CAMBRIDGE
SYSTEMATICS

Pontis 5.0 Bridge Management System

Functional Requirements Specification v1.0

final

report

prepared for

American Association of State Highway and Transportation Officials (AASHTO)

prepared by

Cambridge Systematics, Inc.

February 2005

www.camsys.com

Table of Contents

1.0 Introduction 1-1

2.0 Project Drivers 2-1
2.1 Purpose of the Product 2-1
2.2 Stakeholders 2-3
2.3 Users of the Product 2-4
2.4 Project Constraints 2-5
2.5 Naming Conventions and Definitions 2-7
2.6 Relevant Facts and Assumptions 2-7

3.0 Use Cases 3-1
3.1 Browse Bridge & Project Data 3-1
3.2 Bridge Inventory & Inspection 3-3
3.3 Preservation Model Development 3-6
3.4 Program Simulation 3-8
3.5 Project & Program Development 3-10
3.6 Data Management 3-12
3.7 System Administration 3-14

4.0 Functional Requirements 4-1
4.1 Browse Data 4-1
4.2 Bridge Inventory & Inspection 4-3
4.3 Preservation Model development 4-6
4.4 Program Simulation 4-7
4.5 Project & Program Development 4-8
4.6 Data Management 4-9
4.7 System Administration 4-11
4.8 Data Requirements 4-14

5.0 Non-Functional Requirements 5-1
5.1 Look and Feel Requirements 5-1
5.2 Usability Requirements 5-1
5.3 Performance Requirements 5-2
5.4 Operational Requirements 5-6
5.5 Installation and Deployment 5-7
5.6 Maintainability and Portability Requirements 5-7

5.7 Security Requirements 5-8
5.8 Cultural and Political Requirements 5-8
5.9 Legal Requirements 5-9

6.0 Implementation Plan 6-1
6.1 Overview 6-1
6.2 Resource Requirements Analysis Approach 6-2
6.3 Alternative 1 – Dedicated Design and Release Approach 6-3
6.4 Alternative 2 – Phased Design and Release Approach 6-5
6.5 Alternative 3 – Dedicated Design and Phased Approach 6-6
6.6 Recommended Alternative 6-7
6.7 Risk Analysis 6-9

7.0 Project Issues 7-1
7.1 Open Issues 7-1
7.2 Off-The-Shelf Solutions 7-1
7.3 New Problems 7-1
7.4 Waiting Room 7-1

Appendix A – COSMIC-FFP Overview A-1

Appendix B – Project Schedule B-1
B.1 Dedicated Design and Release B-1
B.2 Phased Design and Release B-2
B.3 Dedicated Design and Phased Release B-4

Document History

Date	Author	Version/Status	Description of Changes
2/28/2005	CS (ARM, SJH, WER, AAP)	Version 1.0	Updated use case diagrams, incorporated comments received from the BRIDGEWare Task Force, TAG and Virtis/Opis development team.
2/10/2005	CS (ARM, SJH, WER, AAP)	DRAFT Version 0, rev. 6	Responded to comments from SD, FL, CA, MT; revised implementation plan, particularly option 2
1/13/2005	CS (ARM, SJH, WER, AAP)	DRAFT Version 0, rev. 5	Added comments from C. Laughlin, made minor spelling changes, revised implementation plan.
1/3/2005	CS (ARM, SJH, WER, AAP)	DRAFT Version 0, rev. 4	Added implementation plan chapter
12/13/2004	CS (ARM, SJH) with TAG comments	DRAFT Version 0, rev. 3	Incorporates comments on functional requirements from Atlanta meeting
12/6/2004	CS (ARM, SJH, AAP, WER) with TAG comments	DRAFT Version 0, rev. 2	Initial TAG comments from P. Jensen and C. Laughlin incorporated, added use case descriptions and UML diagrams
11/29/2004	CS (ARM, SJH, AAP, WER)	DRAFT Version 0, rev. 1	Initial draft of Pontis 5.0 functional requirements specification

1.0 Introduction

This report documents the functional requirements specification (FRS) for the next generation of the Pontis Bridge Management System (BMS), referred to as Pontis 5.0 throughout. These include the functional requirements, which describe the operations that users must perform using Pontis 5.0, the non-functional requirements, which describe the manner in which the software will be designed, developed, and deployed, and the plan for implementation of Pontis 5.0.

Throughout this document, the terms "shall" and "will" are used to identify specific functions and capabilities that Pontis 5.0 must provide (e.g., Pontis inspection interface shall provide the ability to perform data validation on National Bridge Inventory items). The terms "may" and "might" are used to indicate functions and capabilities that may or may not be included in the final product (e.g., Pontis may provide a direct link to GIS software for map-based querying). The status of these items will be resolved through future detailed software design activity or through progress by other vendors in the development and deployment of Pontis add-in applications.

Note that this document is not intended to describe the architecture or data model of Pontis 5.0, nor is it intended to describe the specific design or implementation plan for any functional requirement. This information will appear in later Pontis 5.0 system design documents. Preliminary work has been performed to outline the technical architecture for Pontis 5.0, and a Technical Architecture Description (TAD) was developed in 2003.[1] The TAD presents a future vision for the project, emphasizing a web application, an advanced database architecture, maximum reuse of user reports and forms, and a flexible, extensible user interface.

This document, once approved in final form, indicates agreement between the AASHTO Pontis Task Force, on behalf of the Pontis licensing agencies, and Cambridge Systematics, on the specific functions and capabilities that Pontis 5.0 must provide, providing a clear and specific scope of services for the software design and development effort. Any subsequent changes to the functional requirements must be carefully reviewed and considered, because they may impact the project schedule and/or or expand the scope of work beyond that which is covered by the design and development contract between AASHTO and the software developer.

[1] Cambridge Systematics, *Pontis Technical Architecture – A Future Vision*, 2003 (http://aashtoware.camsys.com/pontis/5_0/Design/Technical_Architecture_Description/Pontis_Future_Vision_04182003.pdf).

This document follows the structure and outline of the Volere Requirements Specification Template.[2] This approach was identified in previous discussions and documents.[3] Some sections of this template do not apply to Pontis 5.0, and have therefore been marked as such in the text of this report, or omitted in their entirety.

The TAD provides a general architecture for the future version of Pontis. It does not elaborate requirements for the new system in detail, although it does identify key stakeholders and stakeholder concerns to be considered in the future Pontis version. A specific requirements gathering effort is necessary before the design and implementation of Pontis 5.0 may commence.

To that end, the BRIDGEWare Task Force has designated the contractor, Cambridge Systematics, to undertake a requirements gathering process with guidance and substantive input from a set of bridge management and information systems experts who will serve as a Technical Advisory Group (TAG). Cambridge Systematics and the TAG work within the framework of the Volere Requirements Process Model as the formal requirements development methodology.

In the Volere methodology, there are several main phases:

- Project Blastoff - Define the project scope, context, and participants, and initiate the requirements process
- Trawling For Knowledge - Collect information needed to write the requirements document
- Reporting - Compile the requirements and write the requirements document
- Quality Gateway – (continuously) Review the requirements as they are drafted for completeness, relevance, and viability
- Prototyping and Scenario Modeling – (eventually) Develop low-fidelity and/or high-fidelity (HiFi) prototypes to better develop the requirements
- Perform Requirements Gathering Effort Post-mortem (Retrospective Review) - review outcome of the requirements process following completion of the requirements document
- Take Stock Of The Requirements – Determine if the requirements are complete and internally consistent

These phases culminate in this requirements specification which can move forward to the formal software design phase.

[2] http://www.volere.co.uk

[3] Cambridge Systematics, *Pontis 5.0 Blastoff Report*, October 2004.

2.0 Project Drivers

2.1 Purpose of the Product

Pontis 5.0 will provide licensing agencies with an up-to-date tool for bridge management, including data management, condition assessment, model development, needs analysis, reporting, and interaction with other agency systems. Pontis 5.0 will be the next generation of the Pontis product, currently deployed in more than 45 agencies in the U.S. and abroad. This product is intended to replace the existing Pontis 4.x product line with a next-generation software application, utilizing state-of-the-art software development technology. Pontis 5.0 will build on and preserve licensee investments of time and money in Pontis 4.x implementation.

The next generation of Pontis eventually will offer the full functionality of the currently deployed Pontis 4.4. The baseline capabilities will be extended to provide licensees the ability to access selected modules from a web browser as well as a dedicated client application, depending on the particular BMS activity. A web version of Pontis will offer easier application setup, administration, and implementation, and will provide a straightforward migration path for existing Pontis 4.x users.

Pontis 5.0 will support a range of new functionality, including: import and export of data in eXstensible Markup Language (XML) based on the TransXML schema being developed through National Cooperative Highway Research Program (NCHRP) Project 20-64, XML Schemas for Exchange of Transportation Data (TransXML); improved approaches to modeling bridge needs through incorporation of research results from NCHRP 12-67, Multiple-Objective Optimization for Bridge Management Systems; improved user interfaces for bridge-level analysis and project planning; and support for multiple asset types. Further, Pontis 5.0 will provide seamless support to its licensees for potential changes in the National Bridge Inventory (NBI) coding standards.

The development of Pontis 5.0 is driven in part by new functional requirements as well as by dramatic technology changes in the software development arena that have overtaken the existing client/server application, particularly in the web application development domain.

Specific Goals for Pontis 5.0

The goals for Pontis 5.0 are to provide the AASHTO member agencies and licensees with:

- A **readily-accessible**, **robust** repository for bridge information, including data inventory, condition, needs, project plans, and accomplishments;

- **Technically correct capabilities** to perform bridge management tasks of all types;
- A **technologically up-to-date** development foundation;
- **Streamlined, simpler application and deployment** mechanisms;
- A **flexible, customizable** application which users can extend to solve agency-specific bridge management tasks; and
- **Preservation of significant agency investments** in BMS generally and Pontis specifically, including expenditures of time, money, training and other agency resources.

The Technical Architecture Description (TAD) identifies additional goals and concerns of a number of Pontis stakeholders, which are reflected in these summary goals.[4]

In this context, the bold-faced terms appearing above are defined as follows:

Readily accessible means that information stored in Pontis 5.0 should be available to most users via Microsoft Internet Explorer from any computer via a network or Internet connection; and the software enables users to access the majority of data and capabilities without specialized technical skills or training.

Robust means that: a) the repository is built on a commercial, high-quality relational database management system (RDBMS); b) the information contained in the system is secure, with access limited to appropriate staff and agency-designated parties; c) information contained in the system can be backed up in a secure location on a regular schedule; and d) the system is reliable and in operation at an acceptable levels of availability and performance. Pontis 4.4 and 5.0 address item (a) and (d) but assign primary responsibility for (b) and (c) to licensing agencies.

Technically correct capabilities means that the various analysis and calculation in Pontis 5.0 operate consistently and comparably with existing versions, and where there may be opportunities for improvements or innovation in technical capabilities these are well-understood, well-defined, and verifiable in the deployed application. As an example, continued proper calculation of NBI Translator results will be part of achieving this goal.

Technologically up-to-date mandates the design and development of the next generation Pontis application using next generation tools, such as .NET, web services, SOAP, XML, Model-View-Controller architecture, and others.

Streamlined, simpler administration means that agencies will find it easier and faster to install, deploy, and upgrade the application, and manage its operation.

[4] Cambridge Systematics, *Pontis Technical Architecture – A Future Vision*, 2003, pp. 5-6.

Pontis 5.0 must address this objective explicitly and must be significantly improved over current version(s).

Flexible, customizable means that the application provides innate mechanisms for extending and adapting the database schema, foundation capabilities, and user presentation elements such as forms and reports. This has been a key benefit of the existing product. The future Pontis 5.0 product should be capable of adapting to changes in data requirements, reporting, standards, and other influencing factors.

Preservation of agency investments means that the significant body of effort to develop staff expertise, data, reports, data forms, procedures, manuals, training courses, and the multitude of other BMS-related agency products must be explicitly acknowledged and preserved in Pontis 5.0 to the greatest extent possible. This goal also means providing a straightforward migration path for Pontis and integrated BRIDGEWare databases when moving to Pontis 5.0.

We explicitly recognize that these priorities and needs may change over time, and that the functional requirements and technical design may necessarily adapt to accommodate these changes in priorities and needs.

2.2 STAKEHOLDERS

The TAD identifies Pontis stakeholders in some detail, and these continue to be applicable. The TAD outlines issues of concern to each stakeholder class which form a basis for identifying constraints on the Pontis 5.0 functional requirements. The stakeholders defined in the TAD include:

- **End users**, grouped into:
  - *power users*, who typically would manage an agency's Pontis database and/or make heavy use of its analytic/modeling capabilities;
  - *routine users*, such as bridge inspectors or staff data analysts, responsible for primary data collection, quality reviews, and normal data reporting. These are the predominant set of Pontis users; and
  - *casual users*, such as planning or programming staff using Pontis inventory data and analytical results.
- **Agencies**: the governmental or other entities responsible for procuring and operating Pontis;
- **AASHTO:** the software product's owner;
- **Developers:** those responsible for building and maintaining BRIDGEWare components, including third party developers;
- **Integrators:** those who integrate Pontis services or data with other tools and systems; and

- **Deployers:** those responsible for installing, configuring and customizing Pontis for use.

All the stakeholders are still important, but for the requirements development phase, we are highlighting the most important stakeholders concerned with reviewing and approving requirements in this FRS. The table below identifies the key front-line Pontis stakeholders and outlines their role with respect to the product.

Stakeholder	Role with respect to Pontis 5.0
Pontis Users	Day-to-day Pontis users, requirements initiation, requirements review and comment.
Pontis Task Force	Product quality and effectiveness, technical correctness, oversight and project requirements, decision-making.
Pontis 5.0 Requirements Technical Advisory Group (TAG)	Development of Pontis 5.0 requirements, architecture, and implementation plan.
BRIDGEWare Integration TAG (formerly, Database TAG)	BRIDGEWare impacts assessment, implementation plan review, schedule coordination, design and development technology coordination, database design coordination.
AASHTO	Product quality and effectiveness, overall project management, marketing, licensing, revenue development.
Contractor	Responsible for developing Pontis software.

2.3 USERS OF THE PRODUCT

Pontis users are classified as described in the following paragraphs for the purposes of this requirements document.

- **Inspectors:** inspectors may be agency staff inspecting bridges full or part-time/seasonally, or consultants. Briefly, their responsibilities entail original bridge data collection, review, and database quantification. Their activities may also include developing field work recommendations and accomplishment verification. These users may or may not be licensed engineers;

- **Bridge project planners:** staff responsible for development, review, approval, and tracking of bridge work of all types, whether maintenance, improvements, rehabilitation or replacement;

- **Bridge management engineers:** staff responsible for multiple aspects of bridge management, concerned with inspection program accomplishment, program development and implementation, bridge policies, and other overarching concerns such as truck routing and permit review;

- **Bridge design/rating engineers:** staff responsible for bridge design and rating, who may need information from Pontis when making engineering decisions, and who may contribute structure design and capacity information to Pontis;
- **Highway information analysts:** staff responsible for collection and maintenance of various data of concern to bridge management such as roadway service levels, traffic, routing restrictions, etc;
- **Highway program planners:** staff concerned with integration of project plans within highway corridors, and in particular, to acknowledge, review, and incorporate bridge project recommendations originating from bridge management or other sources, and to provide feedback to bridge management on project accomplishments; and
- **Other Pontis users:** staff with interest in bridge information such as executives, public relations officers, engineering firms (primarily serving as inspection contractors), and other casual users of the system. These users may or may not have an engineering background and are primarily concerned with data review, performance measures identification and tracking, and summary inventory/condition reports.

Clearly some of these users fall into secondary and tertiary categories. Where there is a conflict, the requirements of the key internal users must take priority over those of secondary users. The tertiary category of users may make use of Pontis, but have no vested interest in the product. The requirements of these users are of lowest priority.

2.4 PROJECT CONSTRAINTS

This section describes constraints on the requirements and the eventual design of the product. These constraints reflect a number of stakeholder concerns identified in the TAD.

Development Technology Constraints

Pontis 5.0 will be developed using Microsoft technologies, as follows:

- Pontis 5.0 will be a Microsoft .NET application consisting of a combination of stand-alone, thin client, and back-end server applications. There may be duplicative capabilities between the thin-client and the stand-alone implementations. Providing a stand-alone field inspection application for disconnected users as well as a web inspection interface is an example.
- The thin client will be designed and tested for the latest version of Microsoft Internet Explorer, version to be determined during design.

- The stand-alone application will require workstations with the .NET Runtime Framework installed and the Windows XP operating system. The specific version will be determined during design. While the .NET Framework is distributed as part of the Windows operating system and with .NET applications for end users, .NET server licenses likely will be needed as well, which may mean an expense for licensing agencies.
- The server application will expect and rely on the .NET Framework and the common language runtime (CLR).
- Microsoft Visual Studio .NET will be used as the primary development tool for Pontis 5.0. The development languages will be heterogeneous and determined appropriately for the application context and therefore may include Visual Basic .NET, C#, ASP.NET scripting, or others as appropriate.
- Pontis 5.0 will maximize the use of third-party code libraries, whether fee-licensed or open-source, to minimize original development costs to the greatest extent possible.

General Solution Constraints

- The Pontis 5.0 thin-client will be designed so it can be made available to users via agency internal network and over the Internet, as appropriate for different application contexts, with final deployment arrangement determined by each licensee.
- Pontis 5.0 will take full advantage of web display capabilities including, in particular, screen real estate management to minimize scrolling, paging and tabbing.
- The BRIDGEWare database design for Pontis 5.0 will be consistent with the database design for Pontis 4.4. Changes are very likely to be necessary but proposed database changes will need to be reviewed and approved by the BRIDGEWare Database Technical Advisory Group (TAG) as part of the Pontis 5.0 design approval process following the agreed database change protocol.[5]
- Pontis will support its existing relational databases (RDBMS) of Sybase Adaptive Server Anywhere, Oracle, and Microsoft SQL Server as the database for its information repository. The versions of each vendor's RDBMS will be determined as part of the software design and

[5] The BRIDGEWare Task Forces have established a specific protocol for requesting, reviewing, and approving any database changes. Each proposed change is described, prioritized, and assigned to an impact/threat category. Substantive justification must be provided for every change, and no changes are permitted to the BRIDGEWare database design without the consensus of the database Technical Advisory Group (TAG).

implementation plan. No additional database support is anticipated as of this writing, but the design will adhere strictly to ANSI-SQL and ODBC/JDBC/.NET standards in order to maximize the potential to support additional databases in the future.

- Pontis bridge information will integrate with a more general asset management database structure which may incorporate other types of assets such as sign structures, high-mast towers, walls, pavements, or buildings.
- Pontis 5.0 will be a GIS-aware application that must operate properly within an agency's GIS technology whether ESRI, Intergraph, Open GIS, or others. Pontis must observe industry standard protocols in this regard and avoid vendor specific implementation approaches.

Anticipated Application Workplace Environments

Pontis 5.0 will be used both in field contexts, such as an open-air bridge inspection, where environmental factors (weather, sunlight) may influence program operation and no network connectivity may be assumed, as well as in a typical agency office environment, which will impose no special restrictions or requirements on the design or operation of the system.[6]

2.5 NAMING CONVENTIONS AND DEFINITIONS

This document relies on transportation and information technology terms that are in common use and do not require detailed definitions. Where a conflict arises, the current Pontis 4.x terminology will apply.[7]

The development of Pontis 5.0 will adhere to a separate software development standards document which will itself incorporate naming conventions.

2.6 RELEVANT FACTS AND ASSUMPTIONS

This section describes assumptions made by the project team as well as any known external factors that affect and/or constrain the Pontis 5.0 system. A brief discussion is presented here. Where appropriate, the external factors may be elaborated further in separate memoranda or the detailed design.

- Pontis 5.0 must not assume a uniform operating environment and must accommodate both disconnected and connected users.

[6] No special support for tablet or handheld computers is anticipated at present but supporting this type of technology may well become a design constraint at some point.

[7] The reader is invited to review the Pontis 4.x User and Technical Manuals, and in particular, the Glossary of Terms, as well as the Federal NBI coding guide for bridge inspection terminology.

- Pontis 5.0 must maintain consistency with the existing deployed 4.x application, which in effect serves to a great extent as a hi-fidelity (hi-fi) prototype for Pontis 5.0.
- The Pontis 5.0 design must incorporate any Federal NBI changes that may arise during the design and development period.
- Pontis 5.0 must use technology that the member agencies of AASHTO can deploy and support, hence, the general constraint identified earlier – that Pontis 5.0 will be a Microsoft .NET application, utilizing industry standard RDBMS technologies.
- The actual development strategy will minimize original coding in favor of utilizing third-part libraries, even at the expense of possibly foregoing some functionality that custom coding could provide. It should be understood clearly by reviewers that this sort of tradeoff of cost and functionality may well retroactively influence the final Pontis 5.0 requirements[8]. This further implies that some current Pontis 4.4 capabilities may be implemented very differently in a future version in order to take advantage of available commercial and open-source code.
- Pontis 5.0 will be designed as a deployed/installed application in agencies, and not designed as a hosted application, although it may well operate perfectly acceptably in a hosted configuration, such as a Citrix environment.
- Pontis 5.0 American Disabilities Act (ADA) compliance[9] constraints on the functional requirements will be determined by AASHTO in consultation with its member agencies. The level and approach for achieving compliance remains an open issue to be resolved[10].
- Pontis 5.0 will be a licensed product, which AASHTO will maintain ownership over, and which will be deployed with license protection to certain licensees such as consultants, international agencies, and academic institutions[11].
- Pontis 5.0 must interact effectively and properly with the other BRIDGEWare products.

[8] As of this version, no particular third-party software has been selected for use in Pontis 5.0.

[9] Section 508 of the Rehabilitation Act: Accessibility in the Information Age can be found at http://www.usdoj.gov/crt/508/report2/web.htm

[10] No particular accommodation for ADA compliance is currently incorporated in the requirements as of this revision.

[11] Copy protection is not enforced in Pontis 4.x for U.S. licensing agencies and it is assumed that policy will prevail for Pontis 5.0. The arrangement for licensing web application users is to be determined at a later stage.

- Pontis 5.0 must provide two-way XML data exchange with TransXML schema once it is established.

- The design of Pontis 5.0 will be informed by review of the results of NCHRP Project 12-67, Multiple-Objective Optimization for Bridge Management Systems. Additional funding may be required to modify the Pontis modeling approach to incorporate new approaches developed as part of this project. The estimates in this document to not attempt to quantify such additional funding requirements.

- Pontis 5.0 must provide well-defined and documented access to selected product functionality and data for third-party developers.

- The Pontis 5.0 business rules will be similar to those of Pontis 4.x.

- Where a conflict arises between 5.0 requirements and 4.x behavior, or when a 4.x feature is being abandoned or significantly reworked, this design decision will be explicitly documented and justified. For example, the Pontis formula language and support for Pontis 2.0 file exchange formats are very likely to be discarded in Pontis 5.0. A conflict or concern may arise where there is an explicitly identified improvement to the 4.x functional design, such as the organization and operation of the Select tool for selecting bridges on the Pontis desktop in a Web deployed environment. An agreed procedure and protocol will be needed to facilitate making a collective decision on these sorts of highly visible functional changes.

- The Project Planning module will be redesigned and replaced.

3.0 Use Cases

3.1 BROWSE BRIDGE & PROJECT DATA

Primary Actor: All Pontis users

Scope: This scenario covers browsing through the bridge and project data.

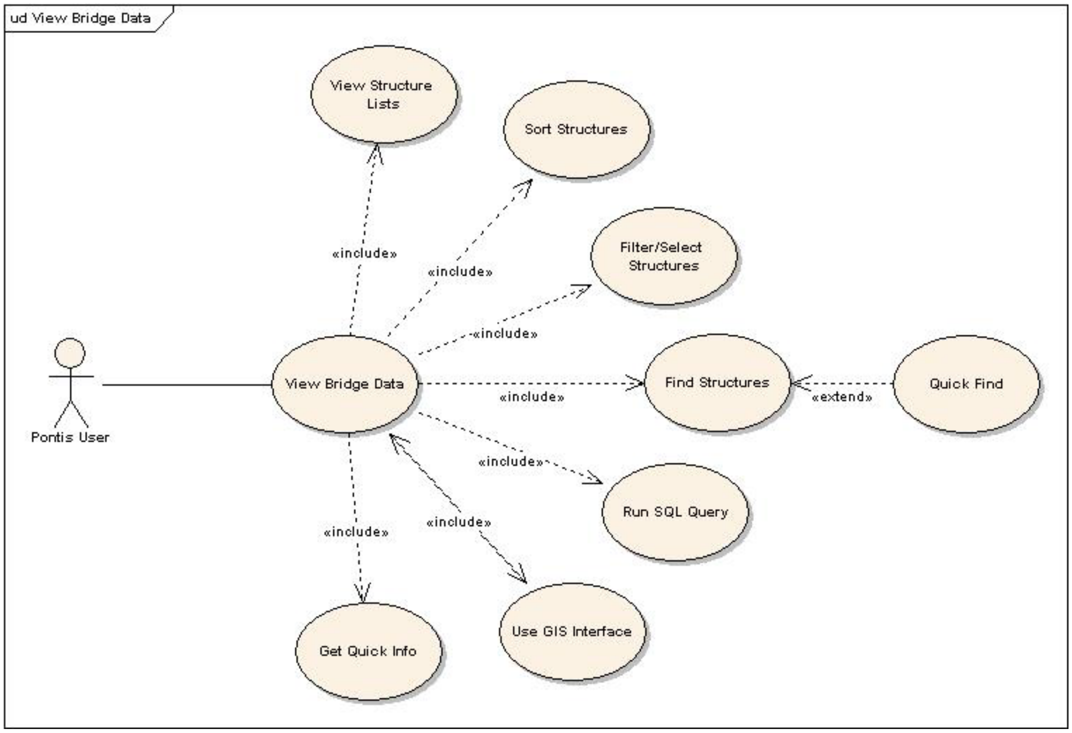
Stakeholders: All Pontis users

Precondition: User is logged into Pontis and has the necessary permissions.

Use Case 1: View Bridge Data

Description: The user uses the find, filter, and select capabilities in Pontis to access the desired structure. A map-based query can be performed.

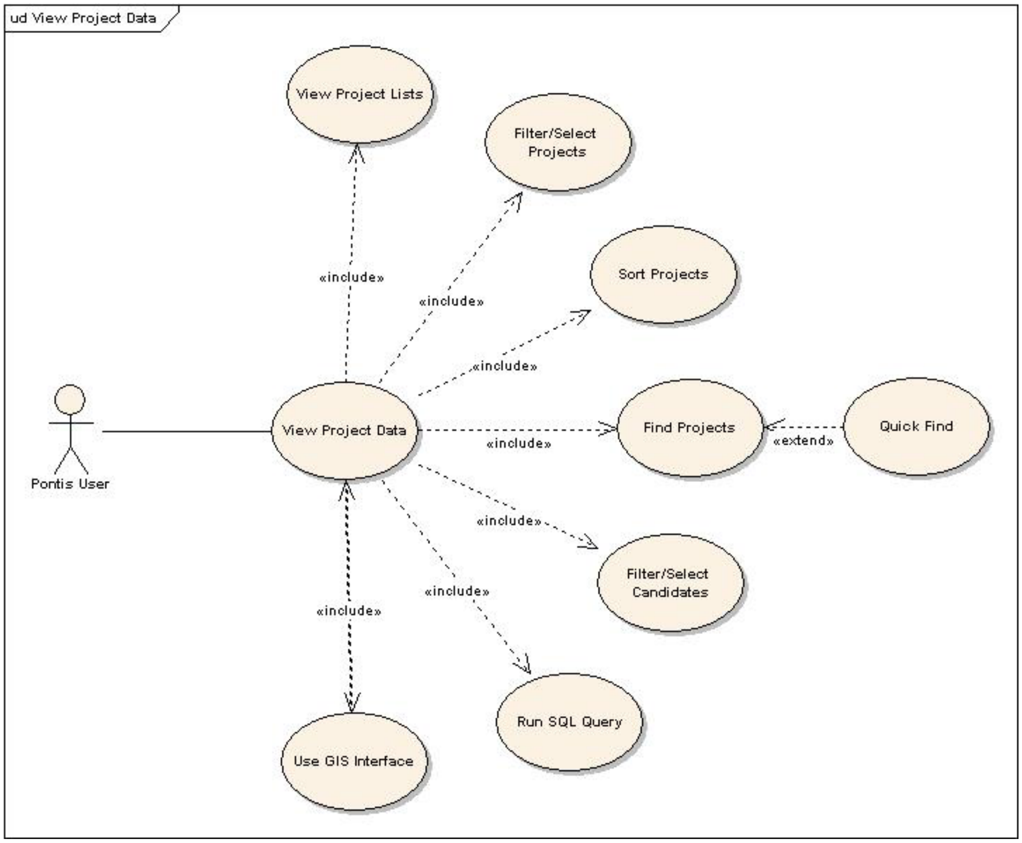
Success End Conditions: User finds and selects the desired structure(s). User views data for the selected structure(s).



Use Case 2: View Project Data

Description: The user uses the find, filter, and select capabilities in Pontis to access the desired project. A map-based query can be performed.

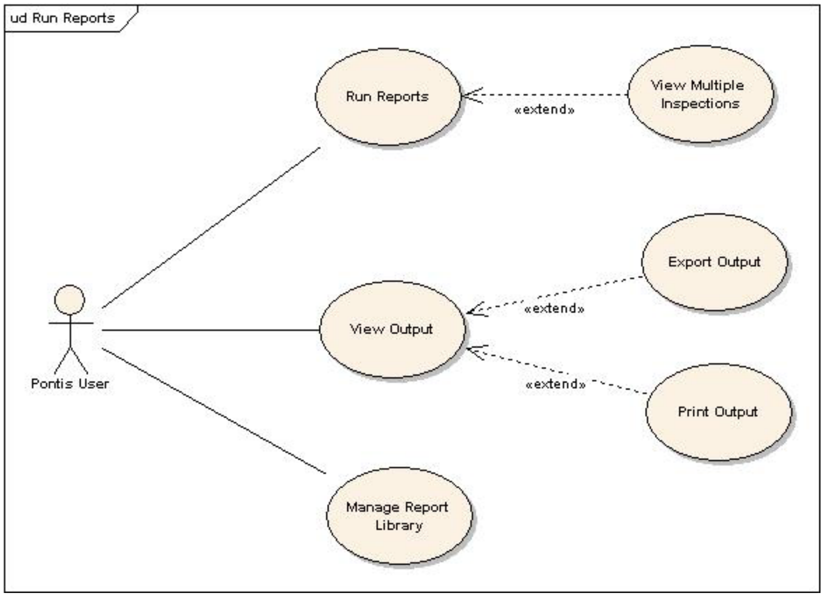
Success End Condition: User finds and selects the desired project(s). User views data for the selected project(s).



Use Case 3: Select and View Reports

Description: The user selects a report to run against the Pontis database.

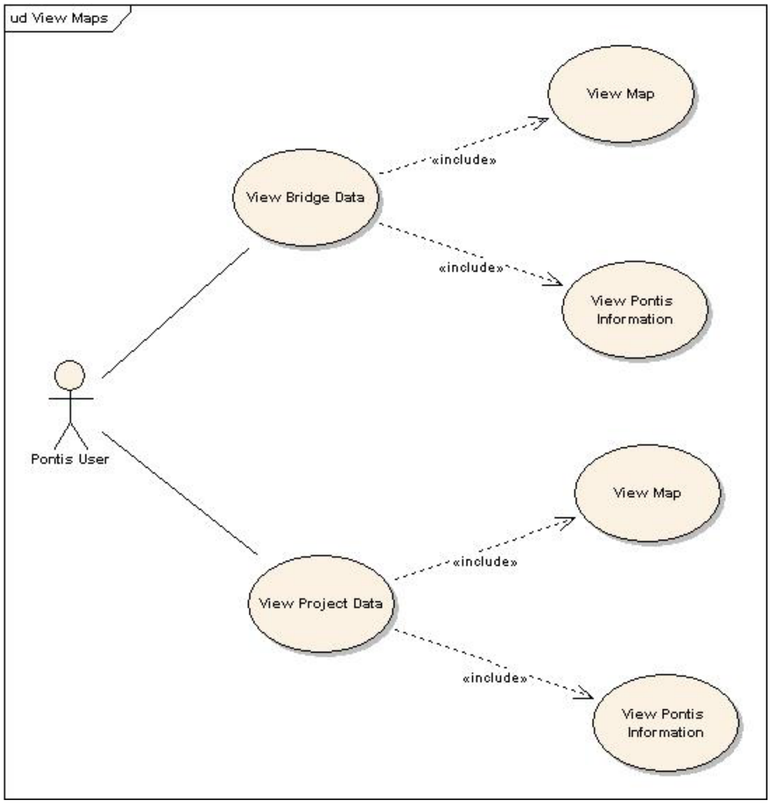
Success End Condition: The selected report is generated and displayed on the screen.



Use Case 4: Select and View Pontis Information Using Maps

Description: The user selects data (bridges and projects) from the Pontis database to generate a map display, or the user selects bridges or projects from a map and sees the supporting Pontis information.

Success End Condition: The selected Pontis information displayed on the screen or a map is generated showing the Pontis information.



3.2 Bridge Inventory & Inspection

Primary Actor: Routine user

Scope: This scenario covers entering/editing inventory and inspection data.

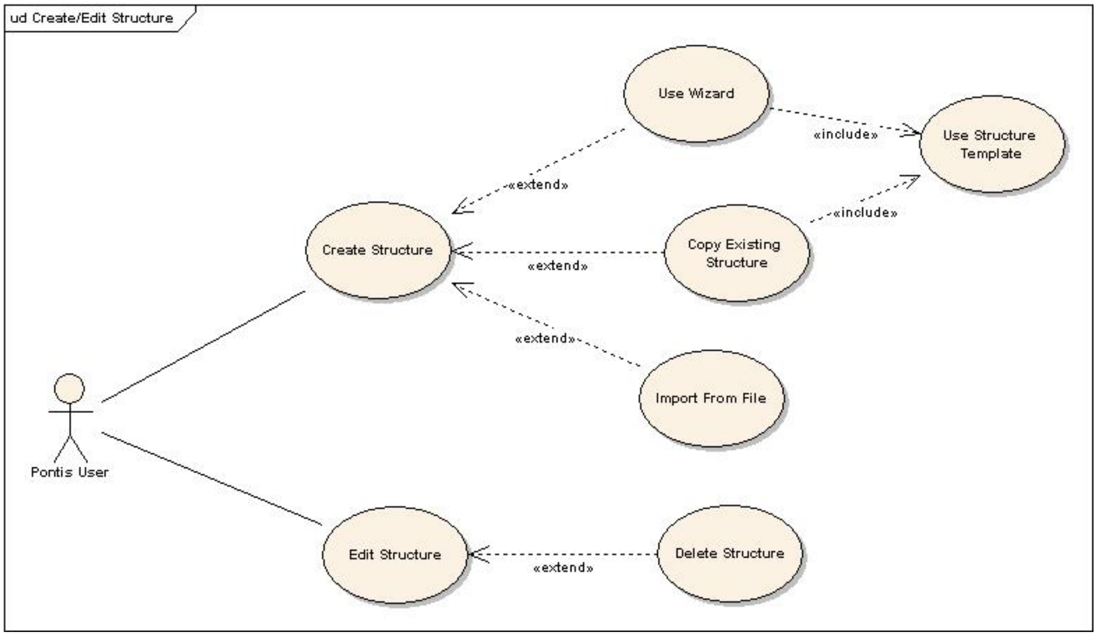
Stakeholders: Inspector

Precondition: User is logged into Pontis and has the necessary permissions.

Use Case 5: Create/Edit Structures

Description: The user creates a new structure by copying an existing structure or manually entering in data for the structure. The user can also modify inventory data for an existing structure.

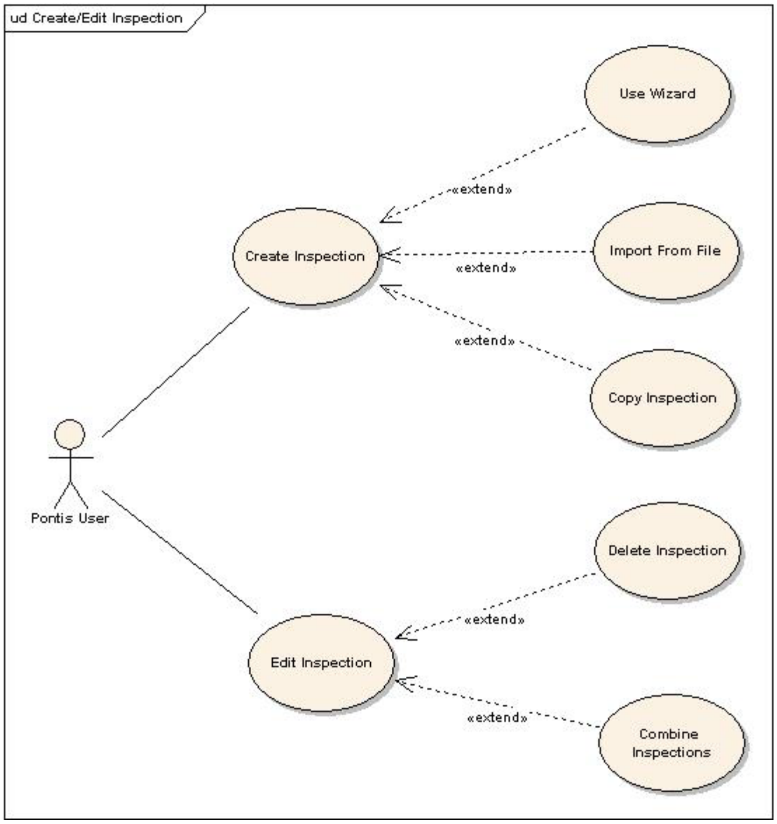
Success End Condition: Data for the new structure and the modifications to the existing structure are stored in the Pontis database.



Use Case 6: Create/Edit Inspections

Description: The user creates a new inspection for a structure by copying the last inspection on the structure, or manually entering in data for the new inspection. The user can also modify existing inspection data for the structure. When appropriate, results from multiple inspectors working together may be merged for one structure to create a single combined or consolidated inspection.

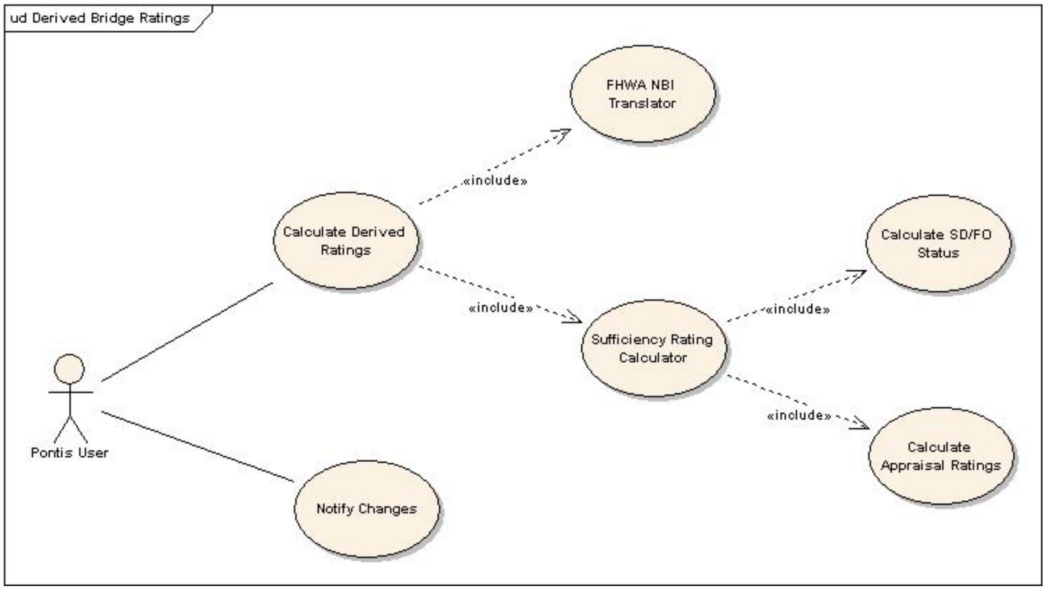
Success End Condition: Data for the new inspection and the modifications to the existing inspections are stored in the Pontis database.



Use Case 7: Calculate Dependent/Derived Inspection Results

Description: The user calculates derived ratings (NBI condition ratings, structural rating, geometric rating, sufficiency rating, and SD/FO status) based on inspection data. Users may override any calculated NBI condition rating such as deck, superstructure, substructure or culvert condition as necessary.

Success End Condition: The dependent ratings are calculated and, if accepted by the user, are stored in the Pontis database. Ratings for deck, superstructure, and substructure, or culvert, condition can be entered directly to override calculated results.


ud Derived Bridge Ratings
FHWA NBI Translator
«include»
Calculate Derived Ratings
Calculate SD/FO Status
«include»
Sufficiency Rating Calculator
«include»
«include»
Pontis User
Calculate Appraisal Ratings
Notify Changes

3.3 PRESERVATION MODEL DEVELOPMENT

Primary Actor: Advanced user

Scope: This scenario covers the development of the element preservation policy.

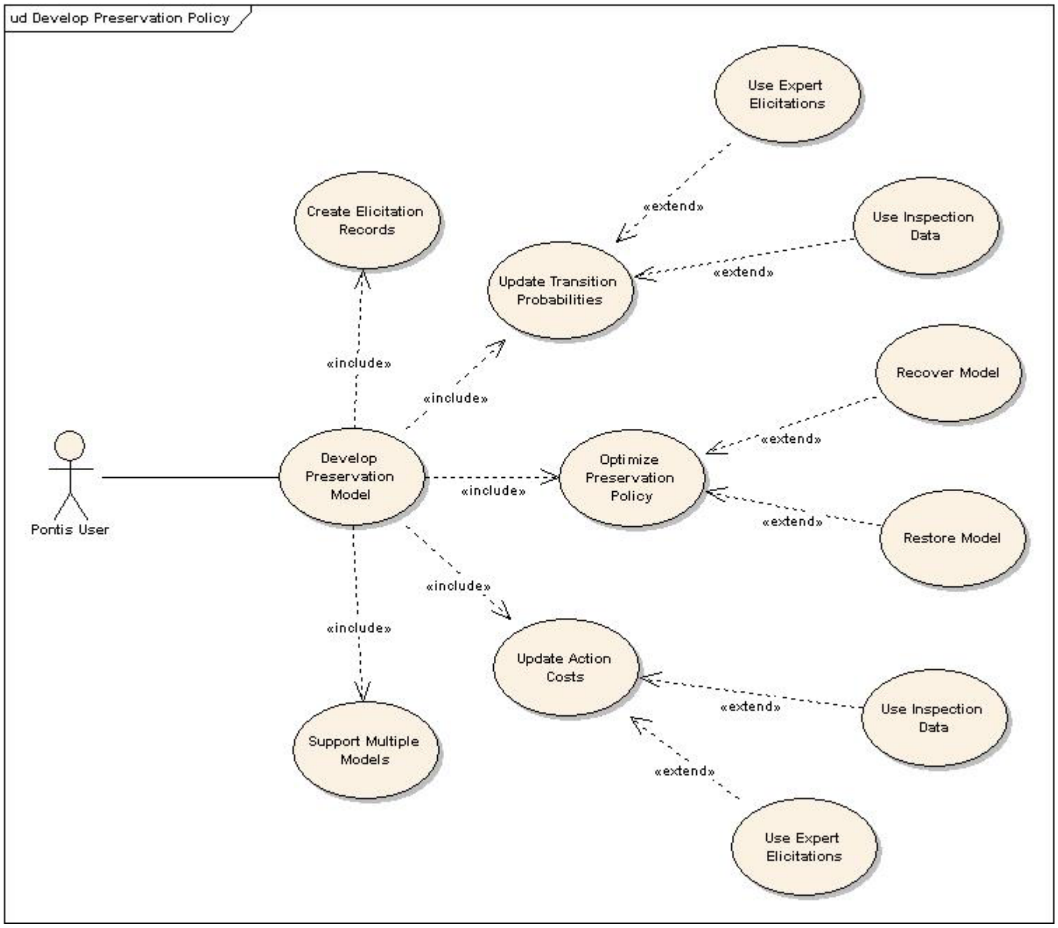
Stakeholders: Engineering

Precondition: User is logged into Pontis and has the necessary permissions.

Use Case 8: Develop Preservation Policy

Description: The user updates the deterioration probabilities and preservation action costs based on expert elicitation and/or inspection data. The costs and probabilities are used to develop the optimal preservation policy.

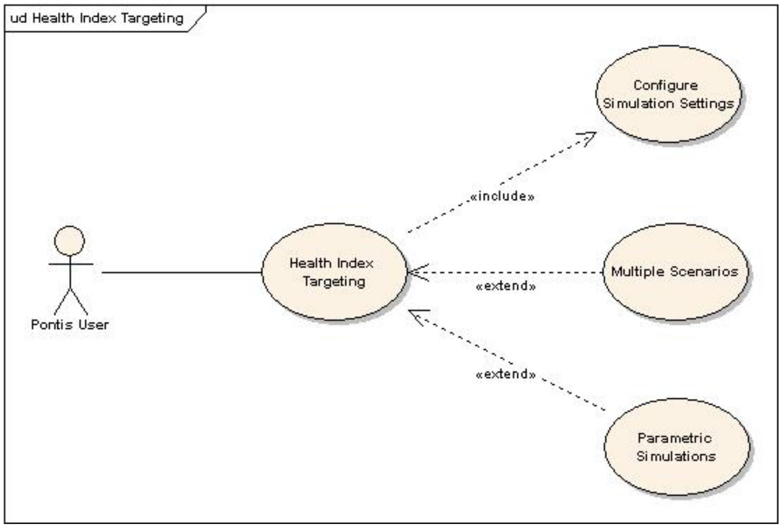
Success End Condition: Updated preservation policy data is stored and displayed.



Use Case 9: Perform Health Index Targeting

Description: The user runs the health index targeting process to determine the long term cost of meeting an average health index target.

Success End Condition: The results of the HI targeting process are displayed.



3.4 PROGRAM SIMULATION

Primary Actor: Advanced user

Scope: This scenario covers all the activities related to running a programming simulation or a bridge simulation.

Stakeholders: Engineering/Planning

Precondition: User is logged into Pontis and has the necessary permissions to use and potentially modify the bridge inventory, inspection history, work recommendations, and project planning data, as well as the necessary model information.

Use Case 10: Configure Simulation Parameters

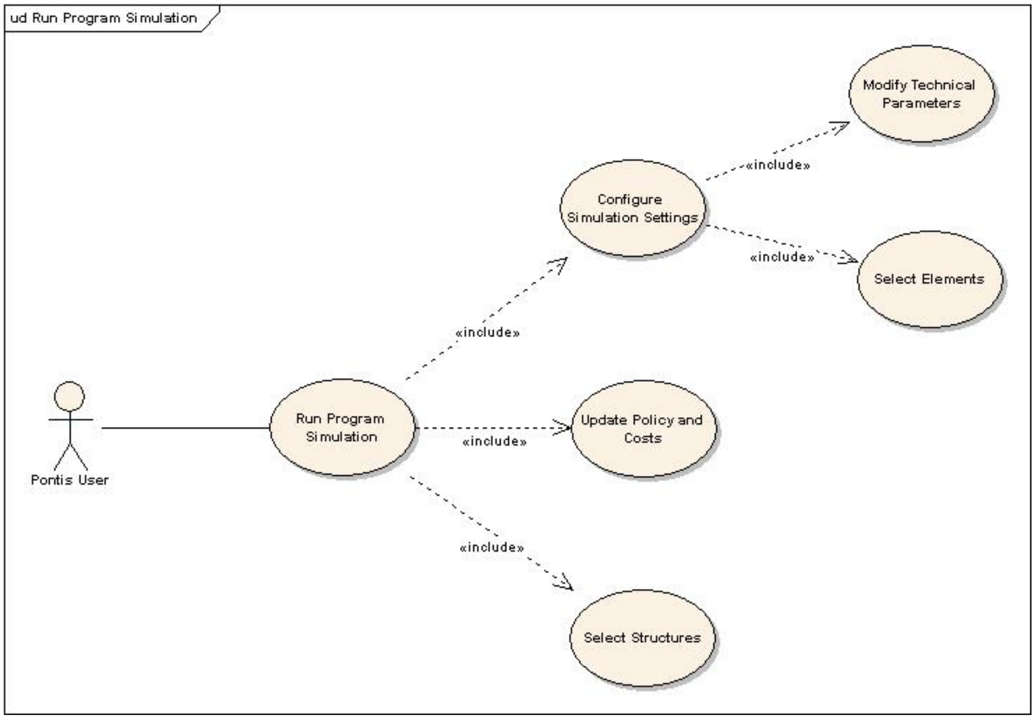
Description: The user specifies all settings for the simulation, including the improvement model parameters, agency rules, the simulation time frame, and the annual budget.

Success End Condition: The simulation settings are stored under a scenario name.

Use Case 11: Perform Program Simulation

Description: The user runs a program simulation for one or more structure(s) with the specified settings in order to view network trends and generate bridge-level work recommendations.

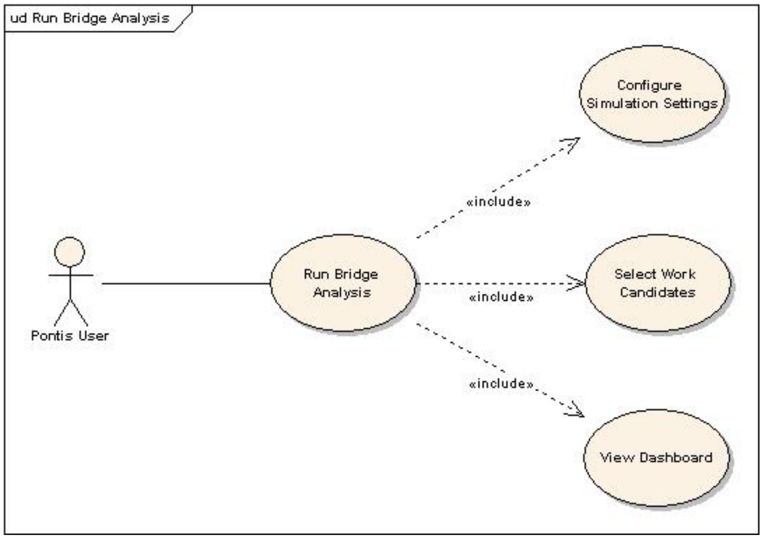
Success End Condition: The results of the program simulation are accurately generated and stored in the Pontis database.



Use Case 12: Perform Bridge Analysis

Description: The user performs a bridge analysis for a structure in order to view the impact of one or more work items on the structure condition.

Success End Condition: The results of the bridge analysis are accurately generated and stored in the Pontis database.



3.5 Project & Program Development

Primary Actor: Advanced user

Scope: This scenario covers all activities related to creating and editing projects and programs.

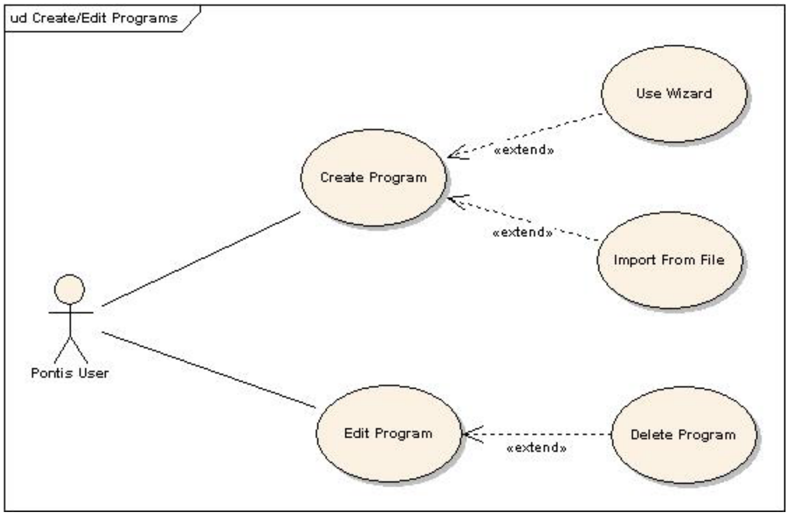
Stakeholders: Bridge Engineer/Planner/Projects Engineer

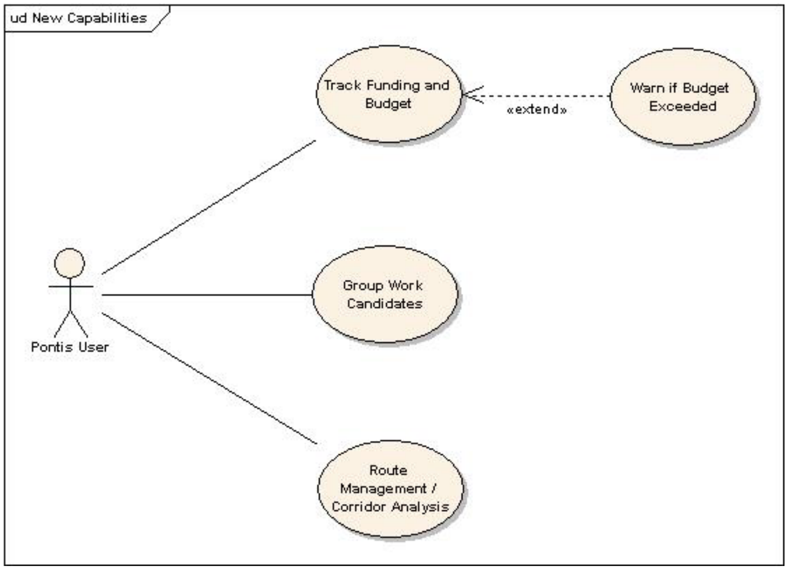
Precondition: User is logged into Pontis and has the necessary permissions.

Use Case 13: Create/Edit Programs

Description: The user creates a new program and specifies the budget for the program by funding source. The user can also edit an existing program.

Success End Condition: Data for the new program and the modifications to the existing program are stored in the Pontis database.

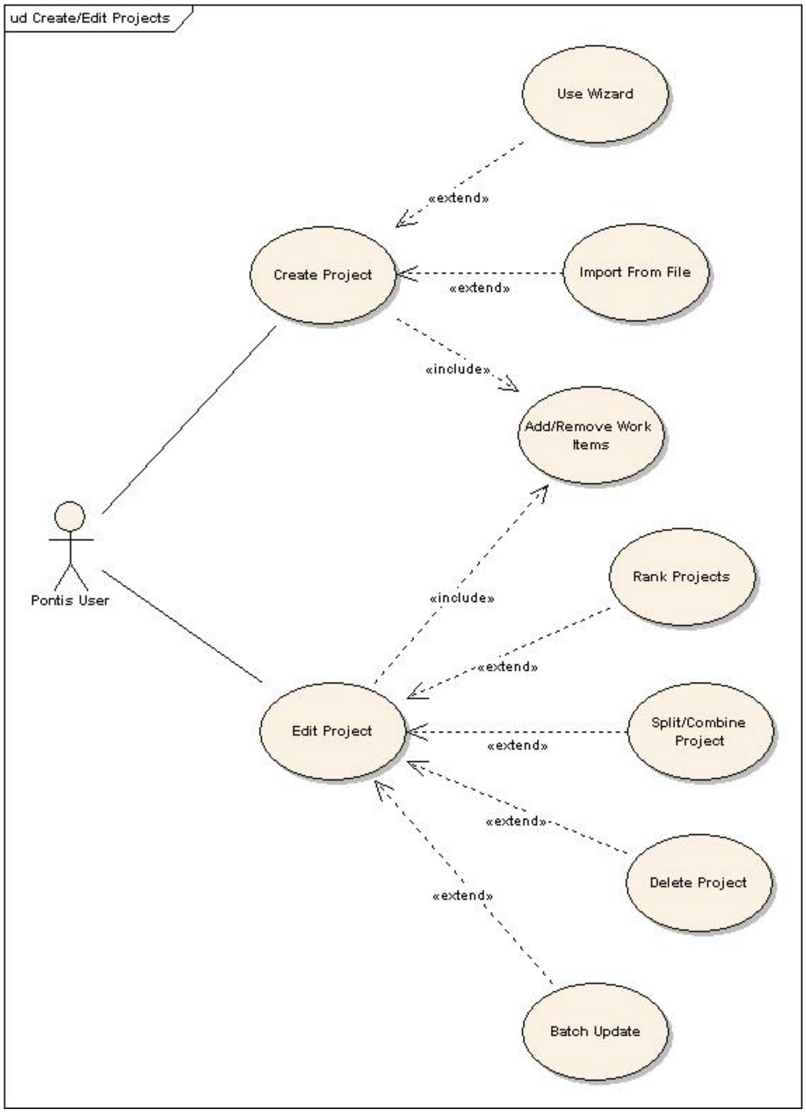




Use Case 14: Create/Edit Projects

Description: The user creates a new project by selecting Pontis-generated work recommendations, inspector work candidates, and user-specified work items, or edits data for an existing project. The user can also edit, split, or combine existing projects. Project information can be imported from a file to create or update projects.

Success End Condition: Data for the new project and the modifications to the existing project are stored in the Pontis database.



3.6 DATA MANAGEMENT

Primary Actor: Inspector, routine and advanced users

Scope: Data Management

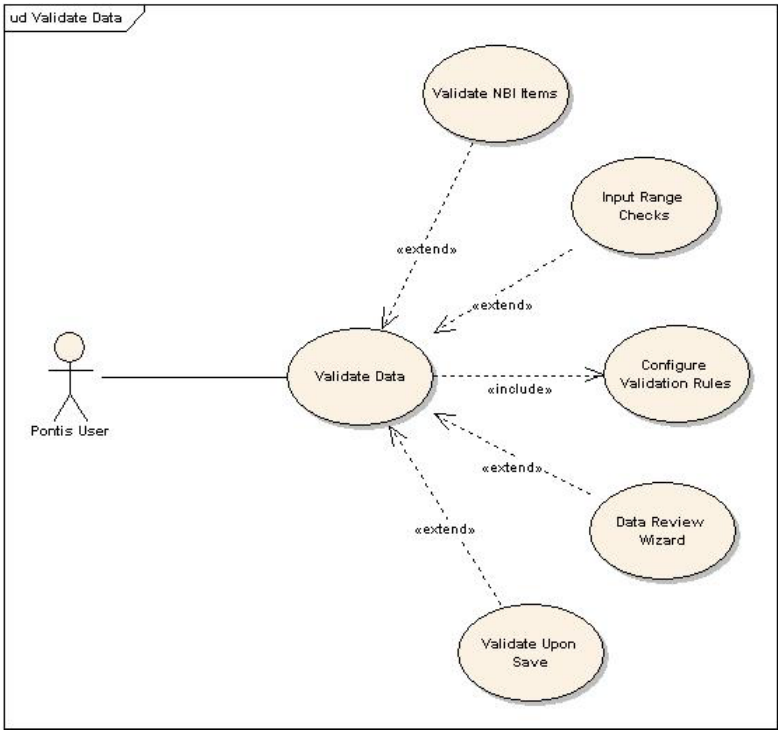
Stakeholders: Inspector, Routine and Advanced User

Precondition: User is logged into Pontis and has the necessary permissions.

Use Case 15: Perform Data Validation

Description: Perform validation, cross-validation, and quality assurance tasks in context and display the result.

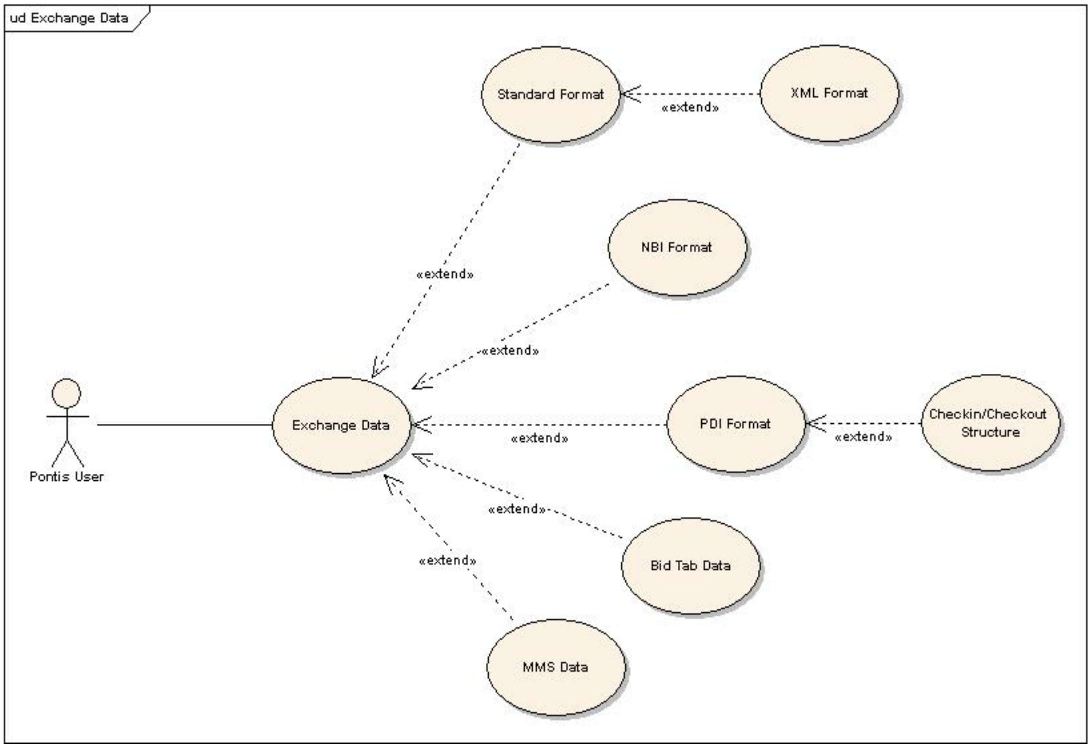
Success End Condition: Invalid data items are listed as a result.



Use Case 16: Exchange Data

Description: Export Pontis data in various formats including NBI, PDI, and XML. Import the data from various sources into Pontis database. Import and export data using the TransXML schema.

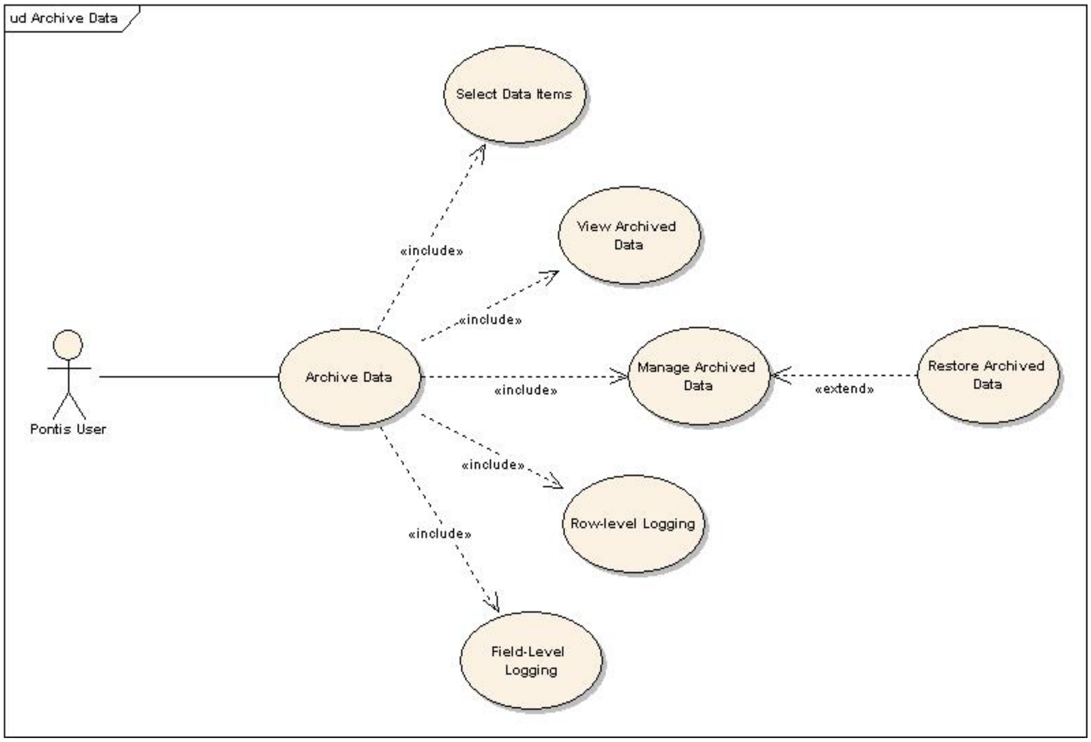
Success End Condition: For the export operation, the specified data are written to a specified format. For the import, the data are correctly read from the file and/or the existing data are updated, and new or modified data is saved to the database.



Use Case 17: Perform Data Archiving

Description: User selects data items (e.g. inventory, inspection and/or project data for removed structures) to be archived.

Success End Condition: Selected data items are archived.



Use Case 18: BRIDGEWare Integration Support

Description: Perform database integration task with other BRIDGEWare products. Transfer load rating data from Virtis to Pontis.

Success End Condition: BRIDGEWare databases are successfully integrated.

3.7 SYSTEM ADMINISTRATION

Primary Actor: Power user

Scope: System Administration

Stakeholders: End users, Agency

Precondition: User has the necessary permissions.

Use Case 19: Define and Manage User Roles.

Description: An administrator creates/edits user roles with different security settings.

Success End Condition: New user roles are added to the system or existing roles are modified.

Use Case 20: Create New Users

Description: An administrator adds a new user to Pontis application and assigns him/her to a role.

Success End Condition: New users are added to the system with designated privileges.

Use Case 21: Perform User Authentication

Description: When a user logs into Pontis, it validates the user name and password. Access to Pontis functionalities are controlled depending on the user role.

Success End Condition: Connect and run Pontis with designated privileges.

Use Case 22: Configure Application Functionalities

Description: User can change the application configurations.

Success End Condition: Modified configuration are updated correctly And application operation changes immediately, or after restart, as appropriate, to reflect the new configuration settings.

Use Case 23: Configure User-Interface Presentation

Description: User modifies the user interface items.

Success End Condition: The changes are saved and applied correctly.

Use Case 24: Administration Functionalities

Description: User manages the administration functionalities that affect application operation.

Success End Condition: The admin functionalities are executed as defined by the administrator.

4.0 Functional Requirements

This section of the report outlines the functional requirements for Pontis 5.0 – the specific tasks and activities that the software enables, other functions needed to support the software, and the data that are required to support these requirements. The requirements are organized loosely around a set of Pontis 5.0 use cases. Each requirement is numbered for convenience. Please note that the "may" requirements in version 0.2 of the FRS have been moved to Section 7.4 - Waiting Room - for future review and discussion.

A separate database of requirements presents each of these narrative requirements generally following the Volere Requirements template specification and may be reviewed in conjunction with this document. In the Pontis 5.0 requirements database, requirements that apply only to the thin-client or thick-client implementations are tagged for clarity. Where appropriate, requirements that involve configurable control of program behavior are noted. For example, it may be that a user is not allowed multiple sessions of the application by agency policy, hence the requirement has a configuration implication.

The numbering convention below is to first identify the use case e.g., *UC-1 View Bridge Data*, then to document individual requirements within the use case e.g., …

1.3 The user shall be able to select structures by structure ID, district, county, administrative area (geographic group), ownership, custodianship, functional class, NHS status, on/off system, inspector, bridge group, inspection due dates, and all or some environment and elements. The query shall be restricted depending on the user privilege.

…

Requirements surrounded by boxes are new or significantly updated requirements for Pontis 5.0.

4.1 Browse Data

UC-1 View Bridge Data

1.1 The user shall be able to view the data stored in the Physical Inventory tables through the Pontis graphical user interface (GUI).

1.2 The user shall be able to switch between predefined structure lists.

1.3 The user shall be able to select structures by structure ID, district, county, administrative area (geographic group), ownership, custodianship, functional class, NHS status, on/off system, inspector, bridge group, inspection due dates, and all or some environment and elements. The query shall be restricted depending on the user privilege.

1.4 The user shall be able to find structures based on the structure ID, structure name, feature intersected, facility carried, route number, LRS Inventory Route, KM/mile posts, structure types, design types and materials, and by searching the notes fields. The query shall be restricted depending on the user privilege.

1.5 The user shall be able to enter a WHERE clause of the SQL query in the Find and the Filter window and run a query against the bridges in the database. The query will be verified by the application and SQL errors will be reported to the user for correction. This capability shall be an application privilege. Bridge-level security of the user shall be applied automatically.

1.6 The user shall be able to select a structure by directly typing in the structure ID in GUI, with the display scrolling to that structure. Bridge-level security of the user shall be applied automatically.

1.7 The system shall display the data in the appropriate unit of measurement as configured for the application. This will be a permanent modal setting for the application. This setting shall be configurable (as stipulated in Requirement 22.1).

1.8 The user shall be able to select bridges on a supported map display and generate a bridge list for use within Pontis, for example, to perform a simulation analysis or to generate an inspection roster.

UC-2 View Project Data

2.1 The user shall be able to view the data stored in the Project Planning tables through the GUI.

2.2 The user shall be able to switch between predefined project lists.

2.3 The user shall be able to filter and select projects by structure ID, project ID, program, action type, project status, review status, treatment, program year, and project end date.

2.4 The user shall be able to find projects by project ID, project name, project status, structure ID, and program.

2.5 The user shall be able to filter and select work candidates by structure ID, project ID, program, action type, project status, review status, treatment, program year, and project end date.

2.6 The user shall be able to enter a WHERE clause of the SQL query in the Find and the Filter window and run a query against the projects in the database. The query will be verified by the application and SQL errors will be reported to the user for correction. This capability shall be an application privilege. Bridge-level security of the user shall be applied automatically.

2.7 The user shall be able to select a project by directly typing in the project ID in the GUI, with the display scrolling to that structure. Bridge or project level security of the user shall be applied automatically.

2.8 The user shall be able to select projects (work) on a supported map display and generate a project/bridge list for use within Pontis, for example, to display results of a simulation analysis or to generate a STIP display.

UC-3 Generate Predefined Reports

3.1 The user shall be able to select and execute predefined reports to view bridge data, project data, preservation model data, program simulation results, and configuration data. The existing Pontis reports shall form the basis for the predefined reports to be provided in Pontis 5.0.

3.2 The user shall be able to create reports in a .NET compliant report generator other than InfoMaker and review these reports from within the Pontis GUI.

3.3 The system shall provide the ability to integrate current selections with a .NET compliant report generator other than InfoMaker. AASHTO may provide users with a list of suggested report generators to replace PowerBuilder/InfoMaker.

3.4 The system shall provide the ability to incorporate PDF files into standard Pontis reports.

3.5 The system shall provide the ability to save reports in PDF, HTML, and XML formats.

UC-4 Select and View Pontis Information Using Maps

4.1 The system shall provide the ability to create any report that is bridge or project based by selecting the relevant "driver" records from a map display.

4.2 The system shall provide the ability to create or refresh a map display from a set of bridge or project records selected within the Pontis desktop.

4.2 BRIDGE INVENTORY & INSPECTION

UC-5 Create/Edit Structure

5.1 The user shall be able to create a new structure with a user-defined bridge key.

5.2 The user shall be able to create a new structure by copying all data of an existing structure.

5.3 The user shall be able to create new structure(s) by importing from a supported file type. The supported file types are to be determined, but will include at least PDI, NBI, and XML.

5.4 The user shall be able to edit and remove existing structures.

5.5 The system shall provide the ability to mark a bridge with any of several status values (e.g. inactive/closed) and automatically filter displays to recognize the status value (e.g. to automatically suppress inactive/closed structures).

5.6 The system shall support storage and management of information for bridges in a design or preconstruction state (functional requirement is related to 4.5 above)

5.7 The user shall be able to renumber the bridge key identifier and have the change made globally. This functionality shall be restricted depending on the user privilege.

UC-6 Create/Edit Inspection

6.1 The user shall be able to create a new inspection.

6.2 The user shall be able to create a new inspection by copying a previous inspection.

6.3 The user shall be able to identify and select any previous inspection as the source for the copy action. The selection can be made by combination of date, inspection type, inspector, or inspection control group.

6.4 The system shall have a short-form interface (display only NBI and element data, similar to SI&A sheet) for inspection data entry with specifics of design and functionality to be determined as part of the technical design.

6.5 The user shall have the option to copy the notes from previous inspection when creating a new inspection.

6.6 The user shall be able to create new inspection(s) by importing an PDI file.

6.7 The user shall be able to add new elements to the structure.

6.8 The user shall be able to edit all the NBI data

6.9 Changes to calculated results such as Sufficiency Rating or NBI Appraisal Ratings will be prohibited except for secured, privileged users.[12]

6.10 The user shall be able to edit element inspection data.

6.11 The user shall be able to edit the bridge inventory items excluding the dependent/derived values (i.e., the appraisal ratings that are calculated automatically).

6.12 The user shall be able to edit inspection and structure notes.

[12] These derived values are only calculated by the FHWA or State implementation of the Edit Update program and are not user-modified.

6.13 The system shall be capable of storing notes in plain text and XML, and shall maximize the notes field size. The user shall be capable of setting the size of the notes field.

6.14 The user shall be able to create and edit work recommendations. The system shall be capable of estimating quantity for bridge type work recommendations.

6.15 The user shall be able to enter and update inspection planning information (i.e., frequency of regular and special inspections and estimated resource requirements).

6.16 The user shall be able to link and manage multimedia files.

6.17 The system shall provide an inspection scheduling tool.[13]

6.18 The system shall provide the capability to "lock down" an inspection record once signed and sealed to prevent deletion or modification.

6.19 Privileged users shall be able to delete an existing inspection unless it is "locked down".

6.20 The user shall be able to sort the element data by unit number, environment, and quantity.

6.21 The user shall be able to view all data from any two inspections, for a single bridge, in a tiled[14] presentation for easy comparison. The presentation (view/report) shall clearly highlight the difference between two inspections.

6.22 The system shall provide a configurable tracking mechanism to ensure that a set of screens and data fields have been reviewed and approved during the course of an inspection

UC-7 Calculate Dependent/Derived Inspection Results

7.1 The system shall be capable of calculating NBI condition ratings using the FHWA NBI Translator.

7.2 The system shall be capable of calculating and storing bridge Sufficiency Rating.

7.3 The system shall be capable of calculating appraisal ratings.

7.4 The system shall be capable of calculating Structurally Deficient/Functionally Obsolete (SD/FO) status.

[13] Range of functionality is still to be determined. This requirement depends on consensus on the scheduling capability.

[14] Side-by-side or top-bottom review panels

7.5 The ability of the user to update the calculated ratings and status in the database shall be a controlled privilege. Agencies may optionally exclude/prohibit direct edits to the calculated inspection results such as the NBI translator output. The agency shall therefore be able to enforce strict use of calculated translator results for calculating NBI condition ratings.

7.6 The system shall be capable of prompting users when the dependent fields for ratings are changed and offer the user option to recalculate the ratings.

7.7 The system administrator shall be able to configure the recalculation behavior of the derived data. The options shall include 1) do nothing, 2) alert user that recalculation is required and give option to recalculate before update/save, and 3) recalculate silently with no confirmation prompt when user saves the inspection.

7.8 The system shall eliminate any and all dependencies on workstation (user) control/parameter files for program operation. For example, parameter information required by the NBI Translator (including information currently stored in ELEMENTS.PRN and ERRORS.LST) will be stored strictly within the Pontis database rather than on individual user workstations.

4.3 PRESERVATION MODEL DEVELOPMENT[15]

UC-7 Develop Preservation Policy

7.1 The system shall be capable of creating cost and deterioration elicitation records for Pontis users.

7.2 The system shall be capable of updating the transition probabilities for preservation actions based on expert elicitations.

7.3 The system shall be capable of updating preservation action costs based on expert elicitations.

7.4 The system shall be capable of updating transition probabilities for do-nothing actions based on historical inspection data.

7.5 The user shall be able to edit the cost and deterioration models through the GUI and immediately see the impact on model recommendations.

7.6 The system shall be capable of using the transition probabilities and preservation action costs to develop an optimal preservation policy as detailed in the Pontis Technical Manual.

[15] Please refer to section 4.3 of the Pontis Technical Manual for more information on the preservation model in Pontis.

7.7 The system shall be capable of recovering the model if the elicitation process fails.

7.8 The system shall be capable of restoring the previous action costs, transition probabilities, and preservation policy.

7.9 The system shall be capable of supporting multiple cost and deterioration models.

7.10 The system shall be capable of incorporating Health Index targets in a preservation (short-term) model.

7.11 The system shall streamline the model generation process (i.e. remove any unnecessary prompts as currently implemented in version 4.x).

UC-8 Perform Health Index Targeting

8.1 The system shall be capable of determining the long-term cost of achieving an average health index target for one or more structure(s), broken down by elements, element categories, for the entire structure.

UC-9 Provide Preservation Model Dashboard

9.1 The system shall provide a preservation model dashboard integrating the capabilities described above. The exact functionality of the dashboard will be informed by review of the results of NCHRP Project 12-67, Multiple-Objective Optimization for Bridge Management Systems.

4.4 PROGRAM SIMULATION[16]

UC-10 Configure Simulation Parameters

10.1 The user shall be able to specify the characteristics of the simulation, including the simulation timeframe, various thresholds for the simulation, the project types included in the simulation, the needs addressed, and the annual budget for the simulation.

10.2 The user shall be able to update the unit costs and the policy standards governing the improvement model.

10.3 The user shall be able to modify the technical parameters governing the simulation, including the parameters of the improvement model and the simulation rules.

[16] Please refer to section 4.5 of the Pontis Technical Manual for more information on the program simulation in Pontis.

10.4 The user shall be able to select the structural elements to be included in the simulation.

10.5 The user shall be able to dynamically determine bridge subsets as part of the scenario settings.

UC-11 Perform Program Simulation

11.1 The system shall be capable of running a program simulation for a selected set of structures and storing the simulation results in the Pontis database.

UC-12 Perform Bridge Analysis

12.1 The user shall be able to choose from Pontis-generated work recommendations, inspector work candidates, and user-defined work items to include in the bridge simulation for a structure.

12.2 The system shall be capable of running bridge analysis for a single structure and storing the simulation results in the Pontis database.

12.3 The system shall provide a Bridge Analysis Dashboard that will allow the user to choose work items for a structure and the timing of the work, view the impact of the work on the condition of the structure, assign the work candidates to a project, and perform life-cycle cost analysis for the structure. The exact functionality of the dashboard will be informed by review of the results of NCHRP Project 12-67, Multiple-Objective Optimization for Bridge Management Systems.

4.5 PROJECT & PROGRAM DEVELOPMENT

UC-13 Create/Edit Programs

13.1 The user shall be able to create new programs by manually entering program information.

13.2 The user shall be able to edit or delete existing programs.

UC-14 Create/Edit Projects

14.1 The user shall be able to create new projects by manually entering the project information.

14.2 The user shall be able to edit or delete existing projects.

14.3 The user shall be able assign Pontis-generated work recommendations, inspector work candidates, and user-specified work items to a project, or remove work items from a project.

14.4 The user shall be able to split a project into multiple projects and combine multiple projects into a master project.

14.5 The user shall be able to edit the characteristics of multiple projects simultaneously (batch update).

14.6 The user shall be able to edit the data stored in the Project Planning tables through the GUI.

14.7 The system shall be capable of tracking and linking funding levels and project budgets. The system shall warn users if the project budget exceeds funding.

14.8 The system shall be capable of grouping work candidates by different action types – bridge-level, element-level, and flexible-level actions.

4.6 DATA MANAGEMENT

UC-15 Perform Data Validation

15.1 The system shall be capable of performing single-field and cross-field validation on NBI data items in batch mode.

15.2 The system shall be capable of performing configurable input range checks using the DATADICT table upon entering data.

15.3 The system shall be capable of performing data validation tasks including cross field validation upon saving data.

15.4 The system shall allow users to configure data validation rules. This shall be a privileged operation.

15.5 The system shall provide data review wizard for comparing incoming bridge data with existing data. The errors will be highlighted for easy review.

15.6 The system shall guarantee atomic database transactions to ensure maximum data integrity, using standard .NET transaction management capabilities.

15.7 The system shall maximize multi-user concurrency and conflict resolution in the application using standard .NET session and transaction management capabilities.

UC-16 Exchange Data

16.1 The user shall be able to exchange data in NBI format.

16.2 The user shall be able to exchange data in PDI format as specified in the Pontis Technical Manual.

16.3 The user shall be able to exchange data in XML format. The emerging TransXML schema specification[17] shall be utilized once established and accepted.

16.4 The user shall be able to exchange bridge inspection data in PDI format.

16.5 The user shall be able to designate a recipient for checked-out bridges.

16.6 The system shall store all export/import control information within the database for central administration, eliminating any dependency on workstation control files to manage export/import behavior.[18]

16.7 The system shall strictly separate transient or volatile batch operation control records – e.g. bridge records to be included in a report – from more static control record – e.g. currently checked-out bridges.[19]

UC-17 Perform Data Archiving

17.1 The user shall be able to archive data for existing and removed structures.

17.2 A selection/filter capability shall be provided to restrict operations on active, inactive, or all structures during any data archiving activity.

17.3 The system shall provide reporting mechanism for viewing archived data.

17.4 The user shall be able to restore the archived data.

17.5 The system shall use timestamps in the database to log when rows were last changed.

UC-18 BRIDGEWare Integration Support

18.1 The system shall continue to provide existing application integration capabilities with BRIDGEWare products.

18.2 The system shall support concurrent storage in the database of planned, active, and inactive structures, and potentially other status/lifecycle phase indicators, to facilitate BRIDGEWare integration.

[17] National Cooperative Highway Research Program (NCHRP) Project 20-64, XML Schemas for Exchange of Transportation Data

[18] For example, the existing export control files (NONBRDG.TXT, BRIDGES.TXT) determine the list of tables to be included in PDI export.

[19] The currently overloaded CICOCNTL table will have only the role of recording checked-out and checked-in bridges in Pontis 5.0. A separate batch management mechanism will be used to manage bridge subsets for reporting, program simulation, and other volatile batch operations.

4.7 SYSTEM ADMINISTRATION

UC-19 Define and Manage User Roles

19.1 The administrator shall be able to create and edit new application roles.

19.2 The administrator shall be able to assign users to application roles.

19.3 The system shall provide user interface for managing application roles.

19.4 The administrator shall be able to assign application permissions to each user roles.

UC-20 Manage Application Users

20.1 The administrator shall be able to add new users to the system.

20.2 The user shall be able to modify his or her basic account profile.

20.3 The administrator shall be able to create bridge-level access filters.

20.4 The administrator shall be able to assign bridge-level filters for each user.

20.5 The administrator shall be able to remove users from the system or disable access as appropriate.

UC-21 Perform User Authentication

21.1 The system shall be capable of validating user name and password during log-in process.

21.2 The system shall be capable of authorizing access to Pontis functionalities depending on the user's privileges during log-in process.

21.3 The system shall be capable of tracking login and logout activity.

21.4 The system shall be capable of running concurrent sessions for a user.

21.5 The system shall be capable of supporting Windows Authentication mechanism for database access.[20]

21.6 The system shall be capable of initiating a session without a login prompt. This capability will provide support for an independent launch shell for Pontis and will be configured by the system administrator.[21]

[20] The Windows Authentication capability is database vendor specific and only relevant to the client/server application components. Pontis 4.4 only supports this for Microsoft SQL Server 2000.

[21] This is similar to Florida DOT's application launcher and consistent with the Single Sign On capabilities currently used by many web portals.

21.7 The system shall be capable of integrating authentication and permissions with BRIDGEWare[22].

21.8 The system shall be capable of operating with a single-sign-on authentication mechanism based on Active Directory Services or the LDAP standard.[23]

UC-22 Configure Application Functionalities

22.1 The administrator shall be able to configure the system units of measure in either English or metric mode. This will be a permanent setting for the application and database.[24]

22.2 The system shall be capable of storing the INI file configuration information in the database or workstation registry as appropriate, eliminating dependency on INI files for program operation.

22.3 The system will accommodate XP directory security and file permissions when storing any configuration or session parameters.

22.4 The administrator shall be able to edit the DATADICT table or its successor(s) through a GUI window.

22.5 The administrator shall be able to add and configure pre-populated new structure templates.

22.6 The administrator shall be able to configure field-level security.

22.7 The administrator shall be able to configure some fields as mandatory for user input.

22.8 The administrator shall be able to configure different levels of permission to control editing of checked-out bridges.

22.9 The system shall adhere to a .NET logging standard.

22.10 The system shall adhere to a .NET exception handling standard.

22.11 The administrator shall be able to configure the values in the dropdown lists in the system.

[22] Note that this requirement implies that corresponding development work will be performed for Virtis/Opis. Where feasible, costs should be minimized by utilizing existing functionality.

[23] It should be recognized that the Virtis/Opis application authentication/security control approach may need to be revised to accomplish this capability globally in the AASHTO BRIDGEWare software suite.

[24] Data exchange operations will still support conversion, and the NBI tape will continue to be created in metric format only subject to NBIS reporting standards change at the Federal level.

22.12 The administrator shall be able to create agency-specific elements, including element definitions, condition states, actions, models and other related information.

22.13 The administrator shall be able to configure definitions of element types, environments, materials, and categories.

22.14 The administrator shall be able to add custom structure lists.

22.15 The administrator shall be able to add custom project lists.

22.16 The administrator shall be able to configure default values for various scenario parameters.

22.17 The administrator shall be able to configure customizable options that affect system operations.

22.18 The system shall validate all configuration options to verify proper program operation, including explicit identification of any potentially conflicting program options.

22.19 Configuration option validation shall be supported by a built-in diagnostic report.

UC-23 Configure User-Interface Presentation

23.1 The administrator shall be able to configure the labels of user interface items and screen displays.

23.2 The system shall be capable of supporting internationalization using the standard Windows/.NET internationalization support.

23.3 The system shall be capable of showing all screens, menus, user interface controls, prompts, error messages and information notices in native language, with some of these elements customizable.

UC-24 Admin Functionalities

24.1 The system shall provide an interface to communicate with an external service or third-party software. A set of standard interfaces[25] will be defined for this capability.

24.2 The system shall provide application administrators with a secured tool for overriding bridge/project check-out status.

24.3 The system shall provide application administrators with a secured tool for clearing batch processing tables used by the system for deadlock/lockout resolution.

[25] Web service and SOAP (Simple Object Access Protocol) will likely be used for these interfaces

24.4	The system shall provide a fixed set of predefined, secured, administration reports to document users, user roles, application privileges, user login sessions, checked-out bridges, and other key information necessary to manage application operation.

4.8 DATA REQUIREMENTS

A full, detailed data model for Pontis 5.0 will become part of the Pontis 5.0 System Design and is beyond the scope of this document. The Pontis 4.x/BRIDGEWare database design is assumed as the basis for these requirements except as indicated. The following is a list of basic table groupings as defined in the Pontis Technical Manual.

- Physical Inventory: This set of tables contains data related to an agency's physical bridge inventory. These include general inventory data, work candidates recommendations, structures exchanged during the check-in/check-out process, multimedia document links, and optional agency-specific inventory data.
- Program Simulation: These tables store data related to the program simulation process. They contain elicitation data, deterioration model updating data, preservation policy, flexible action definitions, program simulation inputs, and simulation results.
- Project Planning: These tables contain information about programs and funding sources, projects and their work items, agency-defined project data, and temporarily stored work items used to support the bridge analysis functionality.
- Definition: Pontis system definitions are stored in these tables.
- Application System: Application system tables store configuration data required for running Pontis. The tables contain system data including the data dictionary, parameter, configurable options, data on users and their roles, formulas, and other system data. It is likely that extensive changes will be made in these tables for Pontis 5.0, particularly to support auditing, logging, versioning, user authentication and session management, and other new capabilities.

5.0 Non-Functional Requirements

5.1 Look and Feel Requirements

The EIM&DSS design will be informed by World Wide Web Consortium (W3C) design guidelines[26] and conform to common industry practices for user interaction behavior and user interface design. For the thin-client portion of Pontis, the AASHTO standards for thin-client web-based applications software will be followed.

The software, through its user interface, will clearly identify the product name (Pontis 5.0.x), the licensee, the connected user, and the organization that developed the system (AASHTO). Additional session identification information will be available on demand.[27] The overall look and feel of the product will be that of a professional and conservative application.

Pontis 5.0 will, at AASHTO's direction, adopt any elements of the look and feel of other existing AASHTOWare/BRIDGEWare applications where appropriate. In particular, UI conventions from Virtis/Opis will be reviewed to ensure that the Pontis 5.0 does not differ markedly in appearance. AASHTO may designate such model applications and must provide the development team with access to fully-documented style sheets/UI layouts or demonstration software to support this requirement.

There will be no support for individual personalization of the user interface (skins) other than the availability of certain features that are tied to a user role (e.g. not all users will have the ability to edit bridge data and thus will not see an "Edit On" button on the corresponding screen).

5.2 Usability Requirements

Pontis 5.0 will be designed to be as straightforward and intuitive as possible, and familiar to existing users. The user interface will provide effective navigation, visual referencing, and task sequencing. Users will have access to on-line help throughout the UI.

The application should look and feel consistent in the web and standalone operating environments, therefore it will conform to standard .NET standalone

[26] Word Wide Web Consortium, Web Content Accessibility Guidelines 2.0, November 2004 (http://www.w3.org/TR/WCAG20/)..

[27] Pontis 4.x Help About and SysInfo windows provide examples.

and web application conventions with necessary accommodations for the deployment environment only where these prove necessary and appropriate.

As a general goal, Pontis users should be comfortable operating and using the software after two full days of training. Users who only wish to access and display bridge information should be comfortable operating and using the software after two hours of training. System administrators should be comfortable operating and maintaining the software after user training and one full day of system administration training.

The system developer shall consider supporting existing custom forms and reports developed in PowerBuilder 9. It shall decide on a development tool for users so that they can continue to customize Pontis and extend its capability. For example, custom forms and reports created with PowerBuilder 9 may be migrated to a .NET application by using the Sybase DataWindow .NET technology which is available in PowerBuilder 10.

Integrating DataWindow .NET is likely a short-term strategy to ease migration of user forms and reports to Pontis 5.0. A stated objective for Pontis 5.0 is eliminating dependency on PowerBuilder. Ideally, the application should not require any particular vendor technology for report or form generation, such as InfoMaker/PowerBuilder, so a sunset plan must be developed to outline a strategy to meet this objective. However, the application shall continue to provide a plug-in capability to integrate third-party .NET data displays and reports in the user workspace.[28]

5.3 PERFORMANCE REQUIREMENTS

This section describes general performance requirements for Pontis 5.0, including its capacity (the volume of all types of data that can be stored in the repository) and its speed (the time it takes to execute certain types of operations).

Some factors that determine the ultimate performance of Pontis are outside the control of this project, and rely instead on decisions and actions of each agency that implements Pontis:

- The capacity and speed of the commercial RDBMS product used for the data repository;
- The speed of the servers on which Pontis, the RDBMS, the analytical and other processing engines, and other information resources are installed;

[28] Plug-in reports and forms may continue to be based on PowerBuilder at the user's discretion. The plug-in must be capable of handling this technology for the foreseeable future, but not solely dependent on it.

- The effective speed and congestion levels on the network that connects the servers and client machines running the application; and
- A standalone database engine will continue to be necessary. Performance objectives for the standalone database should be comparable.

Some of these factors may make it impossible to achieve the performance requirements for Pontis 5.0 outlined in this section. However, based on our assessment of the existing condition of most agency systems and infrastructure, we think this outcome is unlikely.

There are selected licensees, typically smaller agencies, who may have much lower powered computer resources, and their performance must be acceptable as well, although not necessarily comparable to larger installations.

Capacity

The Pontis 5.0 repository must be able to store the following quantities of bridges, work recommendations, scenarios, projects, and users at a minimum:

Entity	**Capacity of Database**	**Comments**
Bridges	50,000	This number is approached by some licensees already.
Inspections per bridge	30	
Work Recommendations	50 per bridge average	Most bridges will have no recommendations at all, or a small number of them(20 or fewer).
Scenarios	20	The number of scenarios that can be stored and displayed in Pontis concurrently.
Projects	50,000	The number of projects that can be stored and displayed in Pontis concurrently.
Project detail	150,000	A concomitant number of work items for each project ay be expected, perhaps 15 per project
Registered Users	500	The number of users accessing key functions – running simulations, updating models, etc. – is expected to be fewer than 5 on a routine basis.
Simultaneous	25 (web), 10	These are transactional read-write users. The purpose of this number is to

Read-Write Users	(standalone)	provide a design guideline for hardware specifications to be developed later on in this project. Pontis will not impose any particular limitation on the number of simultaneous users. Agency web servers may restrict the number of simultaneous web sessions.
Simultaneous View-only Users	150	Report and data browsing users, for all deployment environments
Batch (offline processing)	2	Users that will create the reports at off peak hours and post the results to the website. This could be higher if there is a time issues with the number of long running reports being generated.

Speed

This section presents speed targets for specific system functions. The apparent speed of many operations will be affected by network latency (delays due to transmission of text and images from the Pontis application server to the client desktop machine and web browser, or for standalone operations). The times indicated in the table represent user-experienced response times.

The target times indicated in the table represent 90% thresholds – that is, the performance requirement specified here means that the indicated functions will take less than the indicated length of time an average of 9 out of 10 times.

Operation	Target Time (seconds)	Comments
Login/Logout	2	Depends primarily on database and network performance
Redisplay typical desktop (on startup, zoom, reselect, or other change), assume 250 bridges	5-10	This value applies for displaying the Pontis desktop selection (assuming presentation similar to Pontis 4.x)[29]
Create a new bridge record	3	For example, creating a new inspection or a new structure.

[29] Localization has had a demonstrable impact on GUI performance. Pontis 5.0 may rework this to improve screen display speed and will address localization properly as an architectural issue/constraint.

Export **50** bridges	30	About 2 bridges per second, varies with type of export file and scope of exported information.
Import **50** bridges	30	About 2 bridges per second, likely slower than export, varies with type of import file and scope of imported information.
Retrieve and display single bridge (inspection) detail	< 2	Will take longer if digital photographs, plan data, or other no-native content are being accessed.
Retrieve and display single project detail	< 2	
Validate 50 bridges for NBI edit check	10	Better than 3 bridges per second.
Generate a formatted report	10-20	Will vary substantially by report type.
Save a report data to disk	5	Will vary by report type and data volume
Save/update miscellaneous updates, configuration and model parameters	2	
Run simulation for 100 bridges	< 30	Will vary by simulation time frame
Update cost/deterioration models	<= ½ hour	Will vary by number of bridges and whether history is used.

Reliability and Availability

Pontis 5.0 is not strictly a transactional application or mission-critical for most capabilities. In light of this, the application shall be available for use 18 hours per day, 353 days per year, reserving 6 hours per day and one day per month for routine agency network and infrastructure maintenance. During normal times of operation, Pontis shall achieve 98 percent up time.

Because Pontis 5.0 is not a mission-critical application, there will be no attempt to design or construct the system for uninterrupted operation. Agencies may choose to implement database transaction logging, mirroring, or switchover to backup servers in the event of a hardware or network failure, but support for these decisions or implementation are beyond the scope of Pontis 5.0 effort. Procedures and equipment for backups (database and software) and disaster recovery are also outside the scope of this effort.

5.4 Operational Requirements

This section outlines the minimal requirements for effectively operating Pontis 5.0. The requirements for both thin-client and thick-client applications are discussed. Specific items are to be determined.

The Pontis application server (ASP.NET) software will run on a server located within a (presumably) secure facility at each agency. Because it is an ASP.NET application, the end users can vary in location depending on the network accessibility of the server. The Pontis database, and any other databases to be accessed from Pontis may be co-located on the same server machine, but at a minimum must be accessible from the Pontis application server via standard database communication and connection protocols.

The server machine will run Windows Server 2003 or, given technology evolution, a descendant target server environment identified in the implementation plan. The server machine must have all necessary software to run an ASP.NET application. This specifically includes, but may not be limited to, IIS[30], the .NET Framework and ASP.NET. Other web servers, appliances, and add-in technologies may also be necessary depending on the final application design.

Back-end server stored procedures and server-side functions will be supported and may be assumed to be usable in the technical design of Pontis 5.0. This will require implementation of the triggers in native DBMS procedural code. This implies that it is very likely that the server-side code will vary for each target DBMS, even if the interfaces remain identical, increasing the development and testing effort with the significant benefit of assigning processing to the appropriate application tier(s).

Pontis 5.0 will be designed to be used with Internet Explorer; it may work correctly with other browsers but this cannot be guaranteed. Given the predominance of IE in the browser market, it does not appear to be necessary to consider support for other browsers, although this may need to be reconsidered if external customers are running the application(s) and need to employ different browser technology such as Netscape or Mozilla FireFox.

Standalone capabilities will require a workstation running the latest .NET Framework, presumably Windows XP Professional (SP2) or its successor. Some capabilities may require a network connection. The workstation's hardware requirements will be similar to those of the existing Pontis 4.4 application. There

[30] The TAG should strongly consider the need to support HTTP/www servers other than IIS. Apache and its variants are the most widely-used web servers on the net and a rich platform to exploit open source capabilities. However, many agencies rely on IIS, so this technology consideration must be resolved as part of the design. Other AASHTOWare products (Trans*port) already vary from the IIS standard.

will continue to be a standalone/runtime database engine provided with the application for offline work. The offline database is assumed to be Sybase Adaptive Server Anywhere release 9 or above.

5.5 Installation and Deployment

For a thick-client portion of Pontis, an installer program will be provided to install Pontis on the user's computer. The user shall be able to remove Pontis by using the Windows Add/Remove Programs capability. When necessary, the SQL scripts will be provided to upgrade a previous version of Pontis database to a new version and apply any patches. The system may be capable of supporting automatic upgrades and maintenance fixes using the push technology.

Pontis 5.0 shall be able to coexist with earlier versions of Pontis on the same computer.

5.6 Maintainability and Portability Requirements

Pontis 5.0 shall comply with AASHTO application development standards based on the published recommendations of the AASHTO Technical Applications and Architecture (TA&A) subcommittee.

The latest released versions of Microsoft tools including Visual Studio .NET, ASP.NET, .NET Framework, and databases such as Microsoft SQL Server will be used. The actual versions, the database release version in particular, will be determined as part of the implementation plan and put under configuration management.

All source code written by the contractor will be provided to AASHTO, and reasonable effort will be made to develop clearly documented source code which can be maintained after contract by other than the original developers.

All modules developed in Pontis 5.0 will support automatic testing except as explicitly distinguished in the technical design. Every module will be delivered with a testing driver module.

No commitment regarding the portability of the code is made other than that implied by the use of standard Microsoft .NET tools and well-documented third-party application libraries or frameworks.

Pontis 5.0 in production is intended to be supported through complete and sufficient on-line help and documentation and equivalent printed documentation, with occasional additional assistance provided through the Pontis Support Center. Pontis 5.0 will also continue to provide field-level help where applicable. As with the existing Pontis 4.x product, it is assumed that support will be provided by telephone, via e-mail, and through an on-line issues tracking system.

5.7 Security Requirements

Pontis 5.0 security is a mix of database security and application security. Overall security is provided by the database and the Pontis 5.0 user authentication mechanisms with more refined security over program capabilities - e.g. create a bridge - being addressed by the application.

It is assumed that authentication will utilize a single sign-on (SSO) approach and application security will be declaratively managed through database or directory entries, rather than hard-wired into the application. This mechanism may have implications for BRIDGEWare/AASHTOWare generally and will need to be coordinated with the other AASHTO products[31].

Database security will need to be insured for the Pontis data stored in the RDBMS; most likely this will be done by using a specific username/password combination and running as particular database roles for each database connection. This security structure should be similar to the existing security approach in Pontis 4.x. For web clients, care will be taken to insure that connection information is not accessible from application pages (e.g., userid/password embedded in HTML form variables).

The application level security controls will be consistent with current Pontis 4.x procedures, except more comprehensive; no exceptional controls are required. A specific requirement addresses field-level data security.

For normal users, the configurability of the system via the interface will be limited to changing the parameters used to perform the various analyses and modeling functions. Access to all other configuration parameters, such as module access, will be restricted to Pontis super users or application administrators.

There are no specific privacy or immunity (antivirus) requirements for Pontis 5.0 identified at this time.

5.8 Cultural and Political Requirements

As there are no evident cultural or political application requirements for Pontis 5.0, this section of the Functional Requirements document does not apply for the most part. However, the product likely will be designed for agency/international localization from the ground up with respect to user interface elements, reports, messages, and other application attributes where native language support is necessary. This capability will be usable by all

[31] Single sign-on technology is an issue that SCOJD and TAA should address for all AASHTOWare products. It appears that Kerberos is the most common basis, and Microsoft uses Kerberos for its Active Directory Services. Kerberos is also well supported by database vendors.

licensing agencies as desired, on an optional basis, to customize the appearance of the product to suit their internal standards. This capability will be analogous to the localization capability of Pontis 4.4 but more comprehensive in scope.

5.9 LEGAL REQUIREMENTS

The system shall be capable of exporting the NBI data in the specified NBI format as outlined in the NBI guideline. The system shall be capable supporting the future NBI coding guide changes.

6.0 Implementation Plan

6.1 OVERVIEW

In order to develop an approximation of the total effort involved and create the development plan necessary for project planning, three implementation alternatives have been identified for consideration by the BRIDGEWare Task Force. These alternatives are quite different from each other, particularly in terms of schedule and user impacts. The alternatives are specified in terms of effort, schedule, and deliverables.

Development effort is estimated as outlined in section 6.2. The estimating approach provides a standardized mechanism for estimating the complexity of the effort involved in providing a component of Pontis 5.0, which can be transformed into development hours or cost. Based on the estimate of development effort, three alternative development approaches have been prepared. The schedules for Pontis 5.0 are intrinsically linked to the product's revenue cycles which are directly related to subscribership. Hence, the schedules shown for the alternatives reflect available monies and assume no solicitation effort will be undertaken.

The three alternatives proposed for consideration are:

1. **Dedicated Design and Release**, where the entire Pontis 5.0 product is designed, developed, tested, and delivered as a single multi-year effort, with an explicit cutover to the new version at a future point in time. There are no interim releases or initial products in this alternative;
2. **Phased Design and Release**, where components of Pontis 5.0 are designed, developed, tested, and delivered on a flow basis as a multi-year effort, with initial efforts focused on delivering a web based inspection module; and
3. **Dedicated Design and Phased Release**, which varies from the second alternative by explicitly calling out a dedicated design task that will start and conclude before incremental phased development of Pontis 5.0 components commences.

Each of these alternatives is discussed below, with a summary, an outline of driving assumptions, an outline of major development tasks, resources, and schedule, and a revenue analysis.

Note that each alternative assumes that the Pontis and Virtis/Opis contractors will coordinate throughout the design of Pontis 5.0 and that the Virtis/Opis contractor will be funded outside of the Pontis effort to review Pontis design documents.

6.2 Resource Requirements Analysis Approach

The functional size of Pontis 5.0 has been estimated using the Common Software Measurement International Consortium Full Function Points (COSMIC-FFP) method.[32] The COSMIC-FFP size scale is based exclusively on Functional User Requirements, defined as "the user practices and procedures that the software must perform to fulfill the users' needs." Functional user requirements exclude quality requirements and any technical requirements. The COSMIC-FFP method is designed to work for "data-rich" and "real-time" software, and it provides the concepts of software layers and tiers to help differentiate functional user requirements allocated at different levels of functional abstraction. More information on this methodology can be found in Appendix A.

The COSMIC-FFP method results in an estimate of Cosmic function size units (CFSU). This estimate is multiplied by an adjustment factor reflecting application complexity to obtain "Adjusted CFSU," and it may then be scaled as appropriate to yield an estimate in terms of hours or development cost. The cost estimate for Pontis is presented in Table 6.1. Costs were calibrated by comparing actual development costs to CFSU estimates for recent Pontis efforts. Note that the COSMIC-FFP method is not capable of providing an estimate of the cost of developing the basic application framework and C++ wrappers, so these items were estimated separately based on expert judgment.

Note that the estimates provided below are for the development costs only and do not include the cost of design or testing. Design costs are anticipated to be 25 percent of the development costs, with an additional $25,000 for each design document produced. Testing each release is expected to cost $55,000.

[32] A. Abran, J. Desharnais, S. Oligny, D. St-Pierre and C. Symons, *COSMIC-FFP Measurement Manual, version 2.2*, 2003.

Table 6.1. Estimated Resource Requirements for Pontis 5.0 Development

Module	CFSU	Complexity Multiplier	Adjusted CFSU	Estimated Development Cost
Inspection	268	1.5	402	$520,000
Project Planning	159	1.5	238.5	$310,000
Program Simulation	252	1	252	$330,000
Preservation	80	1	80	$105,000
Results	28	1	28	$35,000
Gateway	84	1	84	$110,000
Configuration	237	1	237	$310,000
Developing .NET wrappers for C++ DLL's	-	-	-	$100,000
Developing basic framework for Pontis 5.0	-	-	-	$250,000
Total	1108		1084.5	$2,070,000

6.3 ALTERNATIVE 1 – DEDICATED DESIGN AND RELEASE APPROACH

Overview

The dedicated design and release approach is a strategic option focused on delivering a complete product to end users at a single point in time, when a wholesale cutover will be performed by agencies to deploy Pontis 5.0. The key characteristics of this plan are fewer, larger tasks, a significant initial design effort, fewer interim deliverables, a dedicated testing phase, and a single release, resulting in an intense, focused deployment effort for licensing agencies.

Driving Assumptions

The driving assumptions behind this approach are:

1. Agencies do not want multiple deployment efforts in order to establish Pontis 5.0 in production, given costs in hardware, training, staff time, and implementation assistance;

2. A single large project with a complete deliverable package is more focused than a phased effort with numerous design, develop, and release cycles, allowing a dedicated, continuous development team effort;
3. Changes in technology will be relatively stable (e.g. the development framework will not change radically over time, and importantly the work done early in the development effort will not need to be revisited later, and possibly overhauled or significantly reworked, in response to a technology change[33]);
4. The entire design and development effort can be contracted out as a package with a fixed price;
5. The effort can be broken down by contract period to accommodate revenue cycles, if necessary; and
6. Pontis 4.x will continue to be maintained throughout the development period until all capabilities are delivered, and at least 2 interim releases may be needed to address interim functional needs in the user base.

Key Deliverables

The key deliverables for this approach are:

- **Prototypes:** Three software prototypes will be developed and demonstrated to the client as part of the software design.
- **Design Document:** The system requirements and software design decisions will be documented in a detailed design document.
- **Pontis 5.0 Alpha Release:** This version of the software will incorporate all the functionality outlined in the design document, and will be used for internal testing by CS developers.
- **Pontis 5.0 Beta Release:** This version of the software will be released to Pontis users for Beta testing.
- **Pontis 5.0 Final Release:** The final version of the software will be released to Pontis users no later than December 2009.

Schedule

The tasks and schedule for this alternative are presented in Appendix B. The cost estimate is summarized in Section 6.6.

[33] This stable technology assumption was a key factor in the Virtis/Opis decision to focus on a COM-based approach, and it has been proven correct until very recently when .NET has emerged to replace COM.

6.4 Alternative 2 – Phased Design and Release Approach

Overview

The phased design and release approach delivers significant components of Pontis 5.0 serially over time with rolling cycles of design, development, testing, and deployment, highlighted by an initial focus on developing the web inspection component. This approach features multiple, less elaborate releases, with more user feedback contributing to the design of later components. It stresses initial development of a stable framework for the product which persists through the several development cycles. There is no "massive" release, but there may be significant impacts from multiple database upgrades, functionality releases, or technology upgrades.

Driving Assumptions

The driving assumptions behind this approach are:

1. Agencies already need some of the projected components/capabilities of Pontis 5.0 earlier, particularly, the web inspection capabilities and want them expedited;
2. Agencies can accommodate serial, increment releases within their own IT planning schedules;
3. It is generally desirable to show results to the users earlier;
4. Phased development may be somewhat less technology dependent in that new technology capabilities (e.g. a new .NET Framework or web development paradigm) can be accommodated as the project progresses and releases components;
5. Smaller, more focused releases may offer additional flexibility and responsiveness to user feedback;
6. Pontis 5.0 components are discrete and may be broken out into modular releases;
7. There may be less risk in modular releases, in that a change of direction/emphasis/schedule in response to external factors may be more possible;
8. Design reviews are less complicated and more easily accommodated in busy management schedules;
9. Smaller releases may map better into product revenue and contracting cycles;
10. Pontis 4.x will continue to provide significant BMS functionality for users until Pontis 5.0 capabilities are delivered; and

11. Pontis 4.x will continue to be maintained throughout the development period until all capabilities are delivered, but no significant interim releases are anticipated.

Key Deliverables

The key deliverables for this approach are:

- **Design Documents:** The system requirements and software design decisions will be documented in three design documents – one for each version of Pontis.
- **Pontis 5.0 Release:** This version of Pontis will incorporate the core functionality of the software, as well as the functionality of the Inspection module, and it will be available no later than March 2007.
- **Pontis 5.1 Release:** This version of the software will incorporate the functionality of the Project Planning and Gateway modules, and it will be available no later than March 2008.
- **Pontis 5.2 Release:** This version of the software will incorporate the functionality of the Preservation, Programming, Configuration, and Results modules, and it will be available no later than April 2010.

Schedule

The tasks and schedule for this alternative are presented in Appendix B. The cost estimate is summarized in Section 6.6.

6.5 ALTERNATIVE 3 – DEDICATED DESIGN AND PHASED APPROACH

Overview

This alternative differs from Alternative 2 by emphasizing a dedicated design effort followed by phased development of Pontis 5.0 over time. The development does not start until the design is completed and approved. This alternative does emphasize significant prototyping efforts intended to ″prove out″ and demonstrate design concepts, which serve to communicate key design concepts and shape expectations about the future product. There are multiple release cycles in this alternative, similar to Alternative 2.

Driving Assumptions

The driving assumptions behind this approach are:

1. A dedicated design effort allows for adequate user input and feedback prior to development;

2. As with Alternative 2, phased development may be somewhat less technology dependent in that new technology capabilities (e.g. a new .NET Framework or web development paradigm) can be accommodated as the project progresses and releases components;
3. A phased, multiple release, development process can be accommodated by agencies, as assumed in Alternative 2;
4. Revenue cycles are a driving factor, and all the development work must fit into current projections;
5. Pontis 4.x will continue to provide significant BMS functionality for users until Pontis 5.0 capabilities are delivered; and
6. Pontis 4.x will continue to be maintained throughout the development period until all capabilities are delivered, but no significant interim releases are anticipated.

Key Deliverables

- **Prototypes:** Three software prototypes will be developed and demonstrated to the client as part of the software design.
- **Design Document:** The system requirements and software design decisions will be documented in a detailed design document .
- **Pontis 5.0 Release:** This version of Pontis will incorporate the core functionality of the software, as well as the functionality of the Inspection module, and it will be available no later than June 2007.
- **Pontis 5.1 Release:** This version of the software will incorporate the functionality of the Project Planning and Gateway modules, and it will be available no later than June 2008.
- **Pontis 5.2 Release:** This version of the software will incorporate the functionality of the Preservation, Programming, Configuration, and Results modules, and it will be available no later than June 2010.

Schedule

The tasks and schedule for this alternative are presented in Appendix B. The cost estimate is summarized in Section 6.6.

6.6 RECOMMENDED ALTERNATIVE

The expenditure by fiscal year for the three alternatives are presented in Table 6.2, and the delivery dates for each module are presented in Table 6.3.

Table 6.2 Annual Expenditure By Development Approach

Approach	Annual Expenditure ('000s)					
	FY05-06	FY06-07	FY07-08	FY08-09	FY09-10	Total
Dedicated Design and Release	$550	$597	$655	$653	$220	$2,675
Phased Design and Release	$592	$809	$585	$500	$341	$2,828
Dedicated Design and Phased Release	$550	$925	$475	$468	$367	$2,785

Table 6.3 Deliverable Dates By Development Approach

Deliverables	Dedicated Design and Release	Phased Design and Release	Dedicated Design and Phased Release
Prototypes	Jun-06	NA	Jun-06
Core Functionality	Dec-09	Mar-07	Jun-07
Inspection Module	Dec-09	Mar-07	Jun-07
Project Planning Module	Dec-09	Mar-08	Jun-08
Gateway Module	Dec-09	Mar-08	Jun-08
Program Simulation Module	Dec-09	Apr-10	Jun-10
Preservation Module	Dec-09	Apr-10	Jun-10
Results Module	Dec-09	Apr-10	Jun-10

We recommend that AASHTO proceed with Alternative 3 (dedicated design and phased release). This project development strategy aligns well with available revenue cycles, allows a formal design review and adjustments, makes significant Pontis 5.0 components such as a web inspection module available relatively quickly after design is confirmed and approved. Further, this alternative provides positive feedback on user expectations, and allows adjustments to incorporate technology evolution during the development period.

6.7 Risk Analysis

The following are the major risks associated with the Pontis 5.0 development effort:

- **Requirement Creep:** There is a risk that system requirements will grow significantly beyond those conceived while developing the development cost estimate. This risk can be managed by developing a requirements document prior to starting the development effort.
- **Obsolescence:** There is a risk that the .NET standard will become obsolete during the product development cycle, or very soon after the new version of Pontis is released. This risk can be mitigated by adopting a phased approach to the design and/or the development effort.
- **Development Cost:** There is a risk that the actual cost of the development effort will significantly exceed the estimates.
- **Development Schedule:** There is a risk that the development effort will take significantly longer than scheduled. This may affect the license base if states decide to develop their own system or move to another system.
- **Resource Risk:** There is a risk that the developer will not have sufficient resources to complete the development, either because of staff turnover or because of a lack of technical expertise.
- **Changes to the NBI:** There is a risk that the changes to the NBI planned by the FHWA will significantly affect the development schedule or increase the cost of the development effort.
- **Agency Custom Forms:** There is a risk that the approach taken to providing agency forms will not support easy migration from Pontis 4.x to Pontis 5.0.
- **Maintaining Pontis 4.x:** There is a risk that the cost to maintain Pontis 4.x will exceed budgeted amounts. This could happen if there are changes to PowerBuilder or Sybase that require major revisions to Pontis 4.x during the Pontis 5.x development cycle. This risk can be partially mitigated through a fixed price maintenance approach.
- **User Satisfaction:** There is a risk that the final product may not meet user expectations, either because it does not meet their requirements or because it does not conform to their technology standards. This risk can be managed through the use of prototypes in the design phase and by following a phased development schedule.

7.0 Project Issues

7.1 Open Issues

There are no open issues as of this FRS version.

7.2 Off-the-Shelf Solutions

No review has been performed and consequently no commercial off-the-shelf solutions (COTS) have been identified as yet to fulfill special requirements in this document. Specific functionality in Pontis 5.0 may be provided by off-the-shelf solutions such as Sybase DataWindow .NET. COTS software may be in the public domain or licensed for a fee by AASHTO. Further COTS review will be conducted as part of the detailed design phase.

A standalone database engine will be required for off-line/standalone operation. It is assumed this will be the Sybase Adaptive Server Anywhere product, release 9 or above.

7.3 New Problems

No new problems have been identified as of this FRS version.

7.4 Waiting Room

As of Revision 0.3, the "may" requirements from Section 4.0 - Functional Requirements - have been moved to this section.

UC-1 View Bridge Data

The system may be configurable by the application administrator to provide additional standard criteria for filtering, finding, and selecting structures. (possible issues when upgrading the software-updating custom configurations)

The system may provide a two-way interface with a GIS map application to display bridge and project records on a map.

UC-2 View Project Data

The system may be configurable by the application administrator to provide additional standard criteria for filtering, finding, and selecting projects/work items.

Database enhancement: Federal aid common code number field may be added to the PROJECTS table.

UC-4 Create/Edit Structure

The user may be able to create a new structure using a step-by-step wizard with pre-defined data requirements.

The user may be able to create a new structure using pre-defined templates.

UC- 5 Create/Edit Inspection

The user may be able to create a new inspection using a step-by-step wizard.

The user may be able to query on the notes field from the GUI.

The system may provide the capability to affix an electronic signature to any record in the system to "lock down" the record.[34]

The system may be capable of transferring multimedia files to and from a server to a local hard drive.

The system may provide the capability to communicate/link with an EDMS (electronic document management system).

The system may provide the ability to combine multiple inspections.

UC-7 Develop Preservation Policy

The system may be capable of incorporating user cost considerations in the short-term and long-term model for deck elements.

UC-8 Health Index Targeting

The system may be capable of running parametric simulations (i.e. series of simulations with varying budget levels).

The system may be capable of running multiple scenarios to generate results sets used in cross-scenario reporting.

UC-13 Create/Edit Programs

The user may be able to create new programs using a program wizard.

UC-14 Create/Edit Projects

The user may be able to create new projects using a project wizard.

[34] This requirements document does not attempt to address the issue of state law and agency policy with regard to electronic signatures by licensed professionals. From the application point of view, the capability will be generic and follow available open source standards for affixing electronic signatures.

The system may provide an interface for route management and/or corridor analysis. Projects may be defined by geography, mile point limits, and other criteria.

UC-16 Exchange Data

The system may provide capability to translate contract bid items to cost data used in the preservation cost model.

The system may provide a specification for exchanging data with a Maintenance Management System, which will likely involve development of an XML schema for this purpose.

UC-17 Perform Data Archiving

The system may provide field-level change logging.

UC-20 Create/Edit Users

The user may be able to send a request for a new user account. This requirement applies to the thin-client platform.

UC-22 Configure Application Functionalities

The administrator may be able to configure the search and filter criteria for viewing data.

The administrator may be able to configure the level of GUI and application disk logging.

UC-24 Admin Functionalities

The system may provide messaging capability to notify users when a key event (such as creating new bridges or deleting bridges) occurs.

Appendix A – COSMIC-FFP Overview

In the COSMIC-FFP method, the software to be measured is broken down into "functional process types," defined as "an elementary component of a set of Functional User Requirements (FUR) comprising a unique cohesive and independently executable set of data movement types."

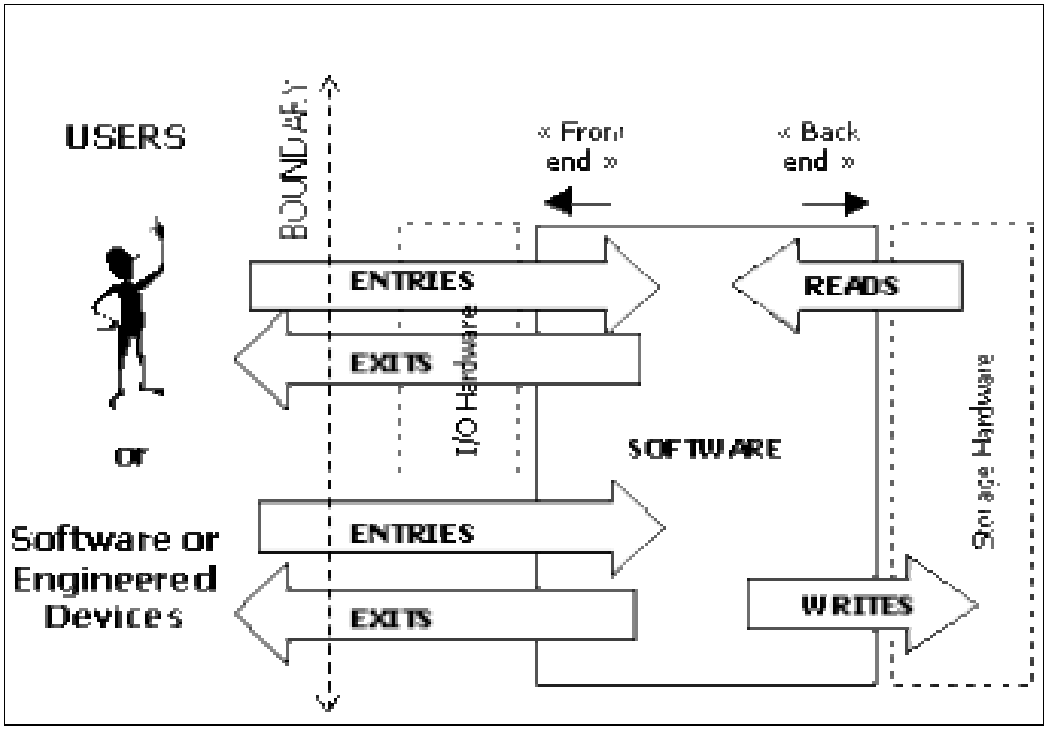


Figure A.1. Generic Flow of Data Attributes from a Functional Perspective[35]

Functional process types are broken down into four data movement types – Entry, Exit, Read, and Write. Each data movement is assumed to include certain data manipulations (e.g. data validation). Entries and Exits cross the boundary between the software being measured and its users (who may be any person or a different software layer or other piece of software). A Read moves data from

[35] A. Abran, J. Desharnais, S. Oligny, D. St-Pierre and C. Symons, *COSMIC-FFP Measurement Manual, version 2.2*, 2003.

"persistent storage" into the software being measured, while a Write makes data "persistent," meaning that the data lasts beyond the life of the functional process. Each instance of data movement equals one COSMIC Functional Size Unit (CFSU). The size of a functional process is the sum of its data movements, and the size of a piece of software is the sum of the sizes of all of its functional processes.

Measuring Pontis 5.0

Purpose of Measurement

The purpose of measurement is to measure the size of FUR of Pontis 5.0 as they evolve as input to cost estimating and to the measurement of productivity.

Scope of Measurement

The scope of measurement is Pontis 5.0, including work-output obtained by using existing re-usable code.

Measurement Viewpoint

The end-user measurement viewpoint allows a functional size to be measured of business application software as specified in the Functional User Requirement.

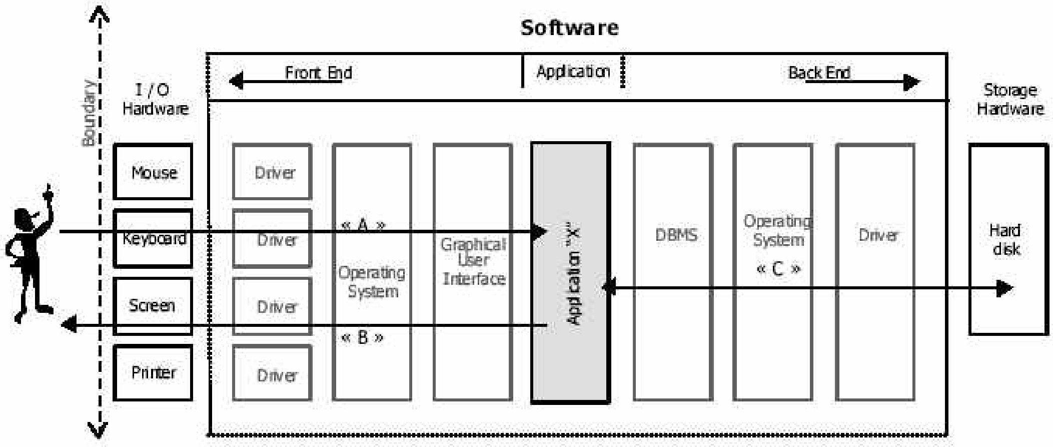


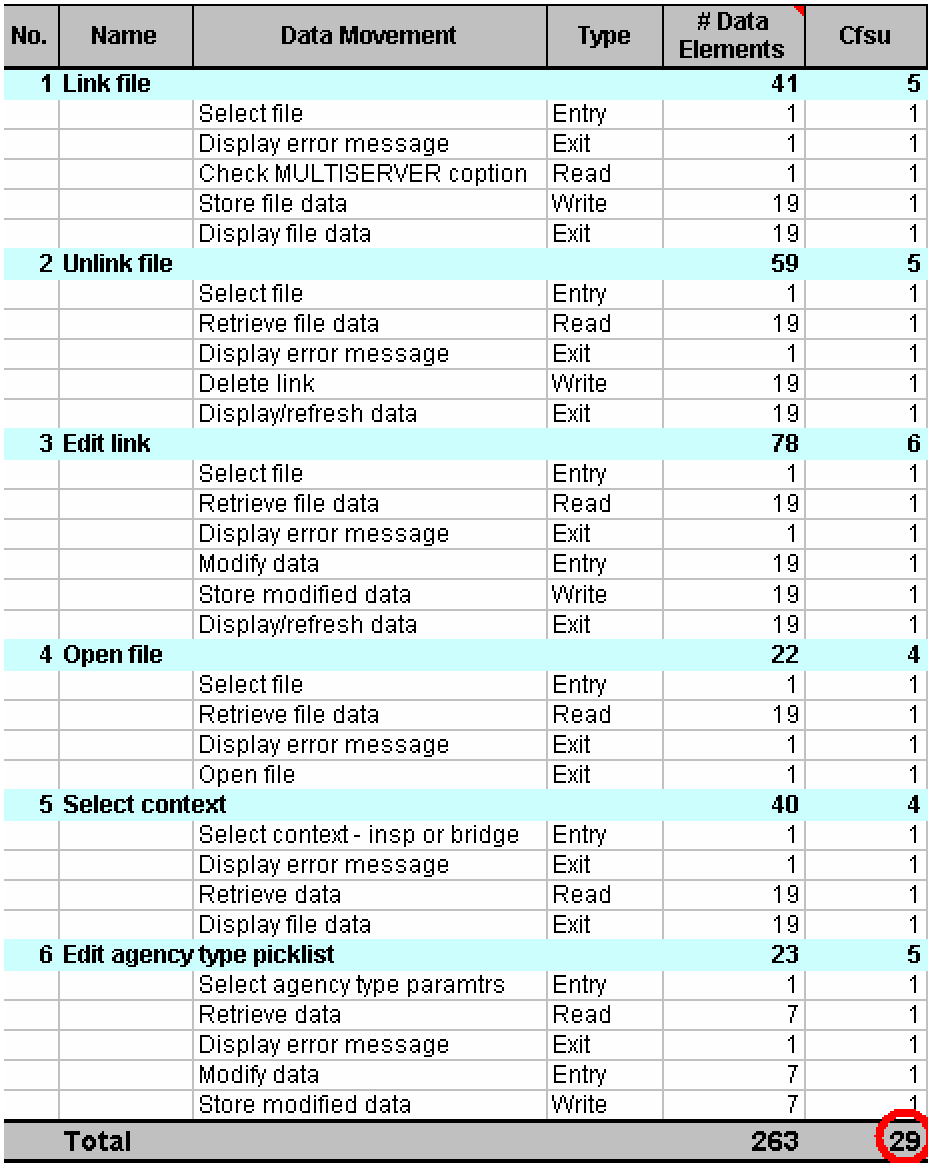
Figure A.2. Application Software Seen From the End User's Perspective[36]

[36] A. Abran, J. Desharnais, S. Oligny, D. St-Pierre and C. Symons, *COSMIC-FFP Measurement Manual, version 2.2*, 2003.

Measuring Mathematical Algorithms

COSMIC-FFP is not suited for measuring complex mathematical algorithms, which are a large, integral part of Pontis. To account for this limitation, the measurer is allowed to assign locally-determined functional size to the exceptional functionality.

Below is an example of COSMIC-FFP analysis on the Multimedia use case. The size of the Multimedia functionality is **29** CFSU.



No.	Name	Data Movement	Type	# Data Elements	Cfsu
1	**Link file**			**41**	**5**
		Select file	Entry	1	1
		Display error message	Exit	1	1
		Check MULTISERVER coption	Read	1	1
		Store file data	Write	19	1
		Display file data	Exit	19	1
2	**Unlink file**			**59**	**5**
		Select file	Entry	1	1
		Retrieve file data	Read	19	1
		Display error message	Exit	1	1
		Delete link	Write	19	1
		Display/refresh data	Exit	19	1
3	**Edit link**			**78**	**6**
		Select file	Entry	1	1
		Retrieve file data	Read	19	1
		Display error message	Exit	1	1
		Modify data	Entry	19	1
		Store modified data	Write	19	1
		Display/refresh data	Exit	19	1
4	**Open file**			**22**	**4**
		Select file	Entry	1	1
		Retrieve file data	Read	19	1
		Display error message	Exit	1	1
		Open file	Exit	1	1
5	**Select context**			**40**	**4**
		Select context - insp or bridge	Entry	1	1
		Display error message	Exit	1	1
		Retrieve data	Read	19	1
		Display file data	Exit	19	1
6	**Edit agency type picklist**			**23**	**5**
		Select agency type paramtrs	Entry	1	1
		Retrieve data	Read	7	1
		Display error message	Exit	1	1
		Modify data	Entry	7	1
		Store modified data	Write	7	1
	Total			**263**	**29**

Figure A.3. COSMIC-FFP Analysis of Multimedia Tab Functionality

Appendix B – Project Schedule

B.1 Dedicated Design and Release

The key tasks in the dedicated design and release approach are listed in Table B-1. A Gant chart showing the project schedule is presented in Figure B-1.

Table B-1. Dedicated Design and Release Tasks

Task ID	Name	Duration	Start	Finish
1.0	Pontis 5.0 Design			
1.1	Detailed Design	260	07/01/2005	06/29/2006
1.2	Prototype 1 Development	120	09/01/2005	02/15/2006
1.3	Demo Prototype 1	0	02/15/2006	02/15/2006
1.4	Prototype 2 Development	120	11/01/2005	04/17/2006
1.5	Demo Prototype 2	0	04/17/2006	04/17/2006
1.6	Prototype 3 Development	120	01/02/2006	06/16/2006
1.7	Demo Prototype 3	0	06/16/2006	06/16/2006
1.8	Pontis 5.0 Design Document	0	06/30/2006	06/30/2006
2.0	Pontis 5.0 Development			
2.1	Core Module Development	848	07/03/2006	09/30/2009
2.2	Inspection Module Development	848	07/03/2006	09/30/2009
2.3	Project Module Development	848	07/03/2006	09/30/2009
2.4	Programming Module Development	848	07/03/2006	09/30/2009
2.5	Configuration Module Development	848	07/03/2006	09/30/2009
2.6	Preservation Module Development	848	07/03/2006	09/30/2009
2.7	Gateway Module Development	848	07/03/2006	09/30/2009
2.8	Results Module Development	848	07/03/2006	09/30/2009
2.9	Pontis 5.0 Alpha Release	0	02/02/2009	02/02/2009
2.10	Pontis 5.0 Beta Release	0	06/30/2009	06/30/2009
3.0	Pontis 5.0 Release			
3.1	Pontis 5.0 Testing	120	07/01/2009	12/14/2009
3.2	Pontis 5.0 Final Release	0	12/31/2009	12/31/2009

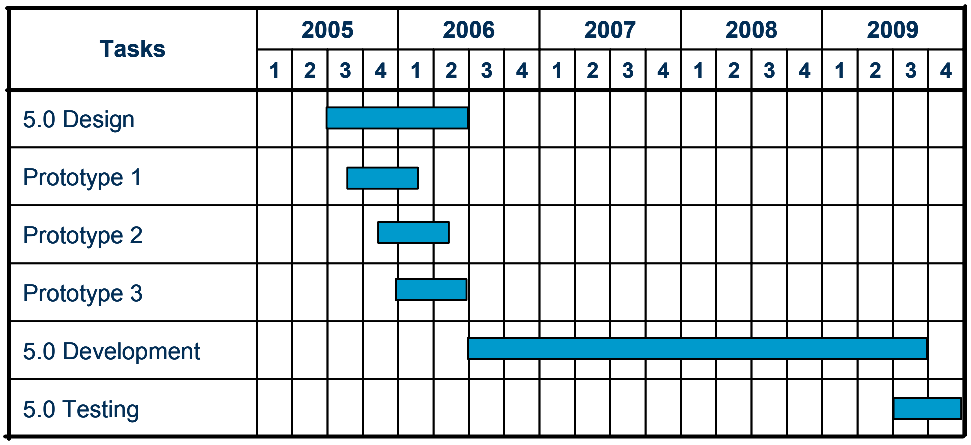


Figure B-1. Dedicated Design and Release Schedule

B.2 PHASED DESIGN AND RELEASE

The key tasks in the phased design and release approach are listed in Table B-2. A Gant chart showing the project schedule is presented in Figure B-2.

Table B-2. Phased Design and Release Tasks

Task ID	Name	Duration	Start	Finish
1.0	Pontis 5.0 Design			
1.1	Core Module Detailed Design	120	07/01/2005	12/15/2005
1.2	Inspection Module Detailed Design	120	07/01/2005	12/15/2005
1.3	Pontis 5.0 Design Document	0	12/30/2005	12/30/2005
2.0	Pontis 5.0 Development			
2.1	Core Module Development	240	01/02/2006	12/01/2006
2.2	Inspection Module Development	240	01/02/2006	12/01/2006
2.3	Pontis 5.0 Alpha Release	0	06/30/2006	06/30/2006
2.4	Pontis 5.0 Beta Release	0	09/29/2006	09/29/2006
3.0	Pontis 5.0 Release			
3.1	Pontis 5.0 Testing	120	10/02/2006	03/16/2007
3.2	Pontis 5.0 Final Release	0	03/30/2007	03/30/2007
4.0	Pontis 5.1 Design			

4.1	Project Planning Module Detailed Design	120	07/03/2006	12/15/2006
4.2	Gateway Module Detailed Design	120	07/03/2006	12/15/2006
4.3	Pontis 5.1 Design Document	0	12/29/2006	12/29/2006
5.0	Pontis 5.1 Development			
5.1	Project Planning Module Development	240	01/01/2007	11/30/2007
5.2	Gateway Module Development	240	01/01/2007	11/30/2007
5.3	Pontis 5.1 Alpha Release	0	06/29/2007	06/29/2007
5.4	Pontis 5.1 Beta Release	0	09/28/2007	09/28/2007
6.0	Pontis 5.1 Release			
6.1	Pontis 5.1 Testing	120	10/01/2007	03/14/2008
6.2	Pontis 5.1 Final Release	0	03/31/2008	03/31/2008
7.0	Pontis 5.2 Design			
7.1	Preservation Module Detailed Design	260	07/02/2007	06/27/2008
7.2	Programming Module Detailed Design	260	07/02/2007	06/27/2008
7.3	Configuration Module Detailed Design	260	07/02/2007	06/27/2008
7.4	Results Module Detailed Design	260	07/02/2007	06/27/2008
7.5	Pontis 5.2 Design Document	0	06/30/2008	06/30/2008
8.0	Pontis 5.2 Development			
8.1	Preservation Module Development	260	07/02/2007	06/27/2008
8.2	Programming Module Development	260	07/02/2007	06/27/2008
8.3	Configuration Module Development	260	07/02/2007	06/27/2008
8.4	Results Module Development	260	07/02/2007	06/27/2008
8.5	Pontis 5.2 Alpha Release	0	06/30/2008	06/30/2008
8.6	Pontis 5.2 Beta Release	260	07/02/2007	06/27/2008
9.0	Pontis 5.2 Release			
9.1	Pontis 5.2 Testing	120	10/30/2009	04/15/2010
9.2	Pontis 5.2 Final Release	0	04/30/2010	04/30/2010

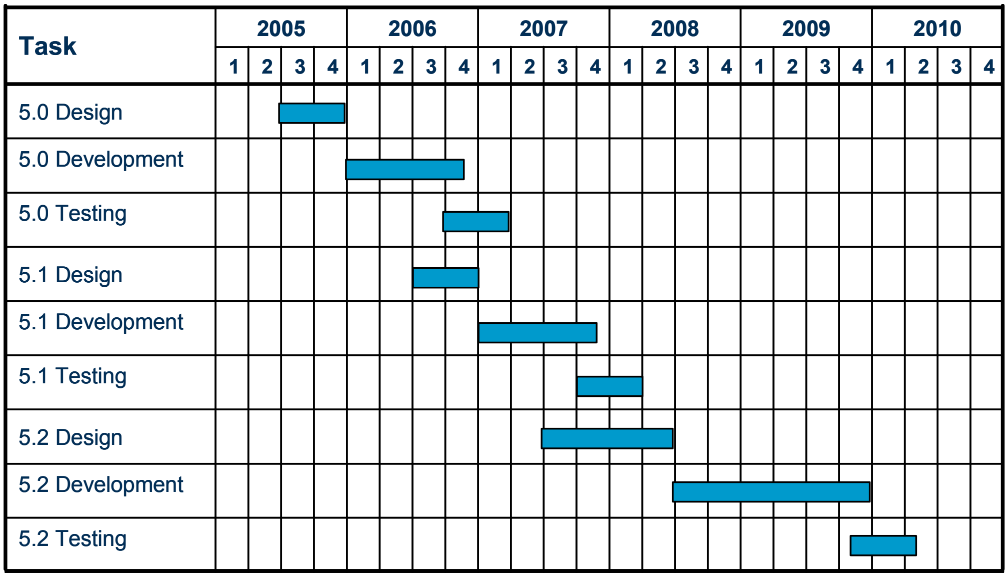


Figure B-2. Phased Design and Release Schedule

B.3 Dedicated Design and Phased Release

The key tasks in the dedicated design and phased release approach are listed in Table B-3. A Gant chart showing the project schedule is presented in Figure B-3.

Table B-3. Dedicated Design and Phased Release Tasks

Task ID	Name	Duration	Start	Finish
1.0	Pontis 5.0 Design			
1.1	Detailed Design	240	07/01/2005	06/01/2006
1.2	Prototype 1 Development	120	09/01/2005	02/15/2006
1.3	Demo Prototype 1	0	02/15/2006	02/15/2006
1.4	Prototype 2 Development	120	11/01/2005	04/17/2006
1.5	Demo Prototype 2	0	04/17/2006	04/17/2006
1.6	Prototype 3 Development	120	01/02/2006	06/16/2006
1.7	Demo Prototype 3	0	06/16/2006	06/16/2006
1.8	Pontis 5.0 Design Document	0	06/30/2006	06/30/2006
2.0	Pontis 5.0 Development			
2.1	Core Module Development	195	07/03/2006	03/30/2007
2.2	Inspection Module Development	195	07/03/2006	03/30/2007
2.3	Pontis 5.0 Alpha Release	0	11/01/2006	11/01/2006

2.4	Pontis 5.0 Beta Release	0	02/01/2007	02/01/2007
3.0	Pontis 5.0 Release			
3.1	Pontis 5.0 Testing	107	02/01/2007	06/29/2007
3.2	Pontis 5.0 Final Release	0	06/29/2007	06/29/2007
4.0	Pontis 5.1 Development			
4.1	Project Planning Module Development	196	07/02/2007	03/31/2008
4.2	Gateway Module Development	196	07/02/2007	03/31/2008
4.3	Pontis 5.1 Alpha Release	0	11/01/2007	11/01/2007
4.4	Pontis 5.1 Beta Release	0	02/01/2008	02/01/2008
5.0	Pontis 5.1 Release			
5.1	Pontis 5.1 Testing	107	02/01/2008	06/16/2008
5.2	Pontis 5.1 Final Release	0	06/30/2008	06/30/2008
6.0	Pontis 5.2 Development			
6.1	Preservation Module Development	457	07/01/2008	03/31/2010
6.2	Programming Module Development	457	07/01/2008	03/31/2010
6.3	Configuration Module Development	457	07/01/2008	03/31/2010
6.4	Results Module Development	457	07/01/2008	03/31/2010
6.5	Pontis 5.2 Alpha Release	0	011/01/2009	11/01/2009
6.6	Pontis 5.2 Beta Release	0	02/01/2010	02/01/2010
7.0	Pontis 5.2 Release			
7.1	Pontis 5.2 Testing	108	02/01/2010	06/17/2010
7.2	Pontis 5.2 Final Release	0	06/30/2010	06/30/2010

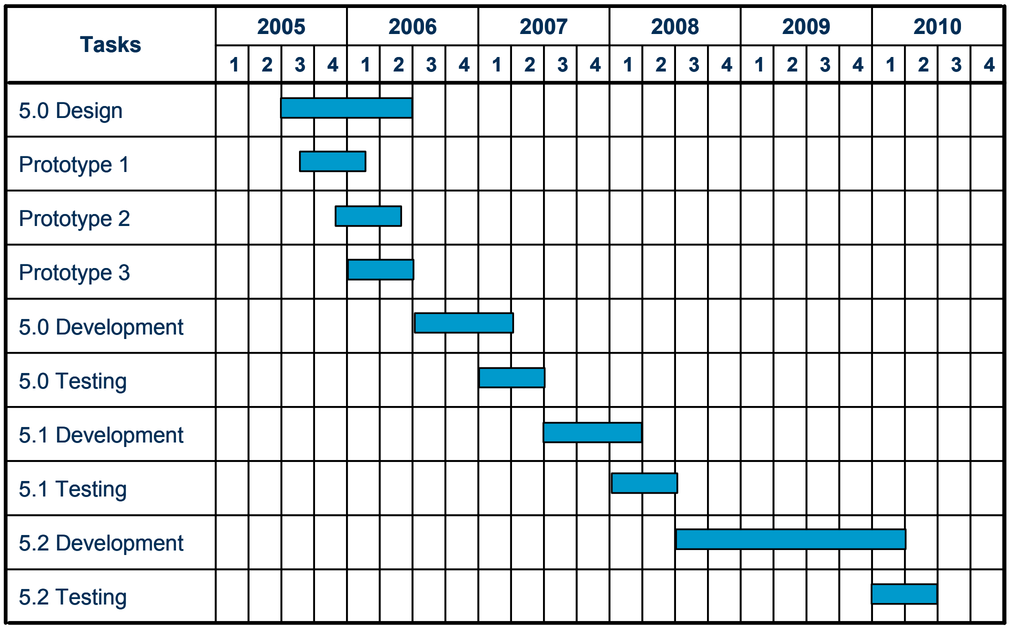


Figure B-3. Dedicated Design and Phased Release Schedule