varchar2
GPS Longitude	Double decimal
Special permit	Varchar2
Road Name	Varchar2

The knowledge modeling team has reorganized and developed the structure of database schema in SQL Server 2005. In coordination with the team’s civil engineers, knowledge modeling team has managed to import the data for three-year cycle i.e. 2000, 2004 and 2006. The Inspection_Report_CDOT table contains the three-year record for CDOT of two-year cycle, i.e., 2000, 2004 and 2006, and the Inspection_Report_NCDOT table contains around 300 bridges for NCDOT. The image table contains LIDAR images, digital images, air-borne images, etc. These data are imported in SQL Server.

2.5 Deployment of Modules on Client System

This chapter describes about the deployment of the knowledge modeling (SIS) modules on the client machine. It also describes the functionalities of each module and how it works.

2.5.1 System Requirements

- Windows XP Professional
- SQL Server
- MySQL Driver
- Java Virtual Machine
- Java API for XML Web Services (JAX-WS) framework

2.5.2 GenOM

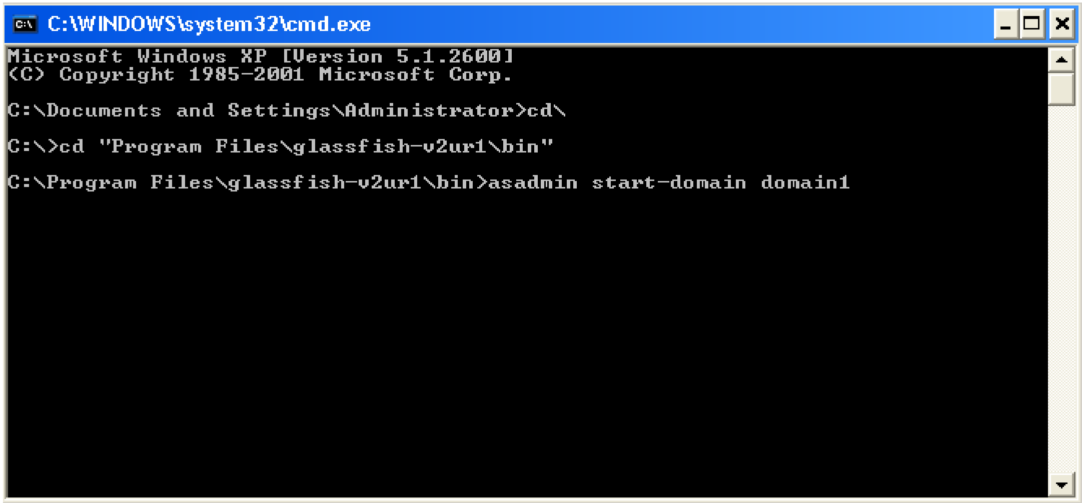
On the client machine, the following files are loaded and made available for access to the other modules.

- Executable jar files
- Library files
- bridgemanagement.owl and bridgemanagement.genom

2.5.3 Web Service

The Knowledge.jar file is loaded on the server machine. Perform the following steps to deploy and host the knowledge service.

1. Install Glassfish
2. Put war file in C:/programfiles/glassfish/domains/domains1/autodeploy
3. Go to Start menu
4. Click on Run
5. A dialog box will appear, then type ‘cmd’ in that dialog box and click Ok
6. Then type ‘cd\’ and press enter
7. Then type ‘cd "Program Files\glassfish-v2ur1\bin"’and press enter
8. Then write ‘asadmin start-domain domain1’ and press enter
9. The Server will start running



<u>Testing the Web Service</u>
Copy the following link on the browser. Replace local host with the hosting machine's IP address if other systems need to access.
http://localhost:8080/KnowledgeService/getBridgeDataService?wsdl

2.5.4 IRSV Prototype and Database

The IRSV Prototype containing the jar files and libraries are loaded on the client machine. Netbeans IDE is used to open and execute the IRSV Prototype. MySQL Driver (provided as jar file) is chosen to connect to the database.

In coordination with the Civil Engineering Department team, the Software and Information Systems has collected and organized domain information pertaining to the bridge inspection process. Sources of information included Charlotte Dept. of Transportation (CDOT) and North Carolina Dept. of Transportation (NCDOT) representatives. To date, the knowledge modeling team has managed to collect and classify data pertaining to the following aspects of the bridge inspection process.

- Inspection_Report_CDOT
- Inspection_Report_NCDOT
- Images

The data collected from the domain experts are organized into tables. The names and structure of these tables are in accordance with the data classification mentioned above. The Inspection_Report_CDOT table contains the three-year record for CDOT of two-year cycle, i.e., 2000, 2004 and 2006, and the Inspection_Report_NCDOT table contains around 300 bridges for NCDOT. The image table will contains LIDAR images, satellite images, air-borne images, etc. These data are imported in SQL Server.

2.5.5 IRSV System Control Policy and Approach

2.5.5.1. System Security

IRSV is designed as a customized, client-based bridge data visualization/management system. The critical elements in IRSV include: 1) bridge information, 2) data management system, 3) data acquisition/analysis processes, 4) personnel involved and 5) service environment. Development of a systemic security plan, the individual clientele must be engaged and dictate the security design including establish the security objectives, defining the control policies and outlining the hierarchy of security measures. U.S. departments of transportation (DOTs) have different security practices/policies; hence, the IRSV system security should be consistent with the sovereignty of the DOT security objectives.

It must be recognized that with the integration of the proposed specific CRS (Commercial Remote Sensor) data (aerial imaging and LiDAR scan) and SI (Spatial Information) technology integration, IRSV may have significant implications to national security that are not encountered in current bridge inspection practices. The security issue is related at both information and software levels.

Information security and software security can be vastly different issues. Software security is "the idea of engineering software so that it continues to function correctly under malicious attack" (McGraw, 2004). For software security, several measures can be established during the life cycle of a software development as shown in Figure 1. Measures typically include establishing a clear security objective and design the software around the objective. Risk-based security tests can be performed during software development to ensure the end-product measures up to the security objective. Proper and efficient feedback systems can be established allowing reporting of security breaks back to the code developers to revise and update the software system.

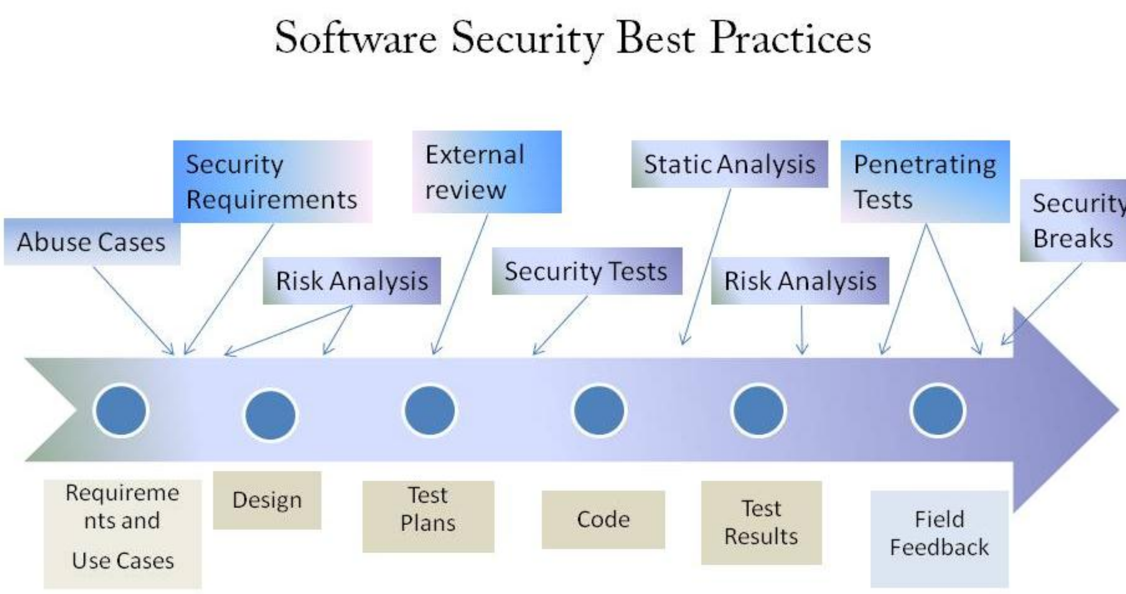


Figure 2.38 Software security measures during development cycle (McGraw 2004)

Information security, on the other hand, refers to the protection of critical information and information systems from unauthorized access, use, disclosure, disruption, modification or destruction (U.S. Code 2009). Similar to software security, effective information security relies on measures such as access protocols and/or cryptographic control, internal/external security assessments, education/awareness building, standalone hardware, and information infusion segregation, etc. It is important to recognize that information security breaches can happen and that risk must be identified.

Our approach to establishing security measures begins with first identifying the human elements that are most likely to result in a security breach. Figure 2.39 shows the human elements identified that may engage in the use of IRSV. The operators are bridge engineers, software developers and DOT bridge database managers. Invited users can be bridge inspectors, subcontracts and researchers. Uninvited users are intruders who may or may not have an ulterior motive to sabotage the system. IT security personnel may vary from DoT to DoT.

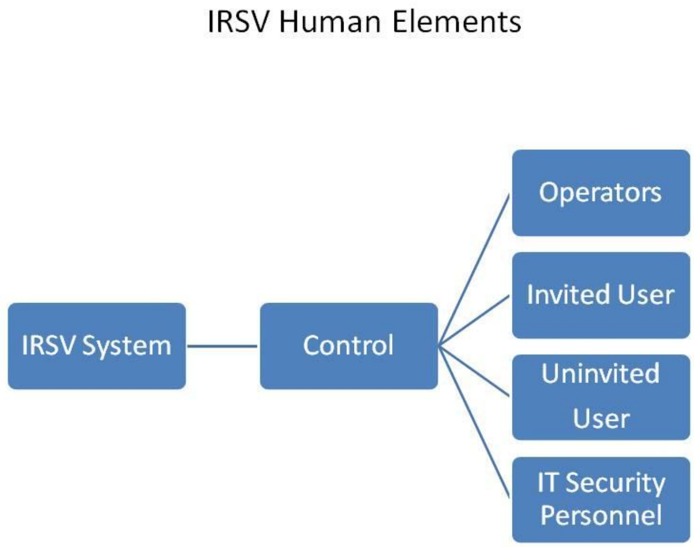


Figure 2.39 Human elements involved in IRSV security

2.5.5.2. IRSV Security Recommendations

To ensure the security of IRSV, several best practices recommendations have been reviewed and an ICE (Identify, Communicate and Establish) strategy, which encompasses the following 9) suggestions and is mostly condensed from ISO-27002-2005 (ISO 2007), is recommended:

1) Identification
   a) Identify key players: who will be involved in the security measures (i.e., security officers, bridge managers, IT personnel, bridge inspectors, software vendors, bridge maintenance engineers, subcontractors, and the general public) who may have an interest in the bridge information, etc.

b) Identify critical information: bridge data, such as high-resolution aerial photos, that are pertinent to possible structural details that may be used for sabotage planning or other abuses.
c) Identify possible information exchange processes: what are the processes that may allow computer viruses or hacker attacks?
d) Identify security technologies: passwords, access card to hardware, limited wireless access, cryptographic controls, are few potential technologies.
e) Identify business operation continuity processes: if failure occurs, the critical business continuity management process that would minimize loss and resume operation (restoration) quickly needs to be identified.
f) Identify risk potentials and management approaches: if failure occurs, determine anticipated loss to information, DOT and public.
g) Identify potential threats.

2) Communication (education/awareness)
   a) Communicate importance of security objectives to all key players.
   b) Communicate the need of security to all players.
   c) Communicate responsibilities and agreements to all key players.
   d) Communicate importance of incident reports, potential attacks.
   e) Communicate good access practices to all players.

3) Establishment
   a) Establish comprehensive security objectives;
   b) Establish internal security organization;
   c) Establish external audits;
   d) Establish access control processes;
   e) Establish asset management objectives;
   f) Establish key player responsibilities;
   g) Establish education/awareness program.
   h) Establish service delivery compliances;
   i) Establish network security;
   j) Establish incident management processes.
   k) Establish protocol to security bleach/incident reporting.

2.5.5.3. IRSV Prototype Security Provisions

All state and federal publications on bridge inspection and management practices (AASHTO 1994-2003, Hearn 2007) do not have specific recommendations on the security measures of bridge management systems. It appears that most DOT IT (Information Technology) specialists are responsible for establishing the information security measures. In the case of North Carolina

DOT, information and software security are warranted predominantly through access control and hardware integration control.

The current IRSV prototype system security protocol is established with minimal access rights (rights limited only to bridge managers accessed via a single workstation). The installation process of the IRSV is password protected to prevent unauthorized installs. The security of the IRSV system is provided by the overarching sovereignty of DOT IT security policies where IRSV authentication support, access control, audit logs, etc. will depend on DOT IT security solutions rather than IRSV-specific practices and procedures. Figure 1 shows the IRSV prototype security relationship and measures. Physical access to IRSV installed workstations will be controlled locally at DOT sites. Cyber-access to IRSV installed workstations will be controls by DOT IT system security. Thus, even though the database (including all image files) that are not encrypted, access to these data are controlled at the physical and cyber levels by DOT IT policies and security solutions. This will allow for the diverse security policies and systems that are anticipated across various DOTs. In fact, it is anticipated that enhanced security measures will be custom designed for each clientele during system implementation.

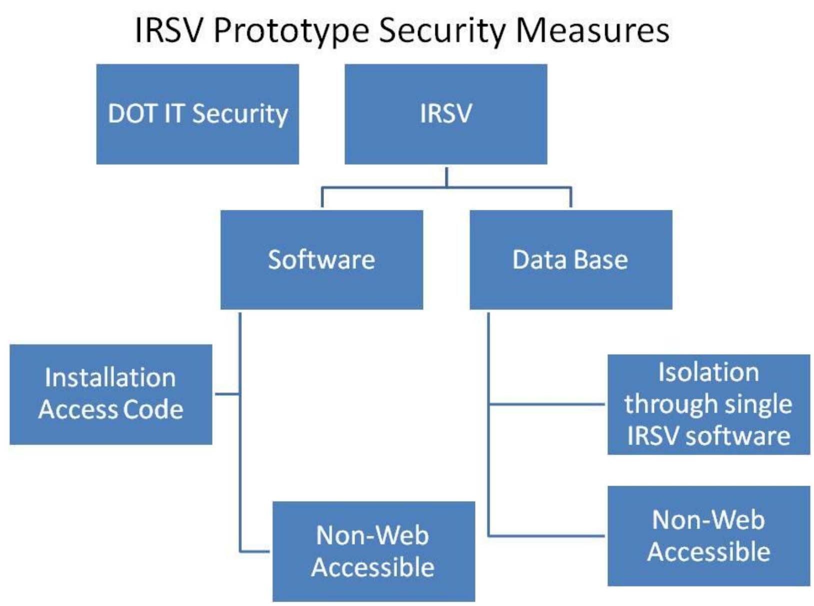


Figure 2.40 Security measures for IRSV prototype system

2.6 Summary and Conclusions

The IRSV system benefits from captured process and assessment knowledge and uses them to support bridge managers in decision making. Also, bridge managers benefit from domain knowledge understanding and the representation of conceptual space by introducing ontology-based semantic matching and mapping the properties of the conceptual space to the instance space. It also provides responses to "what-if" queries from system behaviors through matching various initial conditions and circumstances based on rules in domain model. With the help of SOA, we capture important knowledge and make it available for other system modules. The integration framework provides an opportunity to build a system that can scale and adapt to incorporate evolving processes and technologies and in addition service framework can mediate between various system components, knowledge and process services and can provide right information at the right time.

Appendix A. Bibliography

Lee, S.W., Muthurajan, D., Gandhi, R.A., et al., “Building decision support problem domain ontology from natural language requirements for software assurance,” International Journal on Software Engineering and Knowledge Engineering, Vol. 16, No.6, pp.1-34, December, World Scientific, 2006.

Seok-Won Lee, Robin Gandhi and Siddharth Wagle “Ontology-guided Service-oriented Architecture Composition to Support Complex and Evolving Process Definitions” To appear in the International Journal of Software Engineering and Knowledge Engineering, World Scientific, Vol. 19, No.6, September 2009

Gandhi, R. A., Siddharth, W., and Lee, S.W., "Process Artifacts Defined as an Aspectual Service to System Models" In Proceedings of the 2nd International Workshop on Service-Oriented Computing: Consequences for Engineering Requirements (SOCCER'06), 14th International Requirements Engineering Conference (RE'06), September 11-15, 2006 Minneapolis/St. Paul, Minnesota, USA.

Souripriya Das, Eugene Inseok Chong, George Eadon, Jagannathan Srinivasan, “Supporting-ontology based Semantic Matching in RDBMS.”

Lipyeow, Lim Haixun Wang, Min Wang. “Semantic Data Management: Towards Querying Data with their Meaning”

Jagannathan Srinivasan , Ravi Murthy, Seema Sundara, Nipun Agarwal, Samuel DeFazio *Oracle Corporation*, “Extensible Indexing: A Framework for Integrating Domain-Specific Indexing Schemes into Oracle8*i*”

Generic Object Model (GenOM) for Software Computing – NiSE Project

Java API for XML Web Services (JAX-WS) – Sun Microsystems

Web Service Interoperability Technology (WSIT) – Sun Microsystems

Web services tutorials using the JAX-WS framework – NetBeans 6.0/6.1

Alonso G., Casati F., Kuno H. and Machiraju V., [*Web Services-Concepts, Architectures and Applications*], Springer-Verlag, Berlin Heidelberg (2004).

Papazoglou M. and Georgakopoulos D., Service-oriented computing," in [*Communications of the ACM (CACM) 46(10)*], (October 2003).

Perrey R. and Lycett M., Service-oriented architecture," in [*Proc. of Symp. on Applications and the Internet Workshops (SAINT '03)*], 116-119 (2003).

Lim L., Wang. H. and Wang M., Semantic data management: Towards querying data with their meaning," in [*Proceedings of the IEEE 23rd International Conference on Data Engineering (ICDE 2007)*], (April 2007).

Lee S. W. and Gandhi R., Ontology-based active requirements engineering framework," in [*Proc. 12th Asia-Pacific Soft. Engg Conf.*], 481-490 (2005).

Lee S. W., Gandhi R., Wagle S. and Murty A., r-analytiCA: Requirements analytics for certification & accreditation," in [*Proc. of the 15th IEEE Int. Requirements Engg. Conf. (RE 07) Posters, Demos and Exhibits Session*], (October 2007).

Lee, S. W. and Yavagal D., Genom user's guide v2.0," Tech. Rep. TR-NiSE-05-05, UNC Charlotte (2005).

Wong P. and Thomas J., Visual analytics," in [*IEEE Computer Graphics and Applications, Vol.24(5)*], 20-21 (2004).

Hansen information technologies - bridge management solution. http://www.hansen.com/solutions/transportation/bridge-management.asp.

Pontis - bridge management system. http://aashtoware.org/?siteid=28&pageid=74.

Vatcha R., Lee S.W., Murty A., Tolone W., Wang X., Dou W., Chang R., Ribarsky W., Liu W., Chen S.E., Hauser E., "Towards Sustainable Infrastructure Management: Knowledge-based Service-oriented Computing Framework for Visual Analytics" in Proceedings of the Visual Analytics for Homeland Defense and Security (Conference DS219), Part of program track on Information Systems and Networks: Processing, Fusion, and Knowledge Generation, SPIE Defense, Security and Sensing, 13-17, April, Orlando, Florida, USA. 2009.

Wang X., Dou W., Vatcha R., Liu W., Chen S.E., Lee S.W., Chang R., and Ribarsky W., "Knowledge Integrated Visual Analysis of Bridge Safety and Maintenance" in Proceedings of the Visual Analytics for Homeland Defense and Security (Conference DS219), Part of program track on Information Systems and Networks: Processing, Fusion, and Knowledge Generation, SPIE Defense, Security and Sensing, 13-17, April, Orlando, Florida, USA. 2009.

Appendix B. List of Acronyms and Definitions

AADT – Average Annual Daily Traffic
AASHTO - American Association of State Highway Transportation Officials
ACE – Army Corps of Engineers
ACI - American Concrete Institute
ADT – Average Daily Traffic
AMBIS – Assisted Management Bridge Information System
ASCE – American Society of Civil Engineers
ASTM – American Society of Testing and Materials
BHI – Bridge Health Index
BHM – Bridge Health Monitoring
BMS - Bridge Management System (more accurately called a process)
CBA – Cost Benefit Analysis
CBR – Cost Benefit Ratio
CDOT – City of Charlotte Department of Transportation
COTS – Commercial off the shelf Software
CR – Condition Rating
CRS – Commercial Remote Sensing
CRS-SI – Commercial Remote Sensing and Spatial Information
CTPS – Center for Transportation Policy Studies at UNCC
DEM – Digital Elevation Model
DLF - Dynamic Load Factor
FEA – Finite Element Analysis
FEM - Finite Element Method
FHWA – Federal Highway Administration
GenOM – Generic Object Model
GIS – Geographical Information System
GPR – Ground Penetrating Radar
GPS - Geographical Positioning Satellite
GSM – Global System for Mobile communications
HBRRP – Highway Bridge Replacement and Rehabilitation Program
HPS – High Performance Steel
HTF – Highway Trust Fund
IDE – Integrated Development Environment
ImageCat – a private sector partner in the IRSV Project
IRSV – Integrated Remote Sensing and Visualization
ISTEA – Intermodal Surface Transportation Efficiency Act
LiBE – LiDAR Bridge Evaluation
LaDAR – Laser Detection And Ranging
LiDAR – Light Distancing And Ranging
LOS – Level of Service
MR&R – Maintenance, Repair and Rehabilitation
MSVE – Microsoft Virtual Earth
NBI – National Bridge Inventory

NBIP – National Bridge Inventory Program
NBIS – National Bridge Inspection Standards
NCDOT – North Carolina Department of Transportation
NCRS-T - National Consortium for Remote Sensing in Transportation
NCSBEDC – North Carolina Small Business and Economic Development Center
NDE - Non-Destructive Evaluation
NDI – Non-Destructive Inspection
NDT – Non-Destructive Testing
NEVC – Nondestructive Evaluation Validation Center
NHS – National Highway System
NIST – National Institute for Standards and Technology
NPV – Net Present Value
NSTIFC – National Surface Transportation Infrastructure Financing Commission
OAM – Office of Asset Management, FHWA
Ontology - Synonym meaning Knowledge Modeling
PC – Prestressed Concrete
PCView – Parallel Coordinate View
PDO – Problem Domain Ontology
PMS – Pavement Management System
Point Cloud – A display of 3-D surface points in a laser scanned image
PONTIS – A “Bridgeware” software suite of programs developed through AASHTO that is used by many states as part of their Bridge Management System
RC – Reinforced Concrete
RITA – Research and Innovative Technology Administration
SAR – Synthetic Aperture Radar
SBRP – Special Bridge Replacement Program
SD/FO – Structurally Deficient and/or Functionally Obsolete
SDOF - Single-Degree-Of-Freedom
SFAP - Small Format Aerial Photography
SHM - Structural Health Monitoring
SI – Spatial Information
SIS – Software and Information Systems Department at UNC Charlotte
SMO – Semantic Matching Operation
SOA – Service Oriented Architecture
SPView – Scatter Plot View
SQL - Standard Query Language
STIP – State Transportation Improvement Program
TRB – Transportation Research Board, a part of the NAS/NAE
UNCC – University of North Carolina at Charlotte
USDOT – United States Department of Transportation
VIS – Visualization
VisCenter – Charlotte Visualization Center at UNCC

Appendix C. Data used in Knowledge Modeling

Data	Description
Bridge Property	Type of Service
Material Type	Type of material used for the bridge
Superstructure Type	Describes the concrete type and the span classification of the bridge
Span Classification	Classifies the simple, continuous and cantilever type of bridge
Bridges Images	Describes the LIDAR, satellite and digital images of the bridges
Bridge Data Sources	Describes the rating, proposed improvement, inspection report, geometric data and mitigation of the bridges
Current Inspection and Rating	Describes the defect of the bridges analyzed by the bridge inspectors
LiBA	Describes the defect of the bridges analyzed by Civil Engineers with the help of LIDAR data
AMPIS – bridge deck analysis	Describes the defect of the bridges analyzed by ImageCAT with the help of AMPIS data
Bridge operative process guidance	Describes the NCDOT process

Appendix D: User Guide to IRSV Software Application

IRSV Prototype

The user interface and the underlying logic are modeled primarily on the inputs provided from NCDOT and CDOT. By using a tabbed interface to depict the workflow of these DOT processes, the prototype tries to associate the system's look and feel to the flow of activities followed by the DOT personnel in inspecting and evaluating bridges. The primary focus of IRSV prototype user interface is to combine bridge inspection data and domain knowledge based on the knowledge representation and a goal-driven modeling technique. This prototype will also employ a primitive user interface which helps in exploring and using the functionalities of the services created.

Web Service

A Service-Oriented Architecture (SOA) framework can easily integrate tools with different software solutions in a scalable and adaptable manner. Through services, called the knowledge service, the SOA paradigm with problem domain ontology allows IRSV system to model and expose complex heterogeneous bridge data, such as, sensor and inspection data contained in inspection reports, images, etc. Moreover, the knowledge service not only provides an easy access to relevant data, but also helps bridge managers to understand the nature and correlations among those data.

Database

In coordination with the Civil Engineering Department team, our team (the Software and Information Systems) has collected and organized domain information pertaining to the bridge inspection process. Sources of information included Charlotte Dept. of Transportation (CDOT) and North Carolina Dept. of Transportation (NCDOT) representatives. To date, the knowledge modeling team has managed to collect and classify data pertaining to the following aspects of the bridge inspection process.

- Inspection_Report_CDOT
- Inspection_Report_NCDOT
- Images

The data collected from the domain experts were organized into tables. The names and structure of these tables were in accordance with the data classification mentioned above. The Inspection_Report_CDOT table contains the three-year record for CDOT of two-year cycle, i.e., 2000, 2004 and 2006, and the Inspection_Report_NCDOT table contains around 300 bridges for NCDOT. The image table will contains LIDAR images, satellite images, air-borne images, etc. These data are imported in SQL Server.

GenOM

GenOM (Generic Object Model) is a knowledge acquisition tool that aids the designing and implementation of any "intelligent" software application by using object-oriented technologies. It addresses the following three characteristics:

- Object modeling in its representation
- Usage of objects in its application model
- Ability to aggregate evidence that supports the analysis of object's behaviors (through the associated properties and relationships between objects).

The harmonization of these characteristics often determines the level of intelligence of the applications. When a software computing paradigm converges toward domain-independent interdisciplinary research, the objects (or models) used in each application model should be interoperable and reusable. GenOM is such an interoperable and reusable object computing Model. Also, GenOM provides ways for mapping, merging and integrating domain-specific objects and thus serves as a knowledge base for building object-oriented software applications.

Features

IRSV Prototype:

1. Create data profile for the bridges
2. Correlate profiles to existing data sources
3. Use the correlation to locate relevant images and other sensor information/inspection reports
4. Analyze the available data sources with tools/processes provided by the AMPIS system.
5. Based on the evaluation criteria generated by AMPIS module, defects can be tagged with the help of knowledge structure (ontology) and recommend the bridge with the mitigation strategies.
6. Logical reasoning can be generated based on what – if conditions through the conceptual space (knowledge structure / Ontology)
7. Generate a report that includes the summary of analysis, mitigation strategies with the help of metrics and measures defined in the conceptual space.
8. Store this process as a customized process, which can be repeated in the future

Web Service:

1. Provides the knowledge service to different modules
2. Provides the rules inferred by the inference engine to different modules
3. Provides concept, properties, features and instances to other modules
4. Support the interoperability, scalability and adaptability to facilitate heterogeneous data requirements, operational requirements and the overlapping functionalities
5. Compose meaningful set of services that support other system components needs

6. Knowledge services can mediate between the various system components and the process services.

<u>*GenOM:*</u>

The Inference Engine of GenOM provides Rule based Inference mechanism to enable the user to query the knowledge base by creating rules. The Inference Engine component consists of three parts:

1. Rule Modeler: This panel is used for creating rules in the IF THEN format. The variables are the instances of Object or Feature.
2. Rule Library: This panel displays all the rules defined on the knowledge base using the Rule Modeler.
3. Result Panel: This panel displays the result of executing a particular rule from the Rule Library.

Functionalities

<u>*IRSV Prototype:*</u>

1. The prototype consists of several tabs. In the first tab *'Data Profile'*, the user enters the bridge number (590140) and retrieves the historical data of 3-year inspection cycle (2006, 2004 and 2000), source information and images (digital, LiDAR and satellite) of the bridge 590140.
2. The user then clicks on the *'Analyze Bridge Data'* tab where the user views analysis performed by the AMPIS module and LBDA Module on the bridge number 590140.
3. The user clicks on the next tab *'TIP Chart'* where the user can view the TIP table of the years listed in combo box and recommend a proposal for the TIP.
4. In the next tab, *'Inference Engine'* the user can click on the rules and see the list of bridges that are affected due to the factors mentioned in the established rules.

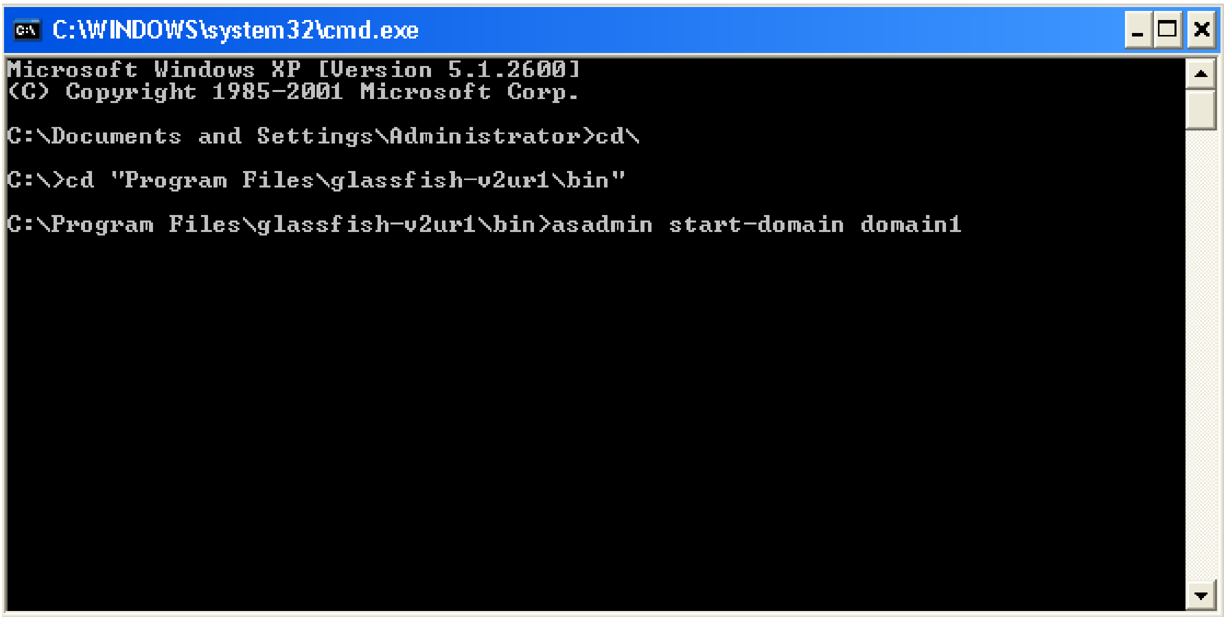
<u>*Web Service:*</u>

By hosting a web service, other modules can access the services and share the common knowledge and common understanding. With the help of service – oriented architecture concept, one of the modules of IRSV system, visualization module can successfully invoke the list of rules implemented based on knowledge structure and execute those rules in the form of visual analytics form.

Steps for Hosting a Web Service (as shown in figure below)

1. Go to Start menu
2. Click on Run
3. A dialog box will appear, then type 'cmd' in that dialog box and click Ok
4. Then type 'cd\' and press enter
5. Then type 'cd "Program Files\glassfish-v2ur1\bin"'and press enter

6. Then write 'asadmin start-domain domain1' and press enter
7. The Server will start running



GenOM:

1. The user switches to the GenOM application and opens the bridge management ontology file. On the left hand side of the *'Object'* tab, all the objects / concepts of the bridge inspection process are listed and on the right hand side the properties of each object are listed.
2. The user then clicks on the 'Properties' tab where the user can view the properties of all the objects.
3. The user clicks on the *'Features'* tab where the relationship between the objects / properties are listed.
4. The user clicks on the *'Instances (Object)'* tab where all the instances of a particular object are listed.
5. The user clicks on the *'Inference Engine'* tab where the user selects each rule from the 'Rule Library' and executes it. On the right hand side, the user can view the list of the bridges that fall under the Rule (1 to 7) conditions.

Benefits

IRSV Prototype:

1. Exploring IRSV Prototype by using a tabbed interface to depict the workflow of the inspection process by specifying a particular bridge number (say 590140).
2. Analyzing the condition of the bridge with the help of historical report and maintain the status of the bridge since the initial inspection cycle. Providing with some relevant source information related to the bridge including LiDAR, satellite, under deck images which describes the defects and condition of the bridge.
3. Analyzing the bridge data by invoking the LiDAR bridge defect analysis (LBDA) and AMPIS module. Inspector can view the evaluation criteria performed by AMPIS and LBDA tools and based on the ratings, measures and metrics the defects can be associated to the bridge with the help of knowledge structure.
4. Based on the defects and type of service of the bridge in the knowledge structure (Ontology), cost estimations can be manipulated.
5. With the help of defect classifications, material type of bridge, structure type, span classifications of the bridge in the conceptual space (knowledge structure) mitigation strategies will be generated.
6. Set of bridges that have similar patterns with respect to defect classification and structure type of the bridge can be concluded through inference engine. For example, Bridge no 590140 indicates bad condition based on evaluation criteria, sufficiency rating, some observation and images. Similarly, list of other bridges can be displayed based on these conditions through knowledge structure. Therefore with the help of inference engine, logical conclusions can be generated based on what-if conditions.
7. Generating the TIP Chart and allocate the funding to the bridge that needs to be rehabilitated/ repaired.
8. The summary report will be generated based on the analysis performed by LBDA tool, AMPIS module and inference engine.

Appendix E: Inference result for CDOT and NCDOT bridges

Following are the scenarios of CDOT and NCDOT bridges which is continued from section.

CDOT Bridges

Scenario for bridge number 590376.

According to the inspection record of year 2006, this bridge is in fair condition. This was built in 1960 and carries 3,300 vehicles per day. It is classified as functionally obsolete bridge but not a scour critical bridge. Waterway adequacy rating is 7 and it is posted for load.

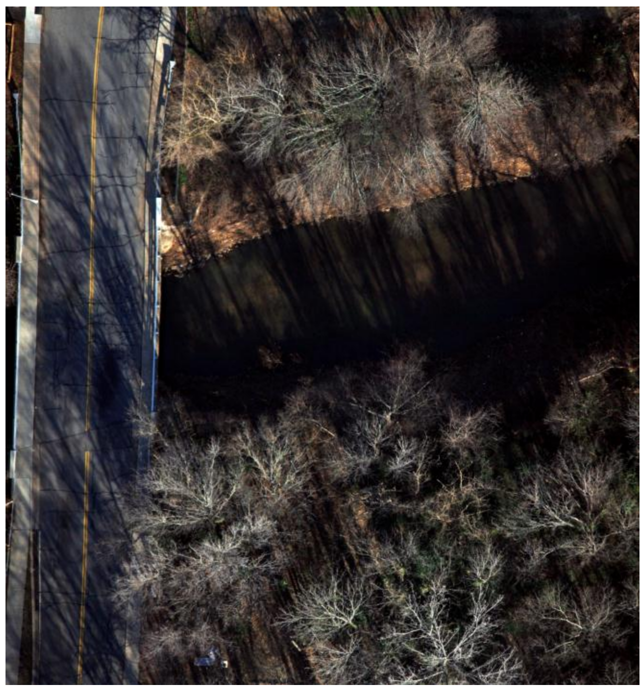


Figure 2.41: Digital image of bridge 590376

Inference:

Scour Criticality == “FALSE” AND
Water Inadequacy == “FALSE” AND
Structural Deficiency == “FALSE” AND
Functional Obsolescence == “TRUE” AND
Present Condition == “FAIR” AND
Load Posting == “TRUE” AND
ADT < 20000

Condition 4 (Obsolete) and Condition 6 (Load posting) are true for the bridge. By applying the rules one by one, the system infers that this bridge needs replacement per Rule 5.

<u>Scenario for bridge number 590379</u>.

According to the inspection record of year 2006, this bridge is in fair condition. This was built in 1965 and carries 500 vehicles per day. It is classified as structural deficient bridge but not a scour critical bridge. Waterway adequacy rating is 5 and it is posted for load.

Figure 2.42: Digital image of bridge 590379

Inference:

Scour Criticality == “FALSE” AND
Water Inadequacy == “FALSE” AND
Structural Deficiency == “TRUE” AND
Functional Obsolescence == “FALSE” AND
Present Condition == “FAIR” AND
Load Posting == “TRUE” AND
ADT < 20000

Based on this inference, condition 3 (Structural deficiency) and condition 6 (Load Posting) apply to this bridge. By applying the rules one by one, the system infers that this bridge <u>needs replacement</u> per Rule 2.

<u>Scenario for bridge number 590700</u>.

According to the inspection record of year 2006, this bridge is in poor condition. This is a new bridge built in 1996 and carries a heavy traffic of 30,600 vehicles per day. It is neither structural deficient nor functionally obsolete. It is not a scour critical bridge. There are no issues of Waterway inadequacy and it is open and has no restrictions.

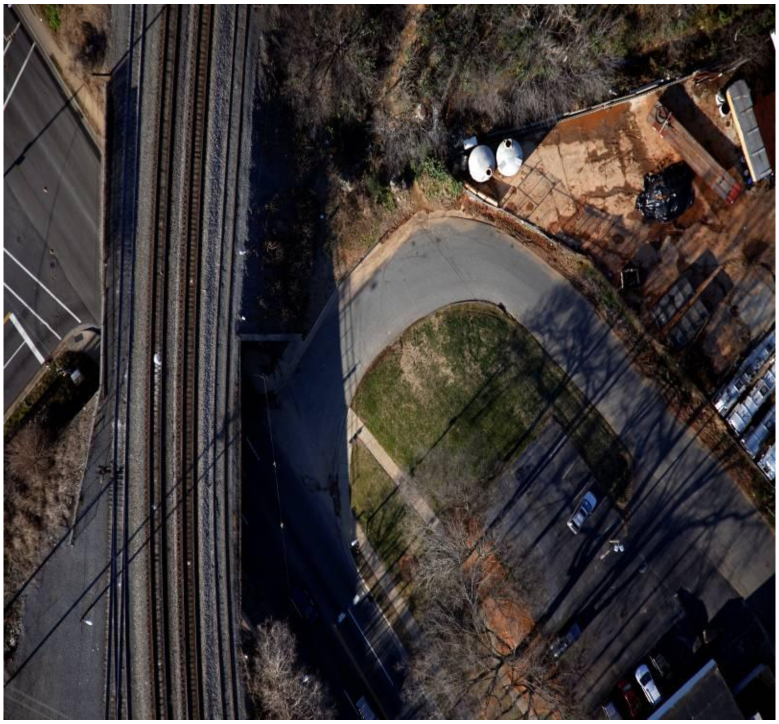


Figure 2.43: Digital image of bridge 590700

Inference:
Scour Criticality == "FALSE" AND
Water Inadequacy == "FALSE" AND
Structural Deficiency == "FALSE" AND
Functional Obsolescence == "FALSE" AND
Present Condition **== "POOR"** AND
Load Posting == "FALSE" AND
ADT > 20000

Condition 5 (poor present condition) and condition 7 (High ADT) are true for the bridge. By applying the rules one by one, the system infers that this bridge needs no replacement per Rule 8.

Scenario for bridge number 590702.

According to the inspection record of year 2006, this bridge is in good condition. This is a relatively new bridge built in 1996 and carries 4,800 vehicles per day. It is neither structural deficient nor functionally obsolete. It is not a scour critical bridge. There are no issues of waterway inadequacy and it is open and has no restrictions.

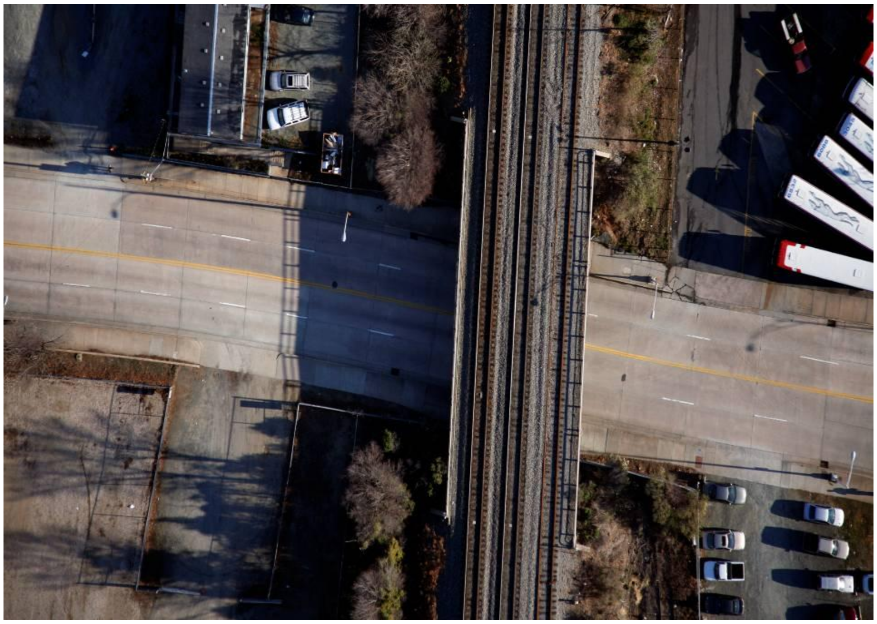


Figure 2.44: Digital image of bridge 590702

Inference:

Scour Criticality == “FALSE” AND
Water Inadequacy == “FALSE” AND
Structural Deficiency == “FALSE” AND
Functional Obsolescence == “FALSE” AND
Present Condition == “GOOD” AND
Load Posting == “FALSE” AND
ADT < 20000

By evaluating the bridge for each of the conditions, system infers that none of the conditions hold true for this bridge. So this bridge does not require immediate attention.

<u>Scenario for bridge number 590704</u>.

According to the inspection record of year 2006, this bridge is in fair condition. This is a relatively new bridge built in 1996 and carries 5,100 vehicles per day. It is neither structural deficient nor functionally obsolete. It is not a scour critical bridge. There are no issues of waterway inadequacy and it is open and has no restrictions.

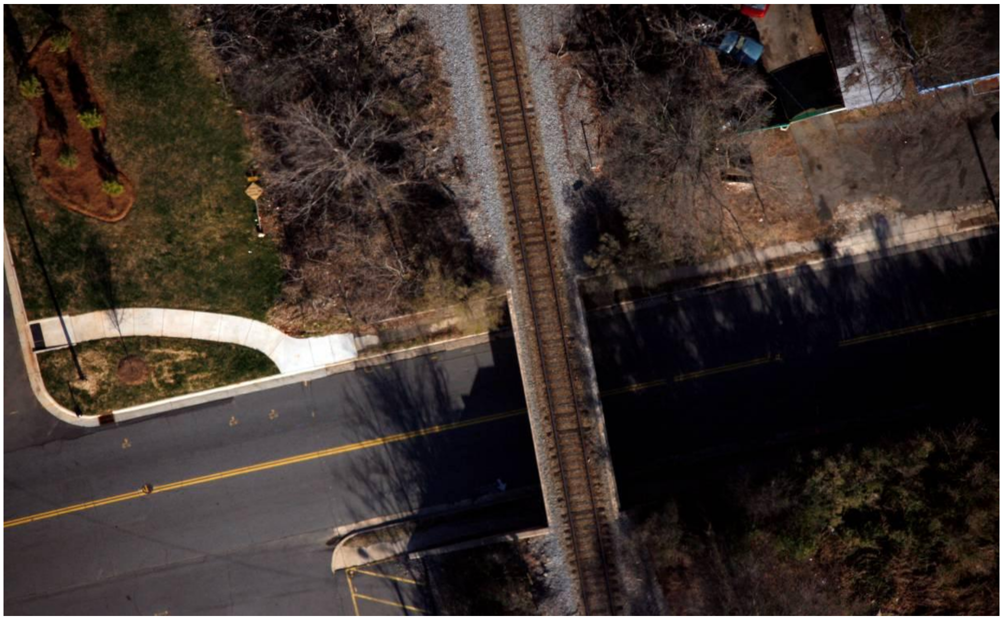


Figure 2.45: Digital image of bridge 590704

Inference:

Scour Criticality == "FALSE" AND
Water Inadequacy == "FALSE" AND
Structural Deficiency == "FALSE" AND
Functional Obsolescence == "FALSE" AND
Present Condition == "FAIR" AND
Load Posting == "FALSE" AND
ADT < 20000

By evaluating the bridge for each of the conditions, system infers that none of the conditions hold true for this bridge. So this bridge does not require immediate attention.

Scenario for bridge number 590706.

According to the inspection record of year 2006, this bridge is in poor condition. This is a relatively new bridge built in 1996 and carries 15,700 vehicles per day. It is neither structural deficient nor functionally obsolete. It is not a scour critical bridge. There are no issues of waterway inadequacy and it is open and has no restrictions.

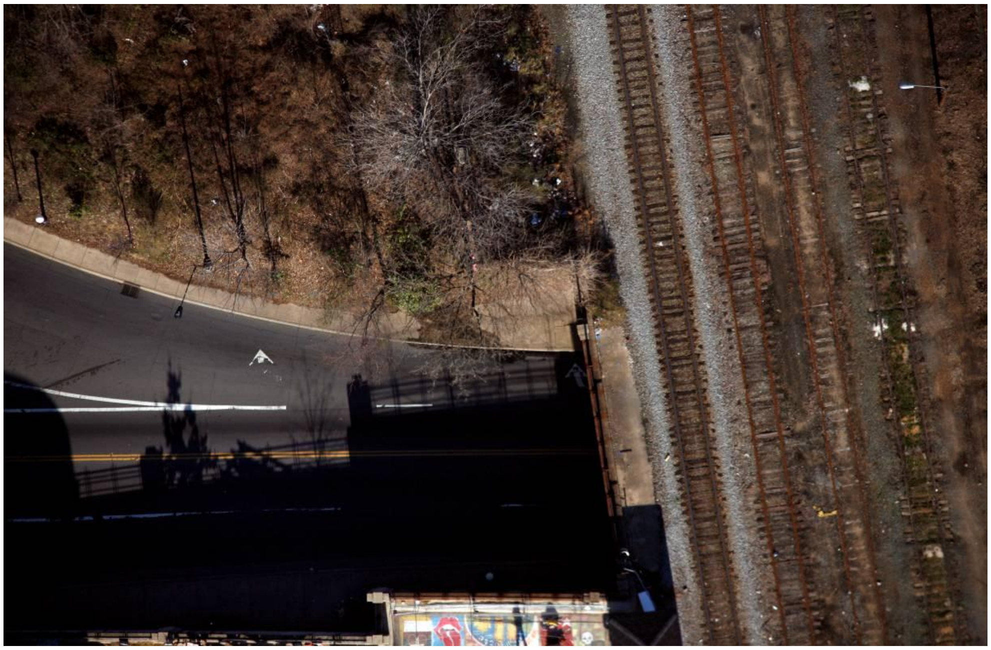


Figure 2.46: Digital image of bridge 590706

Inference:

Scour Criticality == "FALSE" AND
Water Inadequacy == "FALSE" AND
Structural Deficiency == "FALSE" AND
Functional Obsolescence == "FALSE" AND
Present Condition == "POOR" AND
Load Posting == "FALSE" AND
ADT < 20000

Condition 5 (poor present condition) is true for the bridge. By applying the rules one by one, the system infers that none of the rules apply to this bridge.

NCDOT Bridges

<u>Scenario for bridge number 590084</u>.

According to the inspection record of year 2006, this bridge is in good condition. This is a relatively new bridge built in 2004 and carries 9,500 vehicles per day. It is neither structural deficient nor functionally obsolete. Scour criticality rating is 8. Waterway adequacy rating is 8. It is open and has no restrictions.

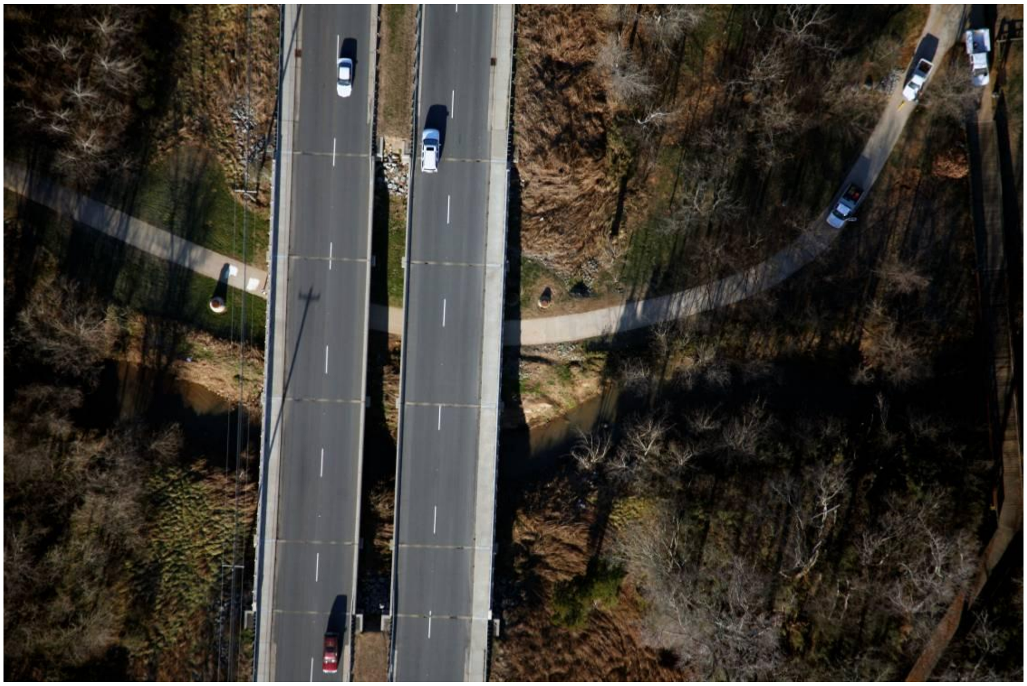


Figure 2.47: Digital image of bridge 590084

Inference:
Following are the results.
Scour Criticality == “FALSE” AND
Water Inadequacy == “FALSE” AND
Structural Deficiency == “FALSE” AND
Functional Obsolescence == “FALSE” AND
Present Condition == “GOOD” AND
Load Posting == “FALSE” AND
ADT < 20000

By applying the rules one by one, the system infers that none of the rules apply to this bridge. So this bridge does not require immediate attention.

<u>Scenario for bridge number 590140</u>.

According to the inspection record of year 2006, this bridge is in fair condition. This is an old bridge built in 1951 and carries 21,000 vehicles per day. It is neither structural deficient nor functionally obsolete. Scour criticality rating is 8. Waterway adequacy rating is 6. It is open and has no restrictions.

Figure 2.48: Digital image of bridge 590140

Inference:

Following are the results
Scour Criticality == "FALSE" AND
Water Inadequacy == "FALSE" AND
Structural Deficiency == "FALSE" AND
Functional Obsolescence == "FALSE" AND
Present Condition == "FAIR" AND
Load Posting == "FALSE" AND
ADT > 20000
By applying the rules one by one, the system infers that none of the rules apply to this bridge. So this bridge does not require immediate attention.

<u>Scenario for bridge number 590147</u>.

According to the inspection record of year 2006, this bridge is in fair condition. This is an old bridge built in 1938 and carries 21,000 vehicles per day. It is not structurally deficient. It is not a scour critical bridge. Waterway adequacy rating is 6 and it is open and has no restrictions. (Refer Figure 2.30: Image of bridge #590140)

Inference:

Scour Criticality == "FALSE" AND
Water Inadequacy == "FALSE" AND
Structural Deficiency == "FALSE" AND
Functional Obsolescence == "FALSE" AND
Present Condition == "FAIR" AND
Load Posting == "FALSE" AND
ADT > 20000

By applying the rules one by one, the system infers that none of the rules apply to this bridge. So this bridge does not require immediate attention.

Scenario for bridge number 590179.

According to the inspection record of year 2006, this bridge is in fair condition. This is an old bridge built in 1937 and carries 22,000 vehicles per day. It is not structurally deficient. It is not a scour critical bridge. It is open and has no restrictions.

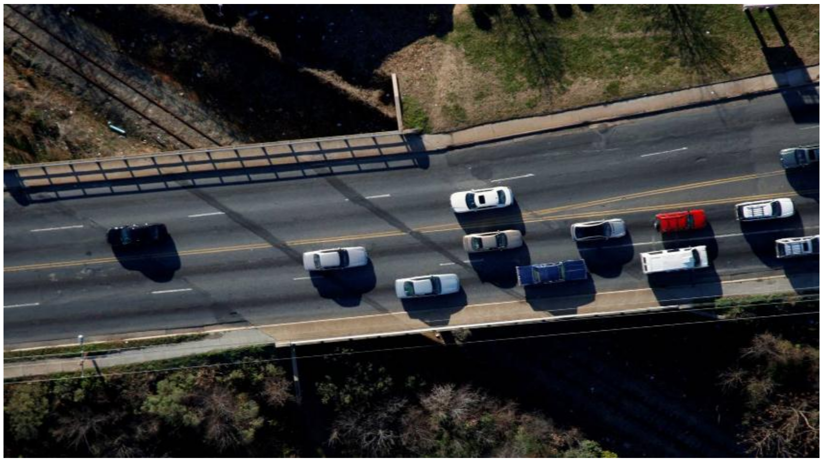


Figure 2.49: Digital image of bridge 590179

Inference:

Scour Criticality == "FALSE" AND
Water Inadequacy == "FALSE" AND
Structural Deficiency == "FALSE" AND
Functional Obsolescence == "FALSE" AND
Present Condition == "FAIR" AND
Load Posting == "FALSE" AND
ADT > 20000

By applying the rules one by one, the system infers that none of the rules apply to this bridge. So this bridge does not require immediate attention.

<u>Scenario for bridge number 590239</u>.

According to the inspection record of year 2006, this bridge is in fair condition. This was built in 1966 and carries 19,000 vehicles per day. It is not structural deficient. It is not a scour critical bridge. It is waterway adequate. It is open and has no restrictions.

Figure 2.50: Digital image of bridge 590239

Inference:

Scour Criticality == “FALSE” AND
Water Inadequacy == “FALSE” AND
Structural Deficiency == “FALSE” AND
Functional Obsolescence == “FALSE” AND
Present Condition == “FAIR” AND
Load Posting == “FALSE” AND
ADT < 20000

By applying the rules one by one, the system infers that none of the rules apply to this bridge. So this bridge does not require immediate attention.

<u>Scenario for bridge number 590298</u>.

According to the inspection record of year 2006, this bridge is in fair condition. This was built in 1967 and has a very high ADT of 435,000 vehicles per day. It is not structural deficient. Scour criticality rating is 8. Waterway adequacy rating is 8 and it is open and has no restrictions.

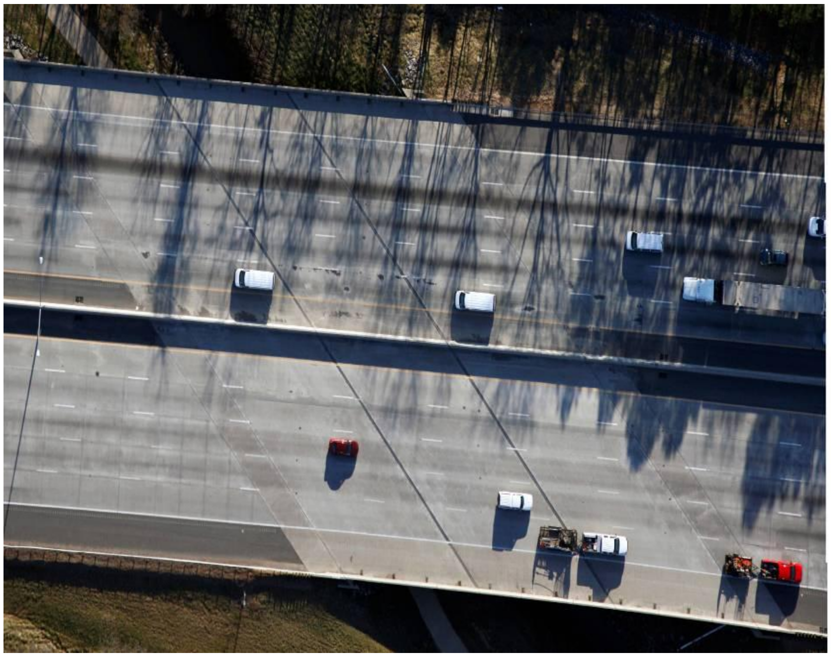


Figure 2.51: Digital image of bridge 590298

Inference:

Scour Criticality == "FALSE" AND
Water Inadequacy == "FALSE" AND
Structural Deficiency == "FALSE" AND
Functional Obsolescence == "FALSE" AND
Present Condition == "FAIR" AND
Load Posting == "FALSE" AND
ADT > 20000

By applying the rules one by one, the system infers that none of the rules apply to this bridge. So this bridge does not require immediate attention.

Scenario for bridge number 590511.

According to the inspection record of year 2006, this bridge is in good condition. This was built in 1987 and carries 26,000 vehicles per day. It is not structurally deficient. It is not a scour critical bridge. It is open and has no restrictions.

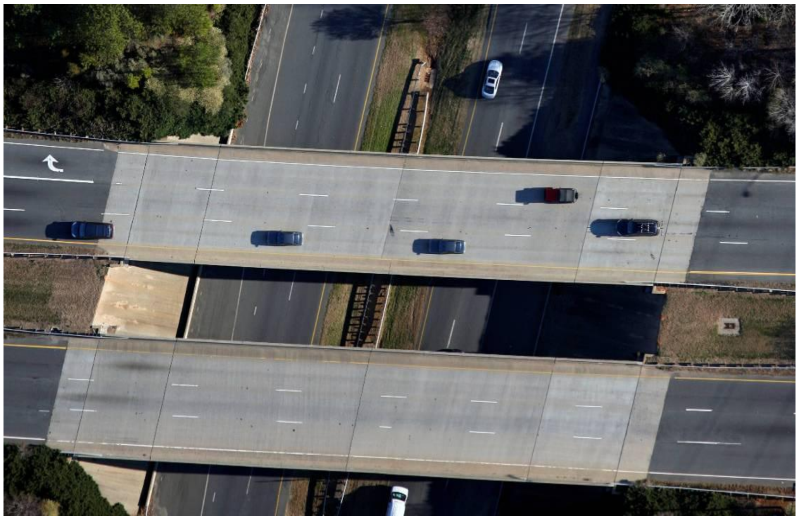


Figure 2.52: Digital image of bridge 590511

Inference:

Scour Criticality == “FALSE” AND
Water Inadequacy == “FALSE” AND
Structural Deficiency == “FALSE” AND
Functional Obsolescence == “FALSE” AND
Present Condition == “GOOD” AND
Load Posting == “FALSE” AND
ADT > 20000

By applying the rules one by one, the system infers that none of the rules apply to this bridge. So this bridge does not require immediate attention.

<u>Scenario for bridge number 590512</u>.

According to the inspection record of year 2006, this bridge is in good condition. This was built in 1987 and carries 26,000 vehicles per day. It is not structurally deficient. It is not a scour critical bridge. It is open and has no restrictions. (Refer to Figure 2.34: Image of bridge #590511)

Inference:

Scour Criticality == “FALSE” AND
Water Inadequacy == “FALSE” AND
Structural Deficiency == “FALSE” AND
Functional Obsolescence == “FALSE” AND
Present Condition == “GOOD” AND
Load Posting == “FALSE” AND
ADT > 20000

By applying the rules one by one, the system infers that none of the rules apply to this bridge. So this bridge does not require immediate attention.

Scenario for bridge number 590038.

According to the inspection record of year 2006, this bridge is in fair condition. This was built in 1945 and carries 13,000 vehicles per day. It is structurally deficient. It is not a scour critical bridge. Waterway adequacy rating is 7 and it is posted for load.

Inference:

Scour Criticality == “FALSE” AND
Water Inadequacy == “FALSE” AND
Structural Deficiency == “TRUE” AND
Functional Obsolescence == “FALSE” AND
Present Condition == “FAIR” AND
Load Posting == “TRUE” AND
ADT < 20000

By applying the rules one by one, the system infers that based on rule 2 this bridge needs replacement.

<u>Scenario for bridge number 590049</u>.

According to the inspection record of year 2006, this bridge is in fair condition. This was built in 1926 and carries 14,000 vehicles per day. It is not structurally deficient. Scour criticality rating is 8. Waterway adequacy is 8 and it is posted for load.

Figure 2.53: Digital image of bridge 590049

Inference:

Scour Criticality == “FALSE” AND
Water Inadequacy == “FALSE” AND
Structural Deficiency == “FALSE” AND
Functional Obsolescence == “FALSE” AND
Present Condition == “FAIR” AND
Load Posting == “TRUE” AND
ADT < 20000

By applying the rules one by one, the system infers that none of the rules apply to this bridge. So this bridge does not require immediate attention.

<u>Scenario for bridge number 590059</u>.

According to the inspection record of year 2006, this bridge is in fair condition. This was built in 1976 and carries a very low traffic of 4,300 vehicles per day. It is not structurally deficient. It is not a scour critical bridge. Waterway adequacy is 8 and it is posted for load.

Inference:

Scour Criticality == “FALSE” AND
Water Inadequacy == “FALSE” AND
Structural Deficiency == “FALSE” AND
Functional Obsolescence == “FALSE” AND
Present Condition == “FAIR” AND
Load Posting == “TRUE” AND
ADT < 20000

By applying the rules one by one, the system infers that none of the rules apply to this bridge. So this bridge does not require immediate attention.

<u>Scenario for bridge number 590108</u>.

According to the inspection record of year 2006, this bridge is in good condition. This is a relatively new bridge built in 2005 and carries 18,000 vehicles per day. It is not structurally deficient. It is not a scour critical bridge. It is open and has no restrictions.

Figure 2.54: Digital image of bridge 590108

Inference:

Scour Criticality == "FALSE" AND
Water Inadequacy == "FALSE" AND
Structural Deficiency == "FALSE" AND
Functional Obsolescence == "FALSE" AND
Present Condition == "GOOD" AND
Load Posting == "FALSE" AND
ADT < 20000

By applying the rules one by one, the system infers that none of the rules apply to this bridge. So this bridge does not require immediate attention.

Scenario for bridge number 590161.

According to the inspection record of year 2006, this bridge is in fair condition. This was built in 1961 and carries 500 vehicles per day. It is functionally obsolete. Scour criticality is 8. Waterway adequacy is 8 and it is open and has no restrictions.

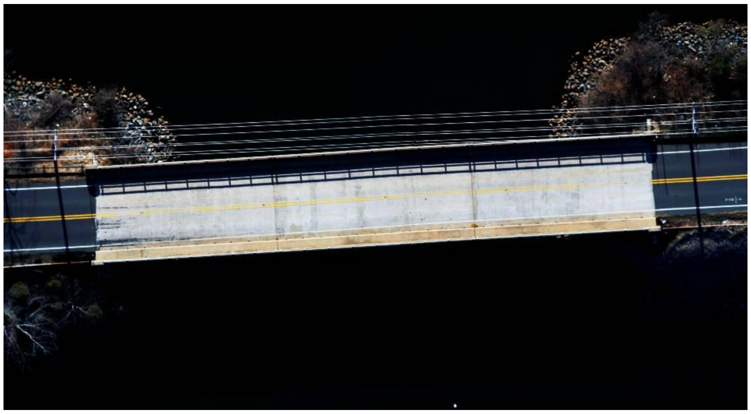


Figure 2.55: Digital image of bridge 590161

Inference:

Scour Criticality == "FALSE" AND
Water Inadequacy == "FALSE" AND
Structural Deficiency == "FALSE" AND
Functional Obsolescence == "TRUE" AND
Present Condition == "FAIR" AND
Load Posting == "FALSE" AND
ADT < 20000

By applying the rules one by one, the system infers that none of the rules apply to this bridge. So this bridge does not require immediate attention.

<u>Scenario for bridge number 590165</u>.

According to the inspection record of year 2006, this bridge is in poor condition. This was built in 1975 and carries 6,800 vehicles per day. It is structurally deficient. Scour criticality rating is 7. Waterway adequacy rating is 8 and it is posted for load.

Figure 2.56: Digital image of bridge 590165

Inference:

Scour Criticality == "FALSE" AND
Water Inadequacy == "FALSE" AND
Structural Deficiency == "TRUE" AND
Functional Obsolescence == "FALSE" AND
Present Condition == "POOR" AND
Load Posting == "TRUE" AND
ADT < 20000

This bridge exhibits condition 3(Structural deficiency), condition 5(Poor present condition) and condition 6 (Load posting). By applying the rules one by one, the system infers that rule 2 applies to this bridge and this bridge needs replacement.

<u>Scenario for bridge number 590176</u>.

According to the inspection record of year 2006, this bridge is in fair condition. This was built in 1955 and carries 16,000 vehicles per day. It is structurally deficient. It is not a scour critical bridge. It is open and has no restrictions.

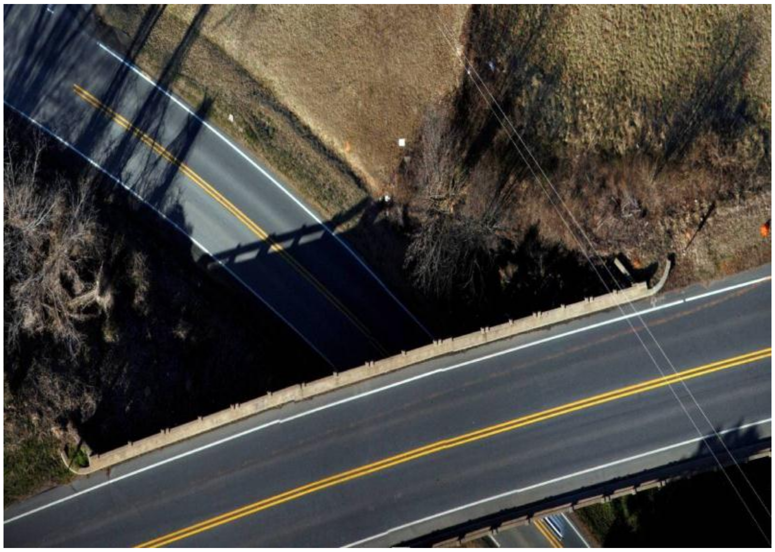


Figure 2.57: Digital image of bridge 590176

Inference:

Scour Criticality == "FALSE" AND
Water Inadequacy == "FALSE" AND
Structural Deficiency == "TRUE" AND
Functional Obsolescence == "FALSE" AND
Present Condition == "FAIR" AND
Load Posting == "FALSE" AND
ADT < 20000

By applying the rules one by one, the system infers that the bridge <u>needs replacement</u> based on rule 2.

Scenario for bridge number 590177.

According to the inspection record of year 2006, this bridge is in fair condition. This was built in 1970 and carries 650 vehicles per day. It is structurally deficient. It is not a scour critical bridge. Waterway adequacy rating is 7 and it is posted for load.

Figure 2.58: Digital image of bridge 590177

Inference:

Scour Criticality == “FALSE” AND
Water Inadequacy == “FALSE” AND
Structural Deficiency **==** **“**TRUE**”** AND
Functional Obsolescence == “FALSE” AND
Present Condition == “FAIR” AND
Load Posting == “TRUE” AND
ADT < 20000

By applying the rules one by one, the system infers that the bridge needs replacement based on rule 2.