

BCOF: AN INTEGRATED OBJECT FRAMEWORK FOR REALIZATION OF ISO BUILDING CONSTRUCTION CORE MODEL

Chien-Cheng Chou, Shang-Hsien Hsieh,* Ming-Teh Wang

Department of Civil Engineering, National Taiwan University, Taipei, Taiwan, R.O.C.

*Corresponding author; E-mail address: shhsieh@ce.ntu.edu.tw

Abstract: An integrated object framework, called BCOF (Building Construction Object Framework), for realization of ISO Building Construction Core Model (ISO BCCM or ISO Part106) is proposed in this paper to address the multidisciplinary problems of information integration in the AEC (Architecture, Engineering, and Construction) industry. Discussions are given first on both the issues related to information integration in the AEC industry and the design considerations of BCOF. Then, the preliminary design and implementation of BCOF are presented.

Keywords: AEC Industry, Building Constitution, STEP, SDAI, Object framework.

1. INTRODUCTION

Many project models have been designed and proposed in the literature for solving the multidisciplinary problems of information integration in the AEC (Architecture, Engineering, and Construction) industry [1-4]. The two most important international standards for integrating all AEC information models are the ISO STEP (STandard for the Exchange of Product model data) [2,5] and IFC (Industry Foundation Classes) [6,7], defined by an international organization called IAI (International Alliance for Interoperability). To promote exchange and sharing of information in the AEC industry, IFC defines a set of standards similar to those AEC-related standards (e.g., Part106, AP225, AP228, and AP230) of ISO STEP. Although IFC addresses more on the software techniques than ISO STEP, both of them focus only on the standardization of data interfaces without consideration of integrating actual software systems needed for their realization. This work proposes an integrated object framework, called BCOF (Building Construction Object Framework), for realization of ISO Building Construction Core Model (ISO BCCM or ISO Part106) using pre-defined software components.

Currently, COMMIT (Construction Modeling and Methodologies for Intelligent integration of information) [8-11] and OPIS [3] are two research efforts very similar to BCOF. Both of them can also be treated as object frameworks. The major difference between BCOF and them is on the mechanism of how to integrate legacy AEC systems. COMMIT has little support for the integration mechanism, while OPIS can integrate legacy AEC

systems in the front tier (i.e., legacy CAD systems can be connected through its front-end GUI). Both of them use only one specific Object-Oriented Database (OODB) to store information. Therefore, existing legacy database systems cannot be easily reused and incorporated into the COMMIT or OPIS environment for development of AEC information management systems. In this work, BCOF is designed to accommodate different databases and to facilitate the integration of legacy AEC systems as well as new ISO STEP systems within BCOF through a SDAI-like data access interface. Furthermore, BCOF takes advantage of advanced information techniques (such as actors, components, and business object frameworks) to achieve high reusability and integration.

In the remainder of this paper, discussions are given first on the problems of information integration in the AEC industry as well as the design considerations of BCOF. Then, the preliminary design and implementation of BCOF are presented. The emphasis is placed on the design of BCOF to facilitate the following two tasks: (1) integration of international standards (especially ISO STEP Part106 and AP225) and domestic industry standards; (2) integration of legacy AEC systems in the BCOF environment. Finally, some conclusions from this work are drawn.

2. INTEGRATION ISSUES AND DESIGN CONSIDERATIONS OF BCOF

This section discusses the issues related to the integration of international and domestic AEC multidisciplinary standards, the integration of legacy

AEC systems, and the integration of product data and business rules. In addition, the design considerations of BCOF are described.

Integration of International and Domestic Industry Standards

The object framework technique can be used to solve the integration problems for information systems (IS) in a domestic industry [12]. For example, a domestic AEC information model and some pre-defined components (i.e., domestic industry standard) can be provided using the object framework technique to enforce various AEC programs within the same software architecture. However, when the problem becomes integration of IS between a domestic industry standard and an ISO international standard, some integration rules are needed to enable the integration of objects in the different frameworks of standards.

Currently, ISO STEP provides the Application Reference Model (ARM) for exchange of information among its parts in different hierarchical levels. For example, the ARM of AP225 (application protocol for building elements using explicit shape representation) can map AP225 objects to AP203 objects and AP228 (for Heating, Ventilation, and Air Conditioning, HVAC) can then reuse the information provided by AP225. All of these APs make references to Part106 (BCCM). However, Part106 does not have an explicit ARM at this moment. Furthermore, even a complete Part106 ARM would not provide mechanism of mapping Part106 objects to the objects in a domestic industry standard, which are the basic building blocks of a typical domestic AEC system. Therefore, some integration rules, which are not addressed by STEP, should be developed for integrating the international and domestic AEC industry standards.

According to the definition of Part106, three kinds of information objects can be classified as follows [13]:

1. Information objects independent of any AEC domain: These objects can be referenced and used in every AEC domain.
2. Information objects related to every AEC domain: These objects must be defined in every parts of an AEC standard. The integration problem becomes how to map these objects within different contexts. For example, a AP228 object can at least use the interface of Part106 to retrieve information from AP225 objects.
3. Information objects related to some AEC domains: An example of these objects are steel structure objects defined in AP230 (for building structural frame: steelwork). They are similar to those of the second kind, so the same integration rules can be used to solve the integration problem for them.
- 4.

It is possible that the information objects with the same name exist in more than two different AEC domains, e.g., the building objects (entities) of AP225 and Part106. They are designed for different purposes and, therefore, have different data structures. From the perspective of AEC processes, it is rational to have this kind of situation that objects in different AEC information domains, although representing the same real world object, manage only attributes needed in its own domain. Therefore, the solution to the integration problem is not to define a general object sharable among all AEC domains for the same real world object. Furthermore, it is often difficult, even for the domain experts, to distinguish whether two objects in different domains of standards correspond to the same real world object or only some of their properties come from the same real world object.

In this work, the following integration rules based on the principle of ARM are used in the design of BCOF:

1. For integration of two objects corresponding to the same real world object, the inheritance relationship can be used to connect the more general one to the more specific one.
2. For integration of two objects corresponding to the similar real world object, the delegation relationship can be used to connect the more general one to the more specific one.
3. No integration is needed for two objects corresponding to different real world objects.

Integration of Legacy AEC Systems

The ISO STEP community has long-term spent huge efforts to develop standards in many different industries. Consequently, STEP Part22 (Standard Data Access Interface, SDAI) is defined to provide a programming interface for information exchange among different STEP domains [2]. Actually, SDAI is a meta-object protocol, i.e., the client program can send a message with the name, method, and parameters of an object to any other object at run-time. Using the meta-object protocol often results in the best flexibility and creates a high threshold for the message broker [14].

In terms of SDAI, the SDAI-compliant driver for databases is hard to implement. But once the database supports the driver, users of the database can have great flexibility of interfacing with the database and integration of legacy systems can be made easier. To provide great flexibility for integrating legacy AEC system within the BCOF environment, a SDAI-like data access interface is designed in BCOF.

Integration of Product Data and Business Rules

The goal of ISO STEP is to provide a standard product data model, not the business object model. On the other hand, modeling business rules

and providing a flexible mechanism that can update these rules easily are the responsibility of a business object model because these business rules are highly dependent to certain company policies and change rather rapidly (which can not become an international standard). Traditionally, this problem can be solved through the use of database stored procedures. In an object-oriented system, the concept of an object can also help us to model these methods and properties within a class definition.

As many researchers have recognized, there are three levels of STEP implementation: simple file exchange through translator, on-memory database exchange using SDAI, and knowledge-based information exchange using agents [2][5]. In this work, the SDAI-like data access interface implemented in BCOF belongs to the highest-level of STEP implementation (i.e., knowledge-based information exchange using agents) and is AP-based. In addition, BCOF uses the business object with properties and methods, instead of the product data, as the building blocks for integration of product data and business rules.

The structure of a BCOF business object is shown in Fig. 1. The reasons for designing two separate objects, i.e., an entity object and a transactional object, in the business object of BCOF are discussed in more detail in [15]. The ideas are briefly described below:

1. The purposes of a transactional object are twofold: to wrap the legacy AEC program and to transfer legacy database's information into SDAI forms.
2. Both the transactional object and the entity object form the concept of a real world object. In this design, the object granularity is kept small, but the lifetime of a business object can be still long like a traditional object. The lifetime of a transactional object is short because the simple principle from the transactional processing field should be followed: "Drop any resource if possible."

In addition, The entity object has three interfaces:

1. The SDAI-like data access interface: the design of the interface is based on the principle of SDAI, which is independent of any implementation approaches [5]. Therefore, any client can use any object that supports the interface. The only constraint is that the client must obey the call sequence showed in Fig. 2. The call sequence is derived from the OODBMS principle which states that any object has three states: the initial state, the transient state, and the persistent state [15].
2. AEC-specific business service interfaces: these are the interfaces related to company policies and are used to model the business rules.
3. Interfaces for ISO STEP or domestic industry standard: these interfaces are designed for integrating various AEC standards. The

integration strategy for these interfaces has been discussed earlier in this section.

3. PRELIMINARY DESIGN AND IMPLEMENTATION OF BCOF

The preliminary design and implementation of BCOF are shown in Fig. 3. There are four object layers in BCOF along with several software components. Some of them are briefly described as follows:

- Service Object Layer: this layer is designed to provide specific services to the process object layer and the business object layer.
 1. Security Management Service: this is designed to provide log-in and log-out information, and to authenticate the validity of the client, the message, and the object.
 2. Relationship Service: if two objects of the business object layer are stored in different databases, this can provide query services.
 3. XML Document Service: this is designed to provide an XML document whose data source is the business object layer and its template is stored in the DTD and XSL files.
 4. Queue Service: this is designed to provide an asynchronous message transfer mechanism to an XML document object.
 - Business Object Layer: this is designed for providing AEC business object services.
 1. Database: there are three different databases used in BCOF currently: UniSQL for storing Part106, Express Data Manager (EDM) for storing AP203, and SQL Server for storing documents.
 2. Persistence and Version Interfaces: this is the SDAI-like data access interface. Note that the version interface uses the persistence interface to implement version control.
 - Process Objects Layer: this is designed to control the execution of an AEC process. It sends messages to many entity objects to manage the execution of an process.
 - Graphics User Interface Layer: this is designed for visualization of the final results of an AEC process.
- In addition, BCOF takes advantage of the actor, component, and object framework techniques to achieve high reusability and integration. The following described briefly how these techniques are used in the implementation of BCOF.
- Actors model the perspective users. They constitute anything that is external to the system we are to develop [16]. In this work,

the actor is the client of any BCOF component. Its purpose and functionality in BCOF are to circumscribe the boundary of a component, while BCOF must support a secured and flexible environment to the actor.

- Components are binary units of independent production, acquisition, and deployment that interact to form a functioning system [17]. In fact, a component consists of some interfaces and objects. The client may be another object, or an independent GUI program. The most important contribution of components to the object technology is the separation of interfaces and objects [18]. In BCOF, the Engineering Drawing component is implemented in C++ and VB with EDM for storing AP203 objects, and the Contract component is implemented in VB with SQL Server for storing documents. They can all communicate with the Part106 component which is implemented in C++ with UniSQL for storing construction core information.
- An object framework is a partially complete software (sub-) system that is intended to be instantiated. It defines the architecture for a family of (sub-) system and provides the basic building blocks for their creation. It also defines the place where adaptations for specific functionality should be made. In an object-oriented environment, a framework consists of abstract and concrete classes [19]. Note that BCOF can be decomposed as many components and objects, meaning that BCOF is undoubtedly an object framework that can be used to help develop AEC information systems.

4. CONCLUSIONS

The preliminary design and implementation of an integrated object framework, called BCOF (Building Construction Object Framework), has been presented in this paper. BCOF is proposed here for realization of ISO BCCM (or Part106) in order to address the multidisciplinary problems of information integration in the AEC Industry. By taking advantage of advanced information techniques, BCOF not only supports a SDAI-like data access interface to facilitate integration of legacy AEC systems and new ISO STEP systems within the BCOF environment, but also provides a highly reusable and flexible object-oriented environment for development of AEC systems.

ACKNOWLEDGMENTS

This ongoing project is supported by the National Science Council of Republic of China under Grant No. NSC-88-2211-E-002-041.

REFERENCES

- [1] Froese, T. (1996). "Models of Construction Process Information," *Journal of Computing in Civil Engineering*, ASCE, **10**(3), 183-193.
- [2] Froese, T. (1996). "STEP Data Standards and the Construction Industry," *Proceedings of the 1996 Annual Conference of the Canadian Society for Civil Engineering*, Montreal, Canada, Vol. 1, 404-415. (Also available at http://www.civil.ubc.ca/~tfroese/pubs/fro96a_step/fro96a_fn.html)
- [3] Fischer, M., and Froese, T. (1996). "Examples and Characteristics of Shared Project Models," *Journal of Computing in Civil Engineering*, ASCE, **10**(3), 174-182.
- [4] Kalay, Y.E. (1998). "Integrated Model to Support Distributed Collaborative Design of Buildings," *Automation in Construction*, **7**(2-3), 177-188.
- [5] Fowler, J. (1995) *STEP for Data Management, Exchange and Sharing*, Technology Appraisals, Great Britain, UK.
- [6] Yu, K., Froese T., Grobler, F. (1998). "International Alliance for Interoperability: IFCs.," *Proceedings of the Fifth Congress on Computing in Civil Engineering*, ASCE, 395-406.
- [7] International Alliance for Interoperability (1997). "AEC Processes Supported by IFC." *Industry Foundation Classes Release 1.0*, Specification Vol. 1, International Alliance for Interoperability (IAI).
- [8] Rezgui, Y., Cooper, G., and Brandon, P. (1998). "Information Management in a Collaborative Multiactor Environment: The COMMIT Approach," *Journal of Computing in Civil Engineering*, ASCE, **12**(3), 136-144.
- [9] Rezgui, Y., Brown, A., Cooper, G., Aouad, G., Kirkham, J., and Brandon, P. (1996). "An Integrated Framework for Evolving Construction Models," *The International Journal of Construction Information Technology*, **4**(1), 47-60. (Also available at <http://www.salford.ac.uk/iti/projects/commit/commit.html>)
- [10] Brown, A., Cooper, G., Rezgui, Y., Brandon, P., and Kirkham, K. (1996). "The Architecture and Implementation of a Distributed Computer Integrated Construction Environment," *Workshops of Construction on the Information Highway*, Bled, Slovenia. (Also available at <http://www.salford.ac.uk/iti/projects/commit/commit.html>)
- [11] Brown, A., Rezgui, Y., Cooper, G., Yip, J., and Brandon, P. (1996). "Promoting Computer Integrated Construction through the Use of Distribution Technology," *The Electronic Journal of Information Technology in Construction*. (Also available at <http://www.salford.ac.uk/iti/projects/commit/commit.html>)

[12] Brodie, M.L., and Stonebraker, M. (1996). *Migrating Legacy Systems: Gateways, Interfaces, and the Incremental Approach*. Morgan Kaufmann Inc., San Francisco, CA, USA.

[13] ISO (1996). "Product Data Representation and Exchange, Part 106: Building Construction Core Model," Committee Draft, ISO/TC184/SC4.

[14] Kiczales, G., and des Rivieres, J. (1993). *The Art of MetaObject Protocol*, MIT Press, MA, USA.

[15] Chou, C.C. (1999). "Design and Implementation of Building Construction Object Framework," M.S. Thesis, Department of Civil

Engineering, National Taiwan University, Taipei, Taiwan, R.O.C. (in preparation).

[16] Booch, G., Rumbaugh, J., and Jacobson, I. (1998). *The Unified Modeling Language: User Guide*, Addison-Wesley, Reading, MA, USA.

[17] Szyperski, C. (1998). *Component Software*, ACM Press.

[18] Laddaga, R., and Veitch, J. (1997). "Dynamic Object Technology," *Communications of the ACM*, 40(5), 36-65.

[19] OMG. Object Management Group, <http://www.omg.org>.

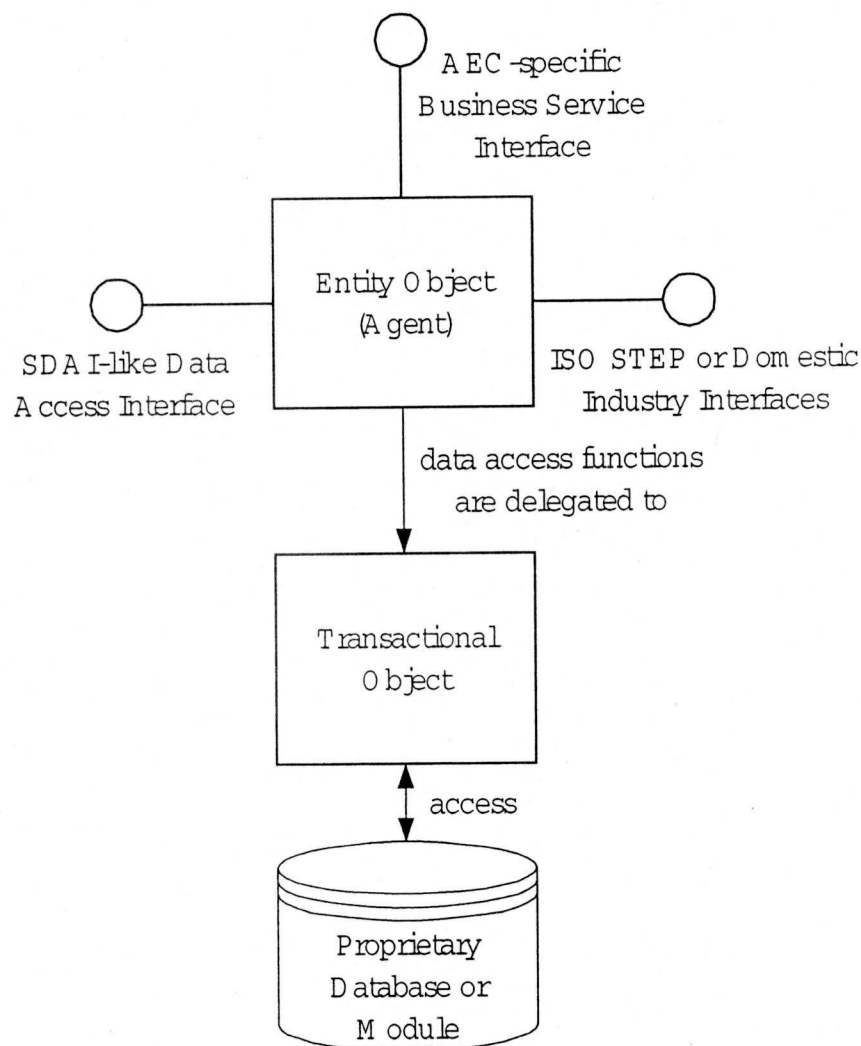


Figure 1. The structure of the business object in BCOF

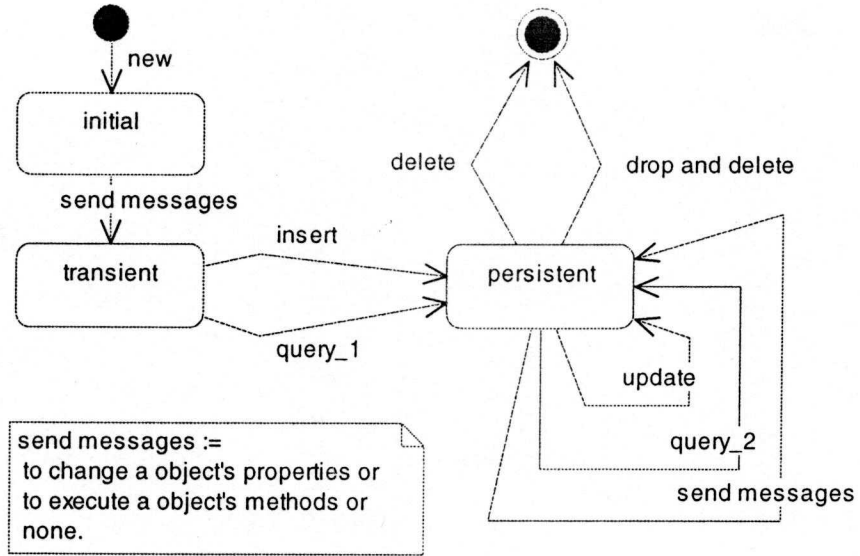


Figure 2. The state diagram of a object that supports the SDAI-like data access interface

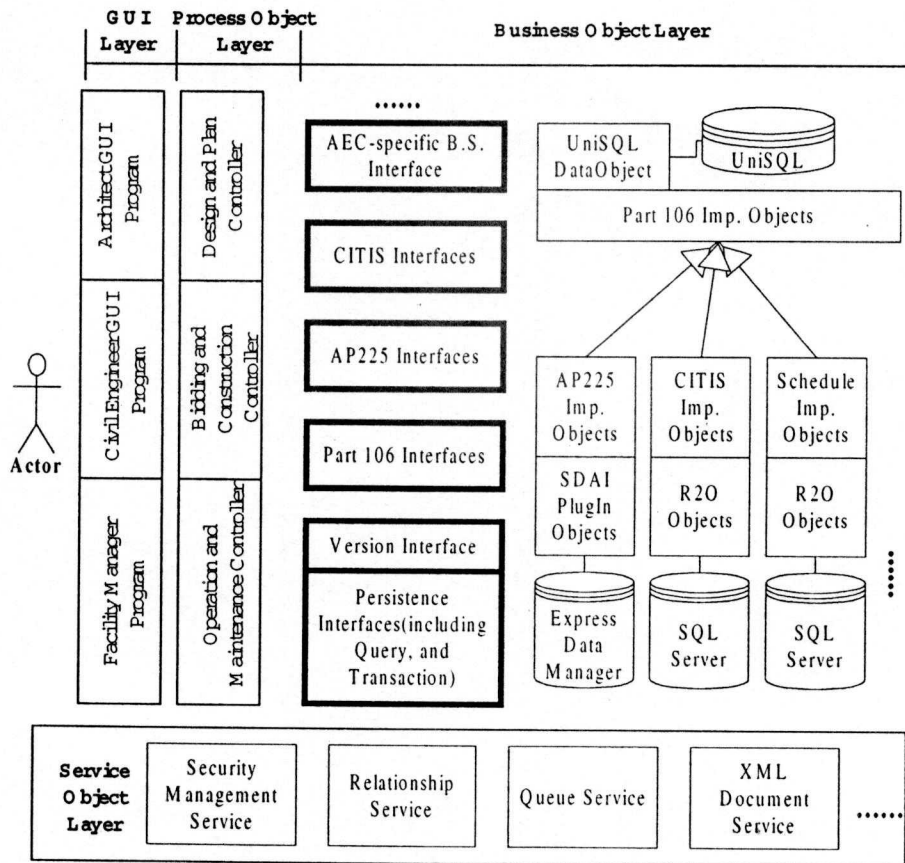


Figure 3. The object layers of BCOF