A NEW BRIDGE MANAGEMENT SYSTEM FOR ONTARIO

Paul D. Thompson, Consultant, USA
Brian Kerr, ITX Stanley Ltd., Canada

Abstract

Ontario's Ministry of Transportation (MTO) is responsible for the management of approximately 3000 bridges on the provincial highway network. A modern Bridge Management System is an essential tool to aid the Ministry in fulfilling its mandate to provide a safe, economical, and effective road network throughout the province. The Ministry has engaged ITX Stanley, Ltd., to provide this tool, to be called OBMS, in an 18-month project, which began on January 6, 1998.

Written in Visual Basic for client-server deployment, OBMS is a completely new system from the ground-up, and not an adaptation of any existing system. The use of object-oriented methods for design and development have led to new insights on the organization of the user interface and data management methods; the ability to incorporate third-party capabilities for mapping and document management; and the introduction of a potentially very fast new process for analyst-in-the-loop optimization at both the network and project levels.

Keywords: structures, Bridge Management System, object-oriented, optimization

1. Introduction

Long an innovator in structural software systems, the Ontario Ministry of Transportation (MTO) in recent years has been evaluating recent developments in Bridge Management Systems (BMS), such as AASHTO's Pontis™ (Cambridge, 1997) and NCHRP's Bridgit (National, 1994), in relation to its own requirements. In 1997, MTO decided to launch its own system. After a competitive procurement process, the Ministry retained ITX Stanley Ltd., with subcontract support from Paul D. Thompson, to develop the new system, to be called OBMS. The 18-month project, which began on January 6, 1998, features a 6-month design phase, followed by system development. This paper presents some of the insights learned so far from the design and development process, in particular the contributions of new object-oriented technology.

2. Background and objectives

The mandate of the Ministry of Transportation of Ontario is to provide a safe, economical and effective road network throughout the province to allow the movement of people and goods. An effective bridge management strategy is very important in fulfilling this mandate. Fiscal restraints are putting more pressure on using government funds for infrastructure spending as efficiently as possible. Consequently tools to assist in deciding how to allocate these limited funds effectively are becoming increasingly important to the Ministry.

The Ministry currently has a well defined process for bridge management. A number of manuals, standards and systems currently in place at the Ministry support this process. However, the existing mainframe based systems do not adhere to the Ministry's new Technical Architecture, which includes a standardized desktop based on the WINTEL platform running Windows 95 with NT and UNIX servers. The systems also lack many of the features of a modern BMS, especially in the area of network level analysis to determine bridge funding needs and priorities. In developing the BMS, therefore, the Ministry's objectives include:

- Supply senior management with the tools and information to assist in making decisions on the bridge network and to recognise the risks involved in the decisions.

- Provide leadership, assistance and facilities to other levels of government within the province of Ontario in the area of Bridge Management practices and policies.

- Provide data to staff in charge of day-to-day maintenance of the highway infrastructure, delivery of transportation facilities, and development of policies and standards for construction and maintenance of the highway infrastructure.

- Support the workflow in all processes involved in managing the bridge infrastructure.

- Integrate all Ministry bridge management activities from inventory and inspection to bridge MR&R decisions into a single system with appropriate links to other systems.

- Improve on present data bases in terms of on-line availability, accurate data entry and integrity, security, integration, efficient computer usage, reporting and system interface so that the system can be easily accessed and used by consultants and Ministry staff.

- Upgrade present systems to conform to current Ministry information system standards.

3. System overview

In order to organize the functionality of the system into identifiable modules, the development team produced a business process model and use-case analysis, in a process similar to what is described in (Jacobson et al, 1995). Fig. 1 shows a high-level summary of the business process model. In general, monitoring, needs identification, and priority-setting are performed by the Ministry's five Regions, while budgeting, funding allocation, and expenditure planning are performed in the head office. Most work implementation is administered by the Regions. This business process model, describing what the agency would like to do with BMS support, evolved into a more detailed use-case model, describing what the system would do to support the agency. This has ultimately provided an outline for the system's graphic user interface.

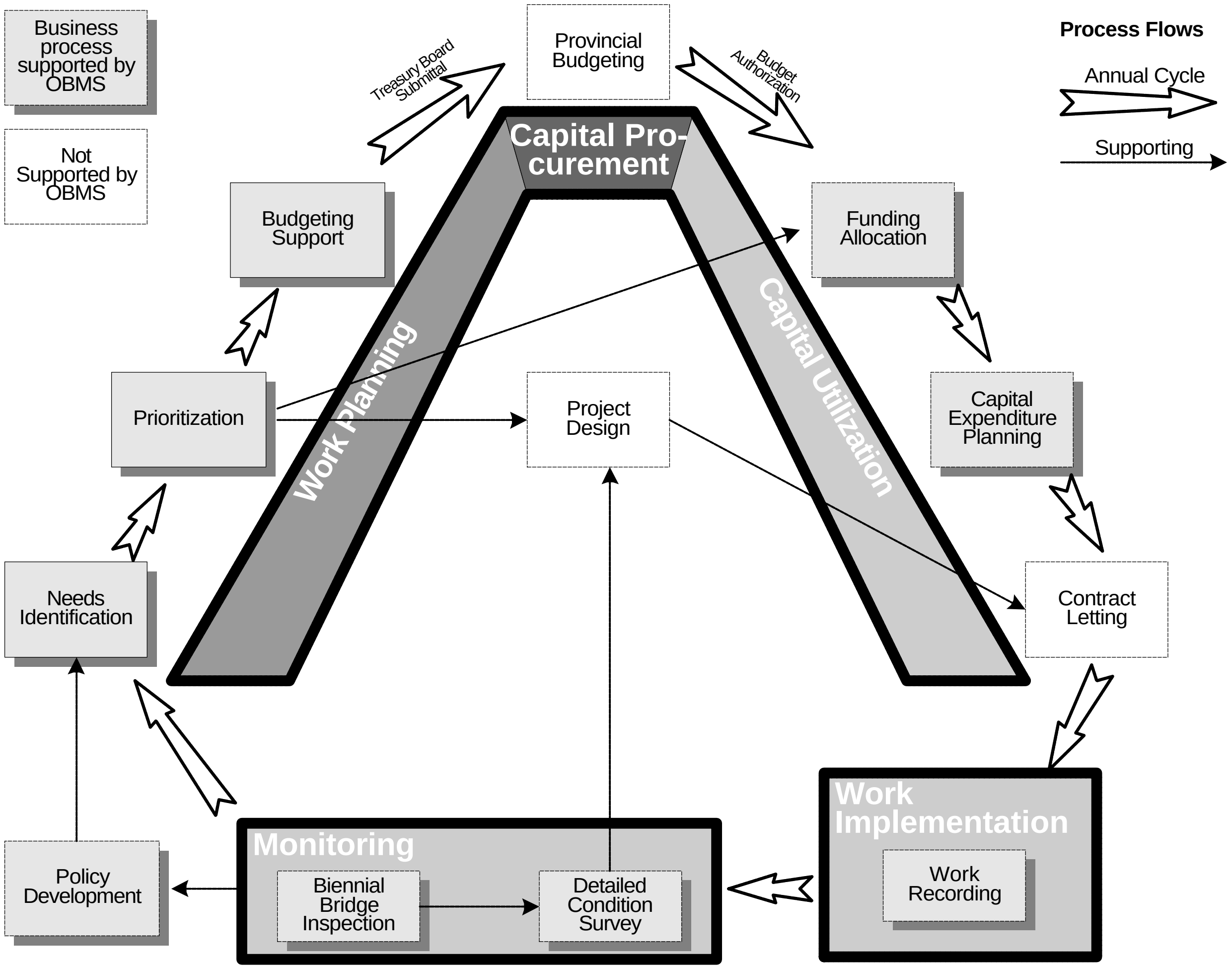

Fig. 1. Business Process Model

Also supporting the design process and organizing the system is a domain model, a concept taken from object-oriented analysis (Booch, 1994). The domain model organizes the "nouns" (things and concepts) which the software is intended to represent. Fig. 2 shows a portion of this model from the OBMS. The domain model has evolved into an outline for the relational database, and also forms a portion of the outline of the software now being written.

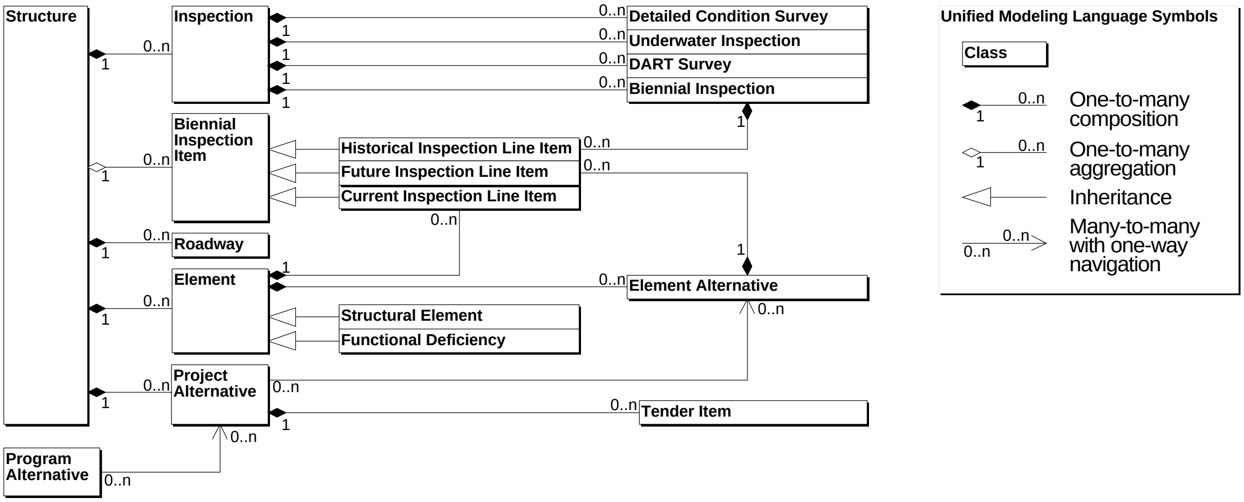

Fig. 2. Domain Model

As the primary source of data for the BMS, both models devote attention to the biennial element-level bridge inspection. This involves recording the type, severity, and extent of deterioration of each major structural element, such as decks, girders, joints, bearings, and pier caps. Consistent with many other BMS, condition is expressed as the percent or quantity of each element found to be in each of a small number of condition states. Compared to typical inspection practice in the United States, the Ontario process has somewhat more elements; for example, wearing surfaces and drainage systems are recorded separately from decks, and coatings separately from steel elements. Each element has four possible condition states, which for convenience are referred to as Excellent, Good, Fair, and Poor, but which actually have precise engineering definitions.

For structures that are viewed as likely to receive work within the next four years, MTO conducts a Detailed Condition Survey and/or a DART survey. These data collection processes measure concrete deterioration effects such as delamination, cracking, corrosion potential, and chloride content. Although expensive to measure network-wide on an on-going basis, these activities are indispensable for detailed scoping and costing of projects.

The policy development and needs identification use-cases exploit some very strong existing MTO tools, including decision rules for the selection of rehabilitation treatments and coatings, and a well-maintained analytical process for project costing based on tender item tabulations. These resources are described below. The agency is also especially concerned with capturing the wealth of accumulated knowledge of a very experienced staff. As a result of this strong foundation, the OBMS includes an extensive knowledge-based component at the project level, and a novel approach to the estimation of direct and indirect costs.

In support of the prioritization, budgeting and resource allocation use-cases, the Ministry also has some very demanding expectations for network-level analysis. In particular, it is required that network-level outputs remain fully consistent with all project-level inputs. In the optimization process, it is therefore necessary to satisfy both network-level and project-level constraints simultaneously. With a high level of interest in features that could directly relate investment levels to transportation system performance, the system demands an unusually speedy mechanism for “what-if” analysis. The unique approach being taken to address this problem, based on object technology, is described below.

4. Project-level models

The project-level analysis of OBMS produces a list of project alternatives at varying funding levels, and selects one alternative for each program period to make up the optimal strategy for the bridge. To select and evaluate alternatives, the model performs a life cycle cost analysis. Fig. 3 shows the major components and data flows of the project-level analysis.

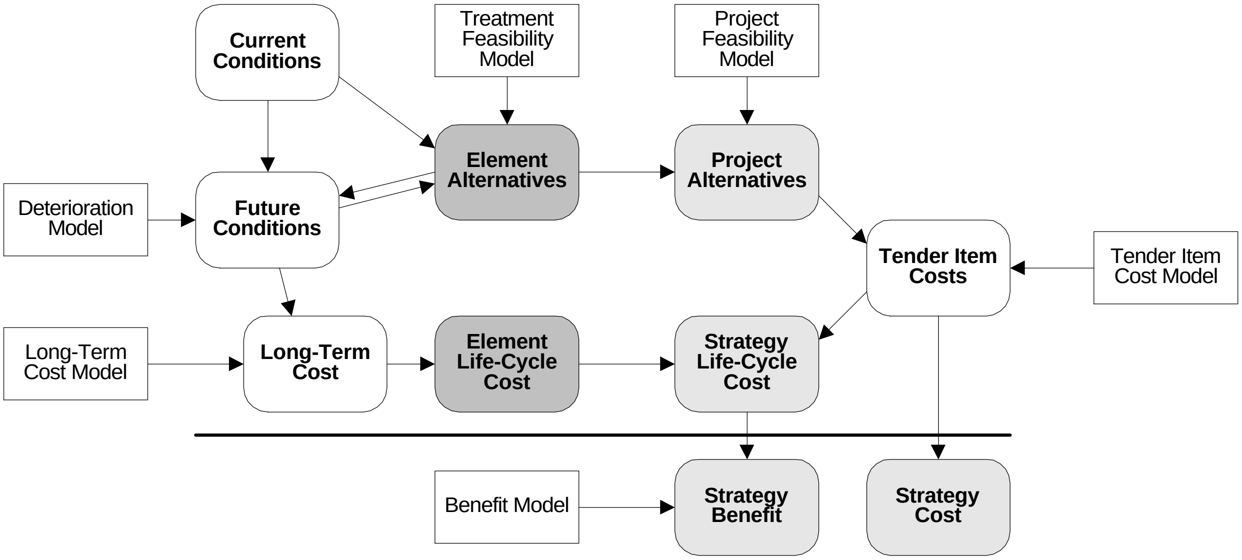

Fig. 3. Project-Level Models

The analysis begins with the identification of needs on individual elements, driven by element condition as determined in recent inspections. Since there are several different kinds of inspections, the software must find the most useful and relevant condition data. For example, a bridge deck element may find this information in a biennial inspection, detailed condition survey, or DART survey. Viewed as an object-oriented modeling problem, the system has a Current Condition object (Fig. 2) which acts as a simplified facade for a potentially complex relationship among different types of recent inspections. Façade patterns such as this are simple but powerful tools to limit the complexity of a system (Gamma et al, 1995).

Based on element condition, a knowledge-based model identifies feasible treatment alternatives. For each possible treatment, the deterioration model predicts the condition of the element at the end of the period. This predicted condition becomes the basis for generating a list of feasible treatments for the second period, which in turn allows the deterioration model to predict the condition at the end of the second period for each treatment. This simulation pattern is a traditional life cycle cost analysis that can be repeated for as many periods as desired.

Each possible combination of element-level treatments on a bridge is a potential project alternative. Although in theory there are billions of possible alternatives on a typical bridge, only a few are practical from an economic and engineering standpoint. The system uses a benefit/cost analysis and a knowledge-based model to reduce the list of alternatives to a manageable number.

Once each project alternative is defined, its cost is estimated by consulting the Ministry's extensive tabulation of tender item unit costs, which cover most typical direct and indirect cost categories. Knowledge-based models estimate the tender item quantities, and average benchmark costs supplement the process for cost items not covered by the Ministry's database.

To avoid unnecessary calculations and reduce computational demands on the system, a long-term cost model predicts the outcomes of typical policies that might be followed beyond the end of the decision-making horizon, based on predicted condition at that time and the deterioration and cost models. The total life-cycle cost of a project alternative is the sum of all discounted agency costs, including the estimated long-term cost beyond the program horizon.

The benefit of a project alternative is the savings in life cycle social cost that is achieved by implementing the project rather than doing nothing. It includes the difference in life cycle cost between the subject project alternative and the do-nothing alternative, plus the predicted savings in user cost. "Soft factors," such as political mandates and project interrelationships, determined manually or by knowledge-based models, can also contribute to the benefit calculation.

4.1 Knowledge-based models

One of the most important steps in the rehabilitation of a bridge is the selection of the rehabilitation method. The Treatment Feasibility model identifies technically viable options for rehabilitation of various bridge elements. The model makes use of the decision trees and tables contained in the Ministry's Structure Rehabilitation Manual (MTO, 1993) and Structural Steel Coating Manual (MTO, 1992). Many of the rehabilitation methods identified in these manuals have been used by the Ministry since 1978 and have been working well. A typical decision tree from the Structure Rehabilitation Manual is given in Figure 4.

The decision trees and tables in these manuals are used when data from a detailed condition survey are available. These decision trees may also include consideration of other information such as traffic volumes and related road work. Where more than one rehabilitation method is possible or where the choice between rehabilitation and replacement is not obvious, the decision tree will recommend that a financial analysis be carried out. In the past, this was normally done using spreadsheets developed by the Ministry as given in the Structure Financial Analysis Manual (MTO, 1990). The new BMS will include financial analysis as an integral part of the treatment selection process, thus providing evaluated alternatives for consideration in the optimization model.

For bridges where a detailed condition survey is not available, the Ministry is currently in the process of identifying standard or accepted repair options for each element condition state that is reported during the biennial visual inspection. This provides a simplified set of decision rules which parallels the thought process traditionally taken by bridge inspectors. These repair options will be included in the treatment feasibility model. Like the more detailed decision rules of the Structure Rehabilitation Manual and Structural Steel Coating Manual, the simplified rules are not intended to find the one best option, but instead are intended to find all feasible alternatives that should be considered in the optimization.

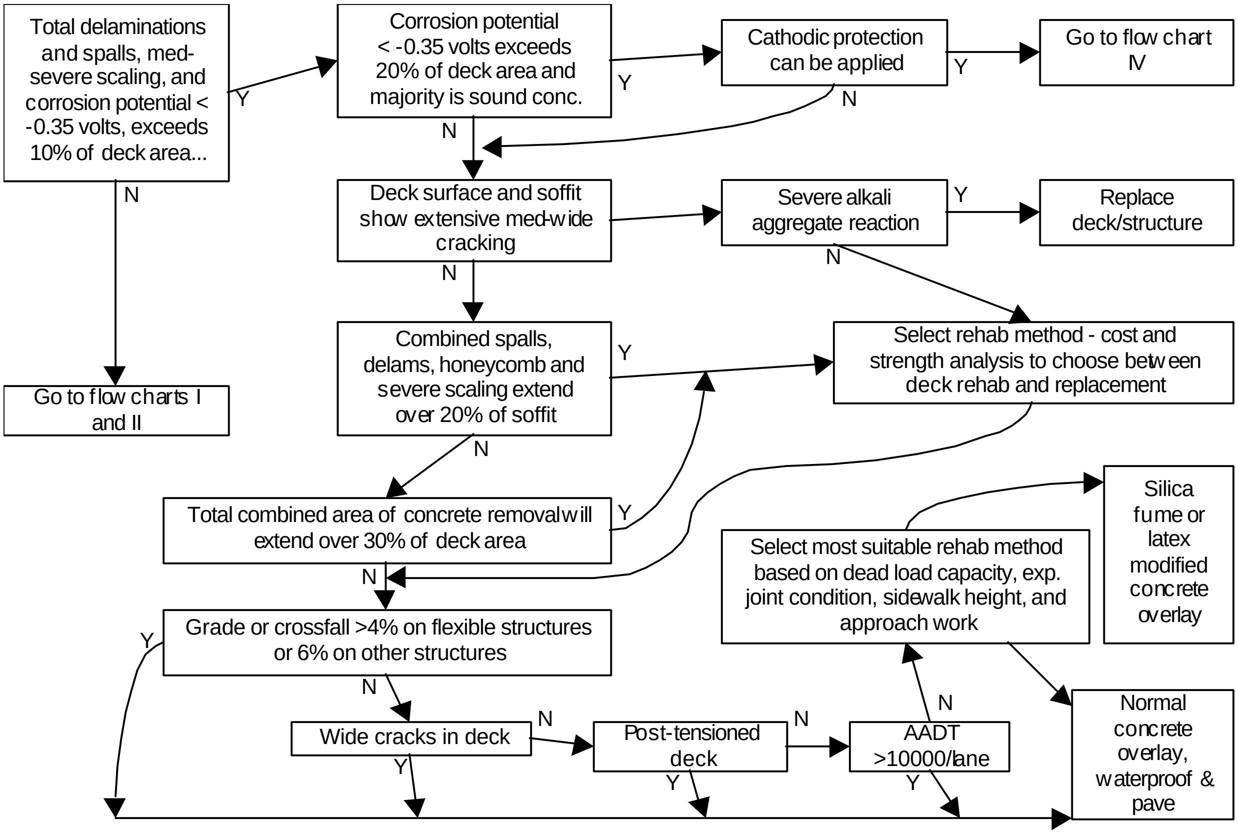

Fig. 4. Sample Decision Tree for Selecting Bridge Deck Rehabilitation Method

4.2 Deterioration models

Element-level needs are driven by a Markovian deterioration model. The choice of deterioration model form tends to be driven by certain factors unique to bridges which are not exhibited to as great an extent in other types of assets. This includes a very small condition state space and long life, which together yield long transition times among states; and deterioration which is not smooth over time, strongly sensitive to phenomena which are not observed in visual inspections.

The Markovian model takes advantage of the discrete condition states identified for inspections, to provide a simple way of describing the likelihood of each possible change in condition over time. Markov models assume that measurements are taken or used at evenly-spaced intervals, and that the condition in the next interval is dependent only on the current condition state and not on any other attribute, including time. Fig. 5 shows how a Markovian model describes the change in condition of a new element over time.

Although it is possible to predict deterioration with other forms of models, including deterministic models, the Markovian model is particularly suitable for the available condition data. It expresses its predictions in the same form as inspections, as a distribution of the element among condition states, explicitly recognizing and using information about the uncertainty of deterioration. It thus provides an unbiased estimate of needs within any time frame. In addition, since the inspection process is being changed to support the new BMS capabilities, it is necessary to apply techniques that do not require long time-series of data for model estimation.

Markovian models require only two successive cycles of inspection, for most elements, before model estimation becomes possible.

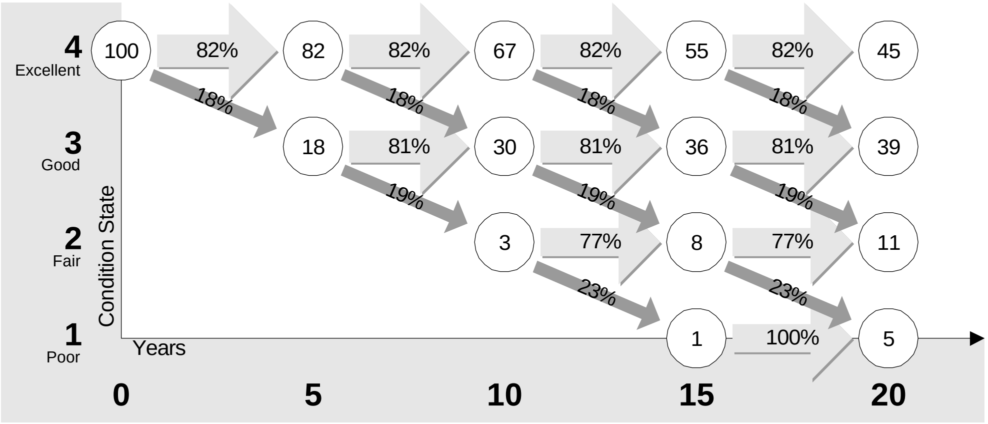

Fig. 5. Example of a Markovian Model.

A concern about Markovian models is the assumption that future deterioration depends only on the current condition state, and not on time or any other attribute of the element or bridge. Since the models are calibrated to reflect the history of a whole bridge population, this assumption is usually acceptable at the network level, where only average behavior of the inventory is needed. However, at the project level, many characteristics of the specific element or bridge may cause its deterioration to be faster or slower than the inventory average for that element.

Since the Ontario BMS models are driven entirely from a project-level life cycle analysis, it becomes possible to modify the Markovian models by calculating project-level adjustment factors to deterioration probabilities. These adjustment factors would come from knowledge-based models reflecting any relevant bridge or element attributes in the database, including the behavior of other elements on the same bridge, and the attributes of the environment in which the element resides. For example, the deterioration of girders can be modified based on the year of construction and the degree of leakage of deck joints.

4.3 Cost models

To the greatest extent possible, it is desired that cost estimates for project alternatives be based on tender item unit costs. The unit costs maintained by MTO are based on actual contract history, are updated continuously, and reflect the differences among the 12 districts in the province.

In order to apply tender item costs, it is necessary to be able to estimate tender item quantities based on the characteristics of the bridge and the scope of work. Often this can be done easily when a recent detailed condition survey or DART survey is available. In the absence of these data sources, it is necessary to look at the element-level treatments and try to deduce the tender item quantities. The modeling framework provides a knowledge-based model to do this.

Separate tender item knowledge models are defined for each type of tender item. Each of these models is applied first at the level of element alternatives, then for project alternatives. This gives each treatment a chance to contribute to the tender item quantity, and then gives the project as a whole a chance to modify the accumulated tender item results and calculate unit and total costs. When applied at the element alternative level, the knowledge model has access to all the same information as the deterioration knowledge model, described above.

Once the tender item quantity is determined, it is multiplied by the unit cost. This is performed within the knowledge model at the project alternative level to allow systematic adjustments to unit costs, to allow a fixed cost to be added, or to allow any other appropriate non-linear calculation. Since many tender items indicate only lump sum costs and not unit costs, the knowledge model may need to provide an allocation method.

In a few cases, it may be impossible to determine either the tender item quantity or the tender item cost. When this is the case, an element-level benchmark cost is used. The knowledge model can use this cost in its calculations in the same way as it would use a tender item unit cost. It is even possible to combine the use of the two types of costs.

5. Network-level models

The network-level analysis of OBMS finds the set of project alternatives that maximizes benefit within budget constraints. Since all benefits are expressed as avoided social costs, the analysis also minimizes social costs. It provides summary predictions of network-wide performance at any given funding level. This makes the analysis very useful in the budgetary process, since it allows the Ministry to express funding needs in terms of the level of performance that can be achieved. Fig. 6 depicts the data flows in the model.

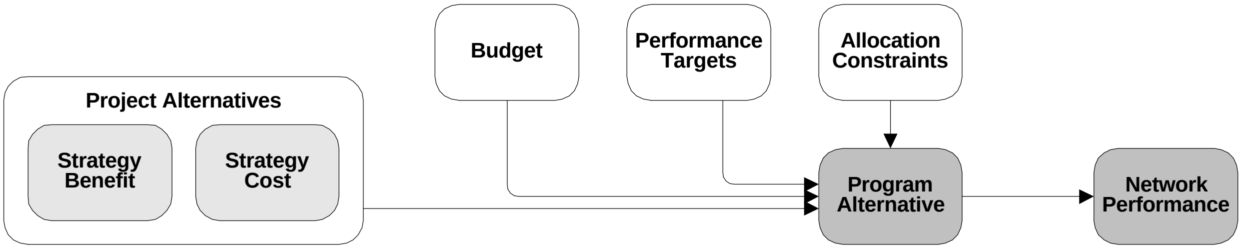

Fig. 6. Network-Level Models

In program management analysis, it is important to distinguish between "hard" constraints and targets. Budgets are considered hard constraints, because in general a program will not be implemented beyond the bounds of available funding. Funding allocation constraints are similar, modifying the budget constraints to affect only a subset of the inventory. With performance constraints, however, there is no institutional mechanism to regulate implementation to ensure that predicted levels of performance are actually achieved. The degree of achievement of performance targets is not known until long after implementation, and there is significant uncertainty in the relationship between the programmatic decisions and the actual outcome.

From these considerations, it is desired that the optimization model respect the overall budget constraint as highest priority, respect the funding allocation constraints as long as there are feasible solutions which can do so, and just report on, but not enforce, the degree to which performance targets are predicted to be met. In every case, whether or not the constraints can all be satisfied, the system must report the solution which can achieve the highest benefit while meeting as many of the funding constraints as possible. It must also report on the level of funding required to satisfy all performance targets.

Network-level optimization problems in bridge management systems have often been solved by defining and solving mathematical programs. Pontis (Golabi et al, 1992), for example, performs its network-level analysis as a Markovian decision process, formulated and solved as a linear program. Deterioration and cost models are expressed in the constraints of the program. For the Ontario framework, however, this approach would be computationally impractical, because the deterioration and cost models are bridge-specific and non-linear.

Fortunately, an object-oriented perspective suggests a different way to approach the problem. As indicated in Fig. 2, each element alternative and each project alternative is defined as a separate object. Each of these objects is responsible for actively keeping itself up-to-date, managing its own persistence, and evaluating its own role in a potential optimal program. The updating capability includes changing its own state when notified of some external stimulus that might affect it. Bridge inspection activity is by far the most common such stimulus. Knowledge-based processes regulate the creation and activity of these objects.

With this framework of self-regulating objects established, it is natural to envision a dynamic, or incremental, approach to maintaining the state of optimality of a program once the initial solution is established. This concept is analogous to “engineer-in-the-loop” or parametric engineering design, so it is referred to as “analyst-in-the-loop.” OBMS will respond to any data input, be it inspection, budget constraints, or knowledge-based decision rules, by starting from a known optimal solution and making incremental adjustments to find a new optimal solution consistent with the new stimulus. The computational effort is then limited to constraining the scope of cascading state changes so a new equilibrium is achieved quickly.

This incremental approach is feasible with many different kinds of optimization algorithms. In the simplex method for linear programming, for example, it is common to perform “what-if” analysis by pivoting from one known solution to another. Since the OBMS problem involves choices from among discrete alternatives, the problem structure is more similar to integer programming. However, since there is only one hard constraint applicable to each bridge, a special case known as incremental benefit/cost analysis, using a gradient search method, is very efficient and produces near-optimal results. The deviation from optimality with this method can be contained to be well within the range of uncertainty of budget constraints.

6. Design and development process

This concept of self-regulating software objects would be rather complex to implement with traditional structured analysis and design techniques, but it is much more suitable for object-oriented development. This implies a much different approach to design and development than has traditionally been used in management systems.

Object oriented design and development promote modular systems and the cost effectiveness associated with modular software. In particular for this project, the Component Object Model (COM) based architecture enables the client side of the application to consist of modules that are independent and interact with each other and the database only through the COM. One implication is that the application is isolated from the physical database, and therefore from specific database management systems. The database management system can be changed without major impact on the application software, and performance of the analytical process is not as strongly affected by the speed of the database manager or network. In addition, the COM provides a standard interface to external systems, which reduces the impact of future expansion.

This is an important consideration for the BMS in that several external systems are to be interfaced with it. The system architecture features a data services layer as the relational representation of the physical database between the database and the domain model layer. External systems communicate with the database through the domain model, which in turn accesses the database through the data services layer. This provides flexibility for such interfaces with reduced development and maintenance effort.

External interfaces in this development include those made to the existing Bridge Document Image Management System (BDIMS), and the Ministry’s ArcInfo GIS. The BDIMS interface will enable BMS users to access bridge related documents (plans, drawings, photographs, etc.) for viewing or to download to the local client for local storage and viewing. A MapObjects interface is planned for the display of bridge data on maps.

While important to the system architecture, the object-oriented approach used in this project is also very important to the project work plan. This project has been organized in three phases: the design phase, a phase for the development of the data management subsystem, and a third phase for the development of the decision support and analysis subsystem. The latter two phases overlap significantly in terms of schedule. The design phase included the following steps:

- Requirements definition, which produced the business process model, a general description of the system requirements, and the domain model.

- Logical and physical database design.

- Analytical model design, identifying the quantitative and knowledge based models.

- Graphic user interface design based on Microsoft’s Windows User Interface.

- Development of the component object model.

- Development of an executable architecture which demonstrates all of the central mechanisms to be developed in the subsequent phases.

This last step is a critical one in that it is a risk management tool for the development team. The executable architecture demonstrates that the system architecture is workable and ‘doable’ and is the bridge between finalization of the design and initiation of development.

In the development phases of the project, the object-oriented approach allows fast track development at reduced risk through the use of a “spiral development” process. An initial release of the system will include core utility classes, interface objects, data services objects, and the basic BMS desktop and screens for data viewing and editing. A second release will include the field inspection subsystem and a populated database. In phase 3, the analytical components will be added and a final release of the system produced. This incremental development approach reduces risk in that it forces the development team (both Ministry and Consultant) to resolve any critical problems early in the development.

7. Conclusions

An object-oriented perspective has allowed the development team to approach a well-known, difficult computational problem and organize it into a form that is far more tractable. Some of the most difficult parts of the problem, including the knowledge-based decision rules for treatment selection, and tender item costing, have already been worked out by the Ministry over many years of research and experimentation. The optimization method, as of this writing, remains conceptual and is still under development. However, it is composed of ingredients that have already been proven in many other systems, such as incremental benefit/cost analysis and self-contained software objects. When completed, the system may demonstrate an attractive new approach to infrastructure management systems.

8. References

Booch, G., (1994), Object-Oriented Analysis and Design with Applications, Addison-Wesley, New York, NY, USA, 1994.

Cambridge Systematics, Inc., (1997), Pontis 3.2 Users Manual, American Association of State Highway and Transportation Officials, Washington, DC, USA, 1997.

Gamma, E., Helm, R., Johnson, R., and Vlissides, J., (1995), Design Patterns: Elements of Reusable Object-Oriented Software, Addison-Wesley, New York, NY, USA, 1995.

Golabi, K., Thompson, P.D., and Hyman, W., (1992), Pontis Technical Manual, Federal Highway Administration, Washington, DC, USA, 1992.

Jacobson, I., Ericsson, M., and Jacobson, A., (1995), The Object Advantage: Business Process Reengineering with Object Technology, Addison-Wesley, New York, NY, USA, 1995.

MTO, (1993), Structure Rehabilitation Manual, Ministry of Transportation of Ontario, Structural Office, Downsview, Ontario, Canada, 1993.

MTO, (1992), Structural Steel Coating Manual, Ministry of Transportation of Ontario, Structural Office, Downsview, Ontario, Canada, 1992.

MTO, (1992), Structural Financial Analysis, Ministry of Transportation of Ontario, Structural Office, Downsview, Ontario, Canada, 1990.

National Engineering Technology Corporation, (1994), Bridgit Users Manual, National Cooperative Highway Research Program, Washington, DC, USA, 1994.

9. Authors

Paul D. Thompson, 2425 Hawken Drive, Castle Rock, CO, 80104, USA.

Brian Kerr, ITX Stanley Ltd., 152 Main Street, Cambridge, ON, N1R 6R1, Canada

The authors wish to acknowledge the contribution of Tony Merlo of the MTO to parts of this paper.