Architecture of a Distributed, Model-Based Asset Management System

M.A. Hassanain, T.M. Froese and D.J. Vanier

This article was originally presented at the:

Conference of the Canadian Society for Civil Engineers

Victoria, BC

May 30 – June 2, 2001

https://csce.ca/en/publications/past-conferences/

Published in the Proceedings of *2001 Conference of the Canadian Society for Civil Engineers, Victoria, BC.*
May 30-Jun 2, 2001. Paper C19.

ARCHITECTURE OF A DISTRIBUTED, MODEL-BASED ASSET MANAGEMENT SYSTEM

M.A. Hassanain[A], T.M. Froese[A] and D.J. Vanier[B]
[A] Department of Civil Engineering, University of British Columbia, Vancouver, BC, Canada
[B] Institute for Research in Construction, National Research Council Canada, Ottawa, ON, Canada

ABSTRACT: This paper discusses the required functionality and architecture for a distributed, model-based, integrated asset-management prototype system. The prototype serves to evaluate and test a proposed generic framework for integrating the maintenance management of built-assets. The paper presents the set of components for the integrated system and introduces the Jigsaw system, which provides a specific implementation of a general reference architecture described in the paper.

1. INTRODUCTION

With the case often stated that the construction industry is highly fragmented, the authors argue that the asset management industry shares this same characteristic. Reasoning to support this argument stems from the observation that the asset management industry is witnessing a proliferation of information technology (IT) tools (Vanier 2001), mainly relational databases, each capable of providing standalone solutions to a multitude of problem areas. This proliferation of IT tools is, however, leading to large volumes of loosely-structured data with poor interoperability.

The industry has become aware of data model standards as a way of representing technical and administrative information content, leading to the development of data structures that allow information to be exchanged among various computer applications. A requirement to achieve integration is conformance to some degree of standardization of the information representation approaches such as ISO International Standard 10303, STEP, or the Industry Foundation Classes (IFC's) developed within the International Alliance for Interoperability (IAI).

This paper reports on an ongoing research project within the Construction Engineering and Management Group, Department of Civil Engineering of the University of British Columbia. The project, integrated systems for asset management, discusses research in the field of data and knowledge integration through shared project models among participants in a maintenance project. The research has proposed a framework, which sets out policy guidelines for the conduct of maintenance in an organization (Hassanain et al. 2001). The framework is generic, meaning that the activities involved can be applied to non-specific assets, rather than to a specific asset type. It can be applied to both the level of individual projects, or to a network of projects. The research also focuses on the development of project models, a combination of product and process views of the product being maintained. The research has chosen to adopt a similar methodology to that of the IAI to develop data standards in the form of IFC's (IAI 1999). Finally, the research implements a prototype application to provide proof-of-concept to integrated maintenance management, which is the scope of this paper.

2. GENERIC ASSET MANAGEMENT MODEL

The generic framework model for maintenance management of built assets is described schematically as an $IDEF_0$ process model, shown in Figure 1 (boxes represent tasks while arrows from the left, right, top and bottom represent inputs, outputs, controls, and mechanisms, respectively). A process model describes the tasks that need to be undertaken. It illustrates how and what information needs to be communicated between tasks (Federal 1993). The framework model consists of five sequential processes. It starts with carrying out an inventory of all assets requiring maintenance during their service life and ends with scheduling maintenance operations. For each of the processes, the authors have defined a number of supporting activities, with their logical sequences and information requirements.

Formulation of the framework has led to an iterative development process of data models for representing the information that will be exchanged among all participants in a facilities maintenance management project. The research has generated a set of data standards (Hassanain et al. 2000) that consider existing standards and representations in IFCs Release 2X and recommend a number of extensions for inclusion in Release 3.0. However, these will not be discussed further in this paper since the focus here is on the required functionality and the architecture of the prototype system.

3. ROOFING SYSTEMS, AS AN APPLICATION DOMAIN

Roofing systems, specially flat, or low-slope conventional assemblies were chosen, in our project, as an application domain to demonstrate the applicability of the generic proposed framework of asset management (Hassanain et al. 1999) Roofing systems represent a type of building products that can be treated as a type of asset.

4. ASSET MANAGEMENT TOOL (AMT), A PROTOTYPE INTEGRATED APPLICATION

The main objective of developing the prototype AMT is to support asset managers through providing a solution pertaining to a multitude of functional areas through software interoperability, rather than providing enhanced features to already commercially available software.

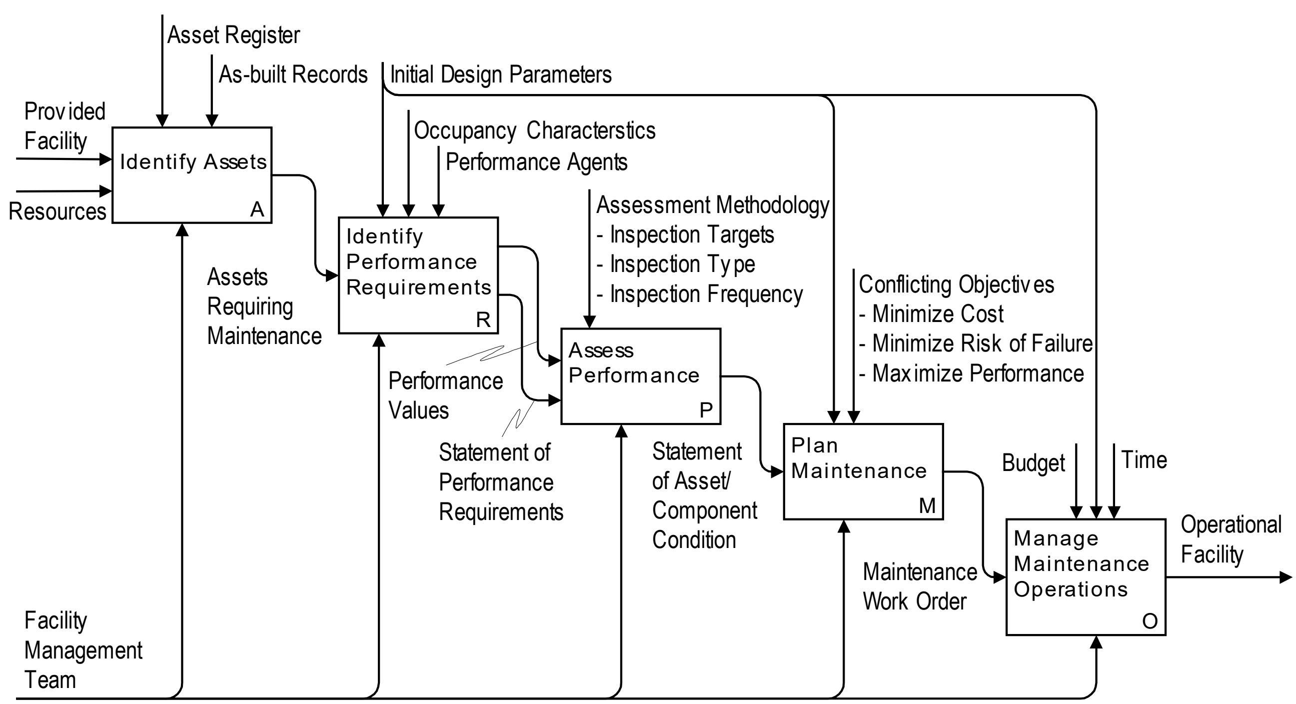

Figure 1: General Processes involved in facilities maintenance management model (Hassanain et al. 2000)

4.1 Required Functionality

The capabilities required of the AMT prototype system are outlined by the following management processes, as gleaned from the framework illustrated in Figure 1:

1. Asset Inventory

- Build up lists of assets within one project file.
- Link assets to products such as buildings and building elements.
- Record generic asset information.
- Drill-down from asset to its particular components.
- Record information pertaining to products treated as assets.

2. Asset Performance Requirements

- Specify applicability of performance requirements to either spaces or building elements (assets) within buildings.
- Specify indicators within each group of performance requirements.
- Specify upper and lower bounded values for performance indicators.

3. Asset Condition Assessment

- Specify inspection technique to be followed.
- Specify inspection frequency.
- Record attributes of defects found that impact the performance of assets.
- Determine existing condition of assets relative to pre-set performance requirements.
- Recommend a maintenance/repair/renewal action based on existing condition.

4. Strategic Maintenance Planning

- Estimate cost of labor, equipment, and material needed to perform maintenance/repair/renewal (MRR) action.
- Specify asset probability of failure if no MRR action was carried out.
- Specify consequences of asset failure (dollar value) if no MRR action was carried out.
- Specify remaining service life of assets if no MRR action was carried out.
- Prioritize MRR actions based on the parameter identified above.
- Create maintenance work request.
- Identify MRR backlog awaiting completion.

5. Maintenance Operations Management

- Specify attributes of work requested.
- Specify MRR work method.
- Breakdown MRR work into set of activities.
- Specify precedence relationship between activities.
- Specify MRR activity duration.
- Produce a schedule of activities.
- Report completed work requests.

4.2 Prototype System Architecture

Figure 2 illustrates a typical set of software components that can be used to describe the architecture for the AMT. These components describe a reference architecture for distributed, model-based, integrated systems for AEC/FM (Froese et al. 2000). The architecture is organized into three logical tiers as outlined below:

1. The application or presentation tier: contains application programs and related user-centric components.

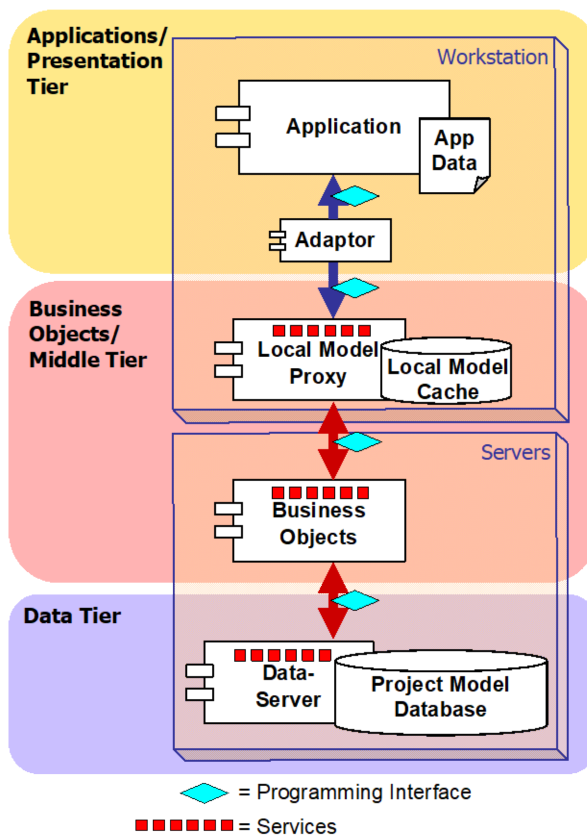

Figure 2: A reference architecture for a distributed, model-based, integrated system (Froese et al. 2000)

2. The business objects or middle tier: brings the data and services of the distributed system to the local applications and implements business logic processes.

3. The data tier: handles the persistence of project model data.

A configuration where these logical tiers may reside consists of two or more physical computer systems, a user's workstation and one or more central servers.

4.3 System Components

A typical tiered system architecture includes the following components:

- **Applications**

Providing the required functionalities outlined earlier in section 4.1, the applications component will include a specifically designed, custom-built application for the management of roofing systems. The application will have the ability to link to Microsoft Project and MicroRoofer (a roofing condition assessment tool), both typical legacy applications. We are using Microsoft Visual Basic 6.0 for implementing the design of our application. Both custom-built and legacy system applications will maintain their own data sets in addition to the information shared through the integrated system.

- **Adaptors**

Adapters are pieces of code that serve to map application schema to the common data schema. This mapping is carried out by an application-specific adaptor software component.

- **Local Model Proxies**

Local model proxies are software components on user's workstations. Applications and adaptors access a local version of the shared data that acts as a proxy for remote data sources. These components expose the distributed information services to client applications, and handle the communication of the local data with the distributed servers.

- **Business object components**

The Jigsaw Modeling Tool (developed within the Construction Engineering and Management Group at the University of British Columbia) is a utility for creating or translating the developed IFC data models (Schemas). Since it can import and export various forms of models (i.e., models with different meta-models), it acts as a "meta-meta model". One export capability that has been built into this tool is the ability to generate software classes for all of the objects defined within the data model. This software code, then, forms the foundation to creating custom business objects for a specific domain (in this case, roofing and asset management).

Data components

While there can be various centralized or distributed data repository alternatives, a typical configuration involves a central database and data server component that interacts with model proxy components across the Internet/Intranet connection.

4.4 The Jigsaw System

The Jigsaw system implements the ideas of the general reference architecture by creating a system in which a wide variety of applications (data clients) can communicate with a wide variety of data sources (data sources) through a standard application programming interface (API), so that the data clients need not know the details of the individual data servers and vise-versa. The standardized interface uses as many "industry standard" elements as possible, namely, XML and the XML DOM for the "data content" API, and the IFC's for much of the data content schema. It adds some custom elements for the data access and control API, and some extensions to the IFC schemas. The Jigsaw data server API is called JsServerDOM. The JsServerDOM is an “abstract” component. It is not intended to be implemented, but it defines the interface for all Jigsaw data server components.

As Figure 3 illustrates, in a typical situation, a data client may work with its own application data, but it uses the Jigsaw interface for all interaction with other applications or data sources. The Jigsaw data server component provides a "wrapper" around some other form of data source, adapting it to the JsServerDOM API. Thus, the AMT acts as both a Jigsaw server and client. As an example of an application adaptor, JsMSProjectAdaptor adapts Microsoft Project to act as a Jigsaw data server.

4.5 Interaction with other components

It is envisaged that the prototype would:

1. Import product models from CAD systems.

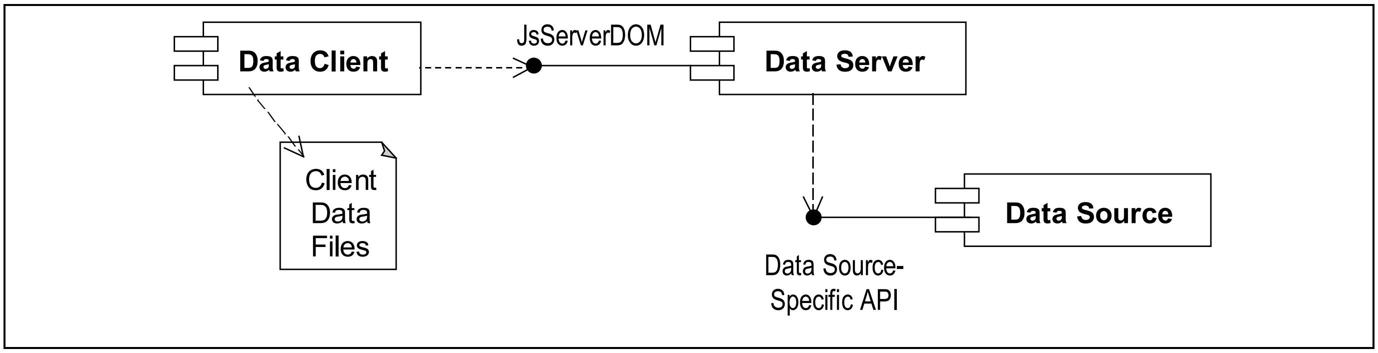

Figure 3: Jigsaw system client-server interaction

2. Import Roof section and condition assessment tables from other applications such as MicroRoofer.

3. Import/export with estimating and scheduling applications.

4. Generally be capable of exchanging all information with any other systems that work with similar information through standard data models and standard distributed system architecture.

This prototype demonstrates a direction for the implementation of IFC's in a distributed, model-based maintenance-management system. It represents one components of our overall research program into integrated computer-based tools that we call "total-project systems" (TOPS) for Architectural/ Engineering/Construction, and Facilities Management (AEC/FM) (Froese et al. 1997). TOPS have the objective of integrating construction and facilities management applications, through implementing the developed industry standards, the IFCs, hence sharing and exchanging information among software applications. Within TOPS, each application has its own data models, thus capturing related information to the specific application.

Through implementing these data models in an object-oriented system, and delivering the content of information over the internet and/or intranets using Extensible Markup Language (XML) datasets, information from each application becomes accessible to other applications within TOPS. Tagging XML data allows a more defined and accurate way to search for particular data. XML enabled documents use semantic markup that identifies data elements according to what they are, rather than how they should appear. As a result, many applications can make use of the information in XML documents.

5. CONCLUSIONS

This paper presents the architecture of a distributed, model-based, integrated asset management system. It addresses the required functionality of the Asset Management Tool prototype system, and presents the set of components for the integrated system, as well as introduces the Jigsaw system, a newly developed system that implements the ideas of the general reference architecture.

6. ACKNOWLEDGMENT

We gratefully acknowledge support for this work by the Natural Sciences and Engineering Research Council of Canada, National Research Council of Canada, and Public Works and Government Services of Canada.

7. REFERENCES

Federal Information Processing Standards 183. (1993) Integration Definition for Function Modeling (IDEF0), *United States National Institute of Standards and Technology (NIST)*, Computer Systems Laboratory, Gaithersburg, MD., USA.

Froese, T., Rankin, J., Yu, K. (1997) Project Management Application Models and Computer-Assisted Construction Planning in Total Project Systems, *International Journal of Construction Information Technology, Special Issues on Integrated Environments*, 5(1): 39-62.

Froese, T., Yu, K., Liston, K., and Fischer, M. (2000) System Architecture for AEC Interoperability, *CIB-W78: International Conference on Construction*

Hassanain, M.A., Froese, T.M., and Vanier, D.J. (1999) Information Analysis for Roofing Systems Maintenance Management Integrated System, *8th International Conference on Durability of Building Components and Materials*, Vancouver, Canada, 2677-2687.

Hassanain, M.A., Froese, T.M., and Vanier, D.J. (2000) IFC-based Data Model for Integrated Maintenance Management. *Proc. 8th International Conference on Computing and Building Engineering*, ASCE, California, USA, 1: 796-803.

Hassanain, M.A., Froese, T.M., and Vanier, D.J. (2001) Development of a Maintenance Management Model Based on IAI Standards, *Journal of Artificial Intelligence in Engineering*, Elsevier Science. Accepted for publication.

International Alliance of Interoperability (IAI). (1999). Specifications Development Guide. Industry Foundation Classes – Release 2.0, March.

International Alliance of Interoperability (IAI). (2000). Home page of UK Chapter [on line], Available from: http://www.iai.org.uk

Vanier, D.J. (2001) Why Industry Needs Asset Management Tools, *Journal of Computing in Civil Engineering*, 15(1): 35-43.