# A PROTOTYPE DISTRIBUTED CIC SYSTEM BASED ON IAI STANDARDS

A Prototype Distributed CIC System

A.L. GORLICK and T.M. FROESE
University of British Columbia, Vancouver, BC, Canada

Durability of Building Materials and Components 8. (1999) *Edited by M.A. Lacasse and D.J. Vanier*. Institute for Research in Construction, Ottawa ON, K1A 0R6, Canada, pp. 2171-2179.


## Abstract

A prototype Computer Integrated Construction system is being developed that models building product and process information using International Alliance for Interoperability standards. The goal of this research is to provide a window into the future of how these standards can be applied in the construction industry. The prototype consists of a project database that is structured according to a common project schema or project data model. The schema is based on emerging International Alliance for Interoperability standard models but it is implemented in a way that allows the dynamic development of the schema (and even of its underlying metamodel) without destroying the information in the database in order to support on-going work in the development of information model standards. The system is modular in nature so that it can be supplemented with plug-in tools to accomplish a variety of project management tasks. It is served over the web through a combination of Microsoft's Active-X Data Object technology and a lightweight version of ISO STEP's Standard Data Access Interface. Data sets served to the client are wrapped in the Extensible Markup Language to allow for the self-description of information.

Keywords: Computer Integrated Construction, IAI, IFC, Product Models, Process Models

## 1 Introduction

The essence of planning is to combine all aspects into one feasible process, which means that information has to be transmitted, transformed and combined. The increased complexity of buildings and of the organization of the construction process has made the transmission and sharing of information more difficult as there is a growing amount of information to be consolidated, distributed, and exchanged

(Jägbeck 1998). Unfortunately, most of the software tools used to generate this information cannot interoperate. In practice, paper drawings or documents are therefore the common medium for exchanging information. Consequently, the manual input of data from one tool to another and the associated risk of error dramatically hamper the duration, quality and cost of the construction process (Debras et. al. 1998). The result is a need for more explicit and self-contained (i.e. less context dependent) information (Jägbeck 1998). The construction management group at the University of British Columbia is addressing this issue through the development of new computer-based tools to support construction management and to increase the overall efficiency of construction operations.

We subscribe to the vision that the architecture, engineering, and construction (AEC) industry will evolve to one in which detailed, integrated, computer-based models of the AEC projects act as the predominate medium for carrying out design and management work and for communicating the results among project participants (Froese et. al. 1997). To this end, the Construction Management Group has developed the TOPS (Total Project Systems) model. TOPS is a conceptual model that provides the basis for the development of Computer Integrated Construction (CIC) systems. The TOPS model is founded on three principal research thrusts: application development, shared project representations, and system architectures and interfaces (Russell and Froese 1997). Much work has been done on application development (see Rankin et. al. 1998; Froese et. al. 1997; and Rankin et. al. 1997 for examples) and shared project representations (see Froese and Rankin 1998; Russell and Froese 1997; Yu et. al.1997; Froese 1996; and Fischer and Froese 1996 for examples). Current research is focussing on the design, development, and prototyping of a system architecture and interface(s) for use in TOPS.

TIP (TOPS Implementation Prototype), the prototype CIC system under development, models building product and process information using IAI (International Alliance for Interoperability) standards. The on-going goal of this research is to provide a window into the future of how IAI standards can be applied in the construction industry. The prototype consists of a project database that is structured according to a common project schema or project data model. The schema is based on emerging IAI standard models, but it is implemented in a way that allows the dynamic development of the schema (and even of its underlying metamodel) without destroying information in the database in order to support on-going work in developing project information model standards. The system is modular in nature so that it can be supplemented with plug-in tools to accomplish a variety of project management tasks. Current plans include the development of a short cycle scheduling tool. TIP is a distributed system, served over the World Wide Web through Microsoft's Active-X Data Object (ADO) technology wrapped in a data access layer that we are developing called the Lightweight Data Access Interface (LDAI), which can be thought of as a simplified and flexible development version of the ISO STEP Standard Data Access Interface, SDAI. Data sets served to the client are encoded in XML (Extensible Markup Language) to allow for self-description of information.

## 2 TIP Architecture

Figure 1 illustrates the architecture of the TIP system. Froese et. al. (1997) characterize the underlying architecture of this class of system as: *open,* in that it is not dependent on specific computing technologies (e.g., it is based on international data standards, it is platform independent, etc.); *modular,* such that a variety of specific applications developed by different sources can be brought to bear as appropriate to create the overall system; and *distributed,* by recognizing the requirements of a variety of users and data sources.

Application modules are connected to the system using standard interfaces. Thus, they can be updated without the need for changes to other parts of the system. Not only is project data shared, the application modules are also shared by different participants at different locations. As the applications and project data are transparent, users must simply send requests to the system through the user interface, invoke appropriate modules, and input the appropriate data. (Froese et. al. 1997)

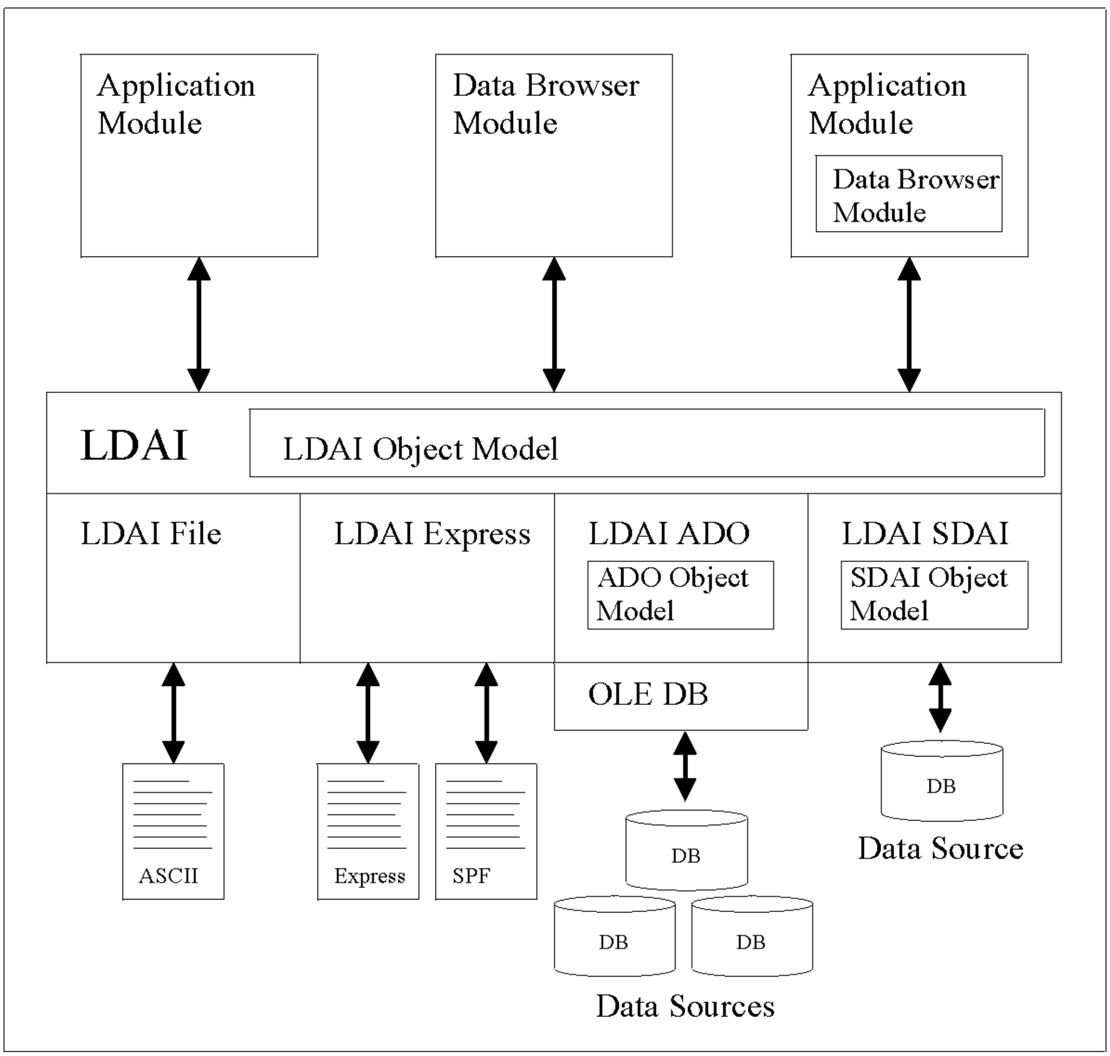


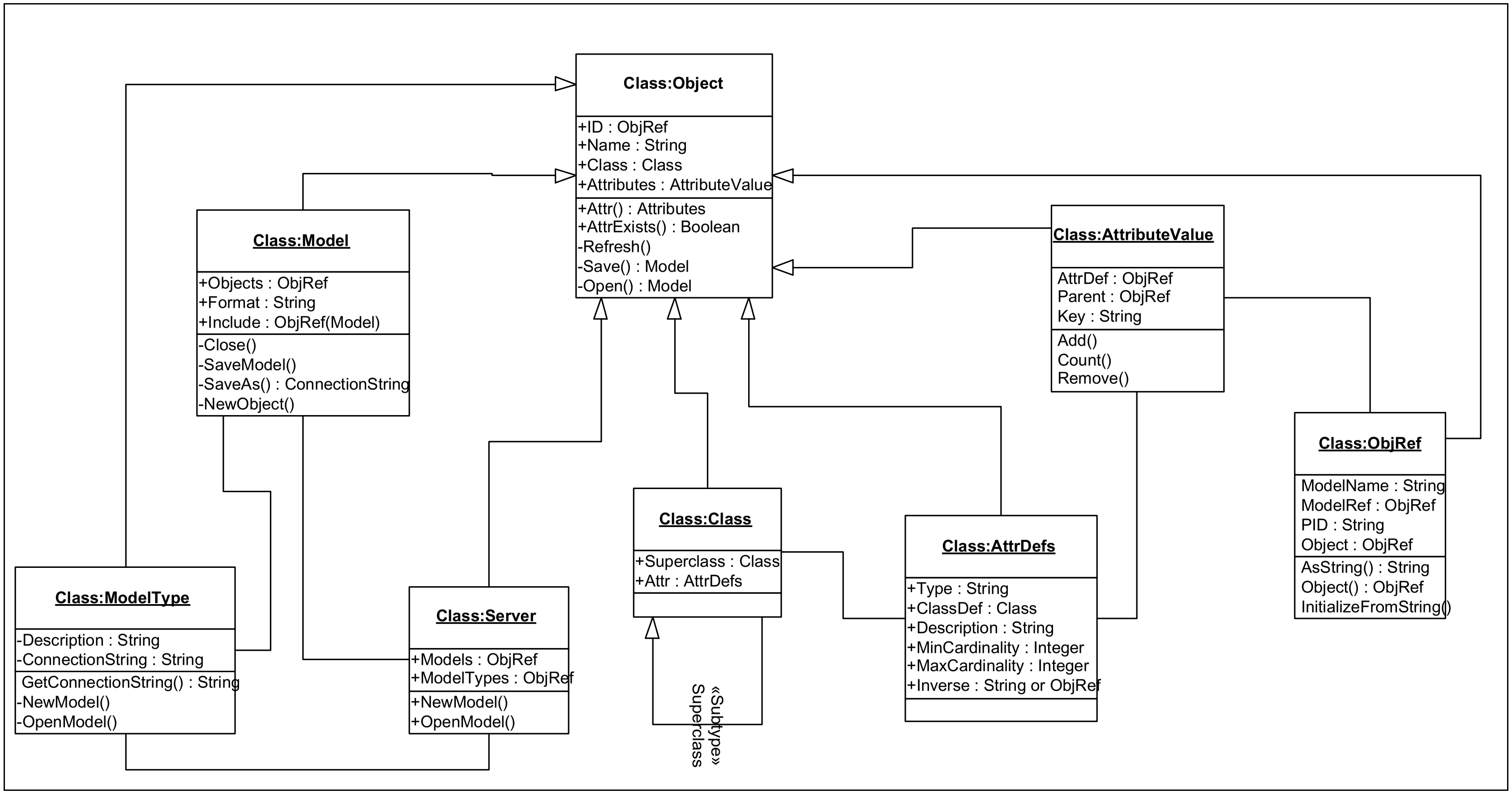

**Fig. 1: TIP architecture**

## 3 LDAI

Applications interact with the integrated system through data access interfaces that are structured according to underlying schemas or data models (in this case, the LDAI object model), which are in turn structured on underlying assumptions about the structure of primitive concepts such as objects, classes, etc. (i.e., the metamodel). The LDAI (lightweight data access interface) defines a functional interface between an application and one or more collections of data. The LDAI is a data access, or "middleware", component. It is similar to other data access interface standards such as the ISO STEP Standard Data Access Interface, SDAI (see International Organization for Standardization (ISO) 1995 for an explanation of the SDAI). We are developing the LDAI, rather than directly adopting the SDAI or another system, because our requirements for a middleware layer are quite different, stemming largely from the fact that we wish to support rapid prototype development in an area where "standards" are frequently changing, rather than support production software. The LDAI, then, has the following defining characteristics:

- It is very simple to work with. It is implemented in Visual Basic and it provides only the most essential and generic data access functionality (it can be thought of as a stripped-down version of the SDAI).
- It can be used as a "front end" interface for a wide variety of data access "back ends", such as EXPRESS, STEP Part 21, or XML files, or SDAI or Microsoft ADO data access interfaces (and it can act as a translator between these different formats).
- Only the smallest possible set of core metamodel data structures are statically encoded in the LDAI; other aspects of the metamodel and all project data models are dynamically loaded at run-time (i.e. late binding) and these models can be altered at any time.
- The LDAI object model is presented in UML notation in Figure 2. We are using UML (see UML Resource Center 1999) notation for developing the LDAI since it is becoming a standard within the software industry for representing data models and other portions of software designs and because we have found more support for UML within development tools than for EXPRESS. We expect to provide the LDAI with the capability of importing and exporting EXPRESS and possibly UML information as well.

The TIP metamodel is drawn directly from this LDAI object model. An important function of this model is its ability to evolve its schema. The main reason for this in our case is that we are developing prototype integrated applications based on standard project data models concurrent with the ongoing development of the standard data models themselves, so the applications' underlying data structures are frequently changing. This capability can also be important for end users to allow the underlying conceptual model used by the project actors to be altered and to evolve over time (by adding, deleting, or modifying an object type property), without affecting the overall consistency of the project information base. (Rezgui, et. al. 1998) The LDAI object model implements schematic evolution by early binding only the smallest possible metamodel structure. All other metamodel constructs are




**. 2: LDAI object model**

built up dynamically as necessary at runtime, and changes can be made to the schema even after it has instance data associated with it. Although the LDAI will be able to maintain data consistency for most types of simple changes to existing schemas, it leaves the ultimate responsibility for maintaining data integrity during schematic evolution to the application developer and users.

It is important to note that LDAI is not intended to provide a new standard for functional interfaces between data and applications, nor is it a replacement for the SDAI. The LDAI is a development interface rather than a production interface. It is advanced as a simple and flexible tool to facilitate the quick development of prototype systems. In a full implementation of TIP, the LDAI would be replaced with an SDAI or other data access interface.

## 4 ADO

The following is excerpted from Microsoft's white paper on Universal Data Access (UDA) (Microsoft 1998). ADO is a part of Microsoft's UDA initiative. The premise of UDA is to allow applications to efficiently access data where it resides without replication, transformation, or conversion. Open interfaces allow connectivity among all data sources. Independent services provide for distributed queries, caching, update, data marshalling, distributed transactions, and context indexing among sources.

ADO is a set of high-level automation interfaces over OLE DB (OLE DB is the Microsoft system-level programming interface to diverse data sources). It enables applications to access and manipulate data in a database server through an OLE DB provider. ADO is provider neutral. That is, although it allows access to the full range of OLE DB data, it does not implement its features specific to any one provider or data source. Underlying ADO are four service components that implement its functionality. These services are outlined in Table 1.

**Table 1: ADO services**

| Service | Description |
|---|---|
| Cursor | Data fetching, data manipulation, local updates, local cursors |
| Synchronization | Propagates updates to the data source, refreshes the cursor from the data source, and coordinates conflicting updates |
| Shape | Provides a hierarchical view of data to offer a flexible view of the data set |
| Remote Data Service | Permits interaction with OLE DB data sources over the Internet |

ADO will be wrapped in LDAI. The result is that the functionality of ADO, and the ADO service components, will be transparent to the user.

## 5 Implementation

The software architecture of TIP was presented in Figure 2. The data repository system will be implemented as a relational database in Microsoft's Access 97. Schemas stored in the database will be IAI models with possible extensions for application-specific functionality. An OLE DB server will be used to serve the database, making it accessible to the Internet. The TIP system will implement the LDAI with four backends: LDAI text files; LDAI Express; LDAI ADO; and, LDAI SDAI. The LDAI File and LDAI Express backends read and write local files. LDAI text file reads and writes LDAI notation in ASCII format. The LDAI Express backend imports or exports Express and SPF (STEP Physical File format) type files. The LDAI ADO backend is the interface that allows the LDAI to use ADO to communicate with data sources. Finally, the LDAI SDAI backend is an interface allowing information to migrate between the LDAI and the SDAI. Of these four backends, current plans are to only implement the first three. The LDAI SDAI backend will be implemented later.

At the application level, three classifications of modules are shown. The data browser is a general purpose user interface to project data that is an integral component of TIP. This data browser can serve as a separate tool or application, similar to that of the Explorer tool in Windows. Its purpose is to allow users to enter, view, and manipulate project data as well as view and modify the underlying models and schemas. The data browser interface can also be incorporated into other, more specialized applications (as shown by the combined Application Module/Data Browser Module element). Finally, the generic application represents any standalone module that plugs into the LDAI to access project information. A variety of applications can be developed—estimating, scheduling, quality control, or document management systems for example—so long as the required underlying schemas have been installed to support the application module.

At present, a short cycle scheduling application is being prototyped. The application is designed for use by construction managers to create on-site look-ahead schedules and to schedule work-in-place activities and materials delivery. A scheduling application was selected for development purposes as scheduling draws upon a broad range of project information (i.e., product, process, and resource information). For example, activities, sequencing, cost, the assignments of resources to processes, etc. As a result, we expect this application to provide a good test of emerging IAI models of construction process-related information. Future papers will report on the implementation of the scheduling application and the performance of the IAI models.

The language binding for TIP is Visual Basic. The data browser and application modules are being developed in a standard web browser environment (Microsoft's Internet Explorer) in order to take advantage of the functionality of ADO. A separate data repository is being created to store XML code. ADO will be used to wrap project data in XML to create a data set that is self-describing.

## 6 Conclusion

This paper has described the continuing development of TOPS by the Construction Management Group at the University of British Columbia. Current research focuses on migrating the TOPS conceptual models to a prototype implementation, TIP. The TIP model addresses the three research thrusts of TOPS: application development, shared project representations, and system architectures and interfaces. The LDAI was introduced as a prototyping mechanism that allows the dynamic modification of the underlying schemas and metamodels without affecting instantiated project data. Prototyping of the system, and a short cycle scheduling application module, is underway and testing is expected to begin in the near future. Results from the complete prototype implementation and field trials will be reported in future papers.

## 7 References


Debras, P., Monceyron, J., Bauer, F., Ballestra, P., and Rocca, F. (1998) From Product Data Technologies to Applications: illustrative cases in the AEC domain. *Proceedings of the CIB Working Commission W78*, pp.163-170.

Fischer, M., and Froese, T. (1996) Examples and Characteristics of Shared Project Models. *ASCE Journal of Computing in Civil Engineering, Special section on Data, Product, and Process Modeling,* Vol. 10, No. 3, pp. 174-182.

Froese, T. (1996) Models of Construction Process Information. *ASCE Journal of Computing in Civil Engineering, Special section on Data, Product, and Process Modeling,* Vol. 10, No. 3, pp. 183-193.

Froese, T., Grobler, F., Yu, K. (1998) Development of Data Standards for Construction – An IAI Perspective. *Proceedings of the CIB Working Commission W78,* pp.395-406.

Froese, T., and Rankin, J. (1998) Representation of Construction Methods in Total Project Systems. submitted to the 1998 *ASCE Congress on Computing in Civil Engineering*, 1998 February 2

Froese, T., Rankin, J., and Yu, K. (1997) Project Management Application Models and Computer-Assisted Construction Planning in Total Project Systems. *The International Journal of Construction Information Technology*, Vol. 5, No. 1, pp. 39-62.

International Organization for Standardization (ISO) (1995) Industrial Automation Systems and Integration – Product Data Representation and Exchange Part 22. Standard Data Access Interface. *ISO 10303-22.*

Jägbeck, A. (1998) IT Support for Construction Planning. *Ph.D. Thesis*, Royal Institute of Technology, Stockholm, Sweden.

Microsoft (1998) OLE DB/ADO: Making Universal Data Access a Reality. [online] *http://www.microsoft.com/data/reference/wp2/sigmod98_0sxl.htm,* [accessed 1998 November 26]

Rankin, J., Froese, T., and Waugh, L. (1998) The Functionality of Computer-Assisted Construction Planning. submitted for the *CSCE Annual Conference*, 1998 February 27.

Rankin, J., Froese, T., and Waugh, L. (1997) Computer Assisted Construction Planning (CACP) in the Context of Total Project Systems (TOPS). *Proceedings of the 1997 Conference of the Canadian Society for Civil Engineers, Sherbrooke, Quebec, May 27-30, 1997*. Vol. 2, pp. 2-41 to 2-50.

Rezgui, Y., Cooper, G., and Brandon, P. (1998) Information Management in a Collaborative Multiactor Environment: The COMMIT Approach. *ASCE Journal of Computing in Civil Engineering,* Vol. 12, No. 3, pp. 136-144.

Russell, A., and Froese, T. (1997) Challenges and a Vision for Computer-Integrated Management Systems for Medium-Sized Contractors. *Canadian Journal of Civil Engineering*, Vol. 24, No. 2, pp.180-190.

UML Resource Center (1999) [online] *http://www.rational.com/uml,* [accessed 1999 February 10]

Yu, K., Froese, T., and Vinet, B. (1997) Facilities Management Core Models. *Proceedings of the 1997 Conference of the Canadian Society for Civil Engineers, Sherbrooke, Quebec, May 27-30, 1997*. Vol. 2, pp. 2-195 to 2-204