Presented at the 17th International Conference on Computing in Civil and Building Engineering, June 22-24, 2018
http://programme.exordo.com/icccbe2018/delegates/presentation/334/

An Integrated Building Information Model (iBIM) for Design of Reinforced Concrete Structures

Jamal A. Abdalla[1] and Tilal M. Eltayeb[2]

1) Ph.D., Prof., Department of Civil Engineering, American University of Sharjah, Sharjah, UAE. Email: jabdalla@aus.edu
2) M.Sc., Formerly Senior Lecturer, Faculty of Engineering, University of Science & Technology (U.S.T,), Omderman, Sudan. Currently Lecturer, Institute of Public Administration, KSA. Email: EltahirT@ipa.edu.sa

Abstract:

The architecture, Engineering and Construction (AEC) industry involve many disciplines during the life cycle of the building from its conception to its demolition. The civil engineering discipline has many domains related to AEC industry such domains include structural engineering, foundation engineering, construction engineering and materials. The structural engineering domain has many tasks that include modeling, analysis, design and detailing. The disciplines, domains and tasks of the AEC industry are related and often require data and information to be transferred electronically among them and/or across them. This paper presents an integrated Building Information Model (iBIM) for design of reinforced concrete structures. The proposed iBIM and its architecture have a framework similar to that of the Industry Foundation Classes (IFC) with its file-based data exchange standard and its central data base repository. The iBIM will facilitate integration of the structural engineering design tasks for reinforced concrete structures and will partially provide for interoperability among other related application programs. EXPRESS language and UML are used to define and represent the product and process models of the iBIM. A proof-of-concept prototype implementation of the tasks of the integrated structural engineering system that uses the proposed model and architecture was coded using Delphi object-oriented programming environment. An example of a beam object is shown to validate the proposed model. It is concluded that integration of other tasks of the structural engineering domain can be achieved using the proposed iBIM that made full utilization of the object-based modeling techniques and the IFC framework.

Keywords: object-based, object-oriented, integrated BIM, AEC, BIM, IFC.

1. INTRODUCTION

The architecture, Engineering and Construction (AEC) industry involve many *disciplines* during the life cycle of the building from its conception to its demolition. Such disciplines include architecture, civil engineering, mechanical engineering, electrical engineering, construction engineering and management (Howard et al., 1992; Abdalla et al., 1991; Powell & Abdalla, 1989, Phan et al. 1992). The civil engineering discipline has many *domains* related to AEC industry such domains include structural engineering, foundation engineering, construction engineering and materials. The structural engineering domain has many *tasks* that include modeling, analysis, design and detailing. The *disciplines*, *domains* and *tasks* of the AEC industry are related and often require data and information to be transferred electronically among them and/or across them. Over the years several attempts have been made to develop systems to assist in the integration of the wide variety of the AEC disciplines, their domains and their tasks (Abdalla, 1991). The integration across AEC building industry *disciplines* and among the *domains* of the different disciplines have been a major area of research during the last two decades. As a result several integration architectures have been proposed and numerous standard data models and Building Information Models have been suggested (buildingSMART, 2018; buildingSMART, 2017; Eastman et al., 2008; IFC2x3, 2006). The use of information technology in the AEC industry and its future trends as well as research topics in construction informatics, BIM frameworks and a roadmap for the use of emerging technology have been well defined (Garrett et al., 2004; Turkm 2007; Succar, 2009; Rezgui et al., 2009; Yabuki, 2012). Moreover the relevance of research of information technology to industry has been examined (Issa et al., 2007) and its use in modeling energy in buildings has been proposed (Abdalla & Law, 2014). Although there is a remarkable progress, in recent years, in developing systems for integration within and across most of the AEC disciplines and their domains, however, there is no strong evidence of achieving similar success in integrating the tasks of the structural engineering domain. This could be attributed, partially, to the lack of a well-defined and properly detailed data model to represent the structural engineering data and a robust architecture to facilitate such integration.

The structural engineering design domain of the civil engineering discipline consists of several tasks that include: (1) conceptual and preliminary design; (2) geometric modeling: (3) structural analysis; (4) detailed design; and (5) detailing; and (6) drafting. Many of these tasks were developed and used as disjoint and loosely integrated software systems. In recent years, there is a noticeable increase in structural engineering systems that integrate some or most of these disjoint tasks. The object-oriented paradigm has been recognized, since its early days, as one of the paradigms that helped fueling such integration progress and had noticeable impact on the development of Building Information Model (BIM) in general and on modeling and implementation of integrated structural engineering systems in particular (Abdalla & Powell, 2005; Powell et al., 1989; Garrett et al., 1989). The advantages of using the object-oriented data models in developing integrated structural engineering systems are well recognized (Howard et al., 1992; Abdalla, 1991; Abdalla & Yoon 1991; Abdalla 1989), however, the object-oriented approach almost reached a level of maturity during the last two decades.

As noted, there are several architectures and data models emerged in recent years and they mainly addressed the integration among and across AEC disciplines and their domains. There are several models that have been developed, tested, and implemented for conceptual design, geometric modeling, structural analysis, detailed design and detailing of buildings components; however, integration among these different tasks of the structural engineering domain is still an open area of research. This paper proposes an object-based integrated building information model and suggests an architecture for an integrated structural engineering system based on a framework similar to that of the Industry Foundation Class (IFC). The proposed Integrated Building Model and architecture integrate the structural engineering tasks and processes and facilitate for interoperability among these tasks and other domains.

Using IAI IFC (2000) as a base, several models, extensions and applications have been developed. Halfawy et al., (2005) used the IFC framework for integration of falsework processes using smart objects. Staub-French et al., (2000) tackled the integration of the construction management tasks such as planning, scheduling and cost estimation using the IFC construct and framework. Yabuki et al., (2004) developed a product model for prestressed concrete bridges based on IFC framework. Arthaud et al., (2007) of IAI French Chapter developed an IFC-Bridge model. The effort of Yabuki et al., (2004) and IAI French Chapter were later merged and unified to form the IFC-Bridge standard incorporated into IAI IFC (Yabuki et al., 2006, Arthaud et al., 2007). Yasaka et al., (2006) developed an IFC based product model for concrete called IAI ST-2 which later became part of the IAI IFC Standard. Barak et al., (2009) developed functional requirements for development of a BIM tool for cast-in-place reinforced concrete structures. Bazjanac et al., (1997) used the IAI IFC framework in developing simulation tools for the building industry. Owolabi et al., (2006) developed an IFC assembly viewer. Weise et al., (2003) used the IFC modeling framework to extend structural analysis tasks and Kiviniemi et al., (2005) used IFC model as server to integrate multiple product models.

In this study, an object-based Integrated Building Information Model (iBIM) for integration of reinforced concrete structural design tasks based on a framework similar to that of the IFC is proposed. The following sections present the underlying architecture, product model and process model of the proposed iBIM for structural engineering applications. Details of these elements and their structure will be presented in subsequent sections.

2. THE INTEGRATED BUILDING Information MODEL (iIBM)

The integrated building model consists of three elements, mainly its *integration architecture*, *product model* and *process model*. The *integration architecture* defines the major layers of the iBIM and how they are related. The *product model* defines the form, content, function and behavior of the objects of the iBIM and their class hierarchies. The *process model* defines the interaction scheme and mechanism among the objects of the iBIM (Eltayeb, 2000; Abdalla & Powell, 1991, Abdalla & Eltayeb, 2008).

2.1 Integration Architecture

The advent in computer software and hardware led to the surge and proliferation of integration technology in several engineering fields. As a result, there are several integration methods, philosophies and integration architectures that have been suggested over the years (Abdalla, 1991; Howard et al., 1992). Among the suggested architectures are the translator, neutral file and the central data base architectures, among others. The neutral file is used for exchanging data and information and the central data base is used for sharing data and information. Development of a unified AEC model to facilitate integration through data exchange has been the trend in recent development in the field.

The architecture proposed in this study is a hybrid between neutral file with IFC framework and a Central Data repository (Abdalla, 1991; Eltayeb, 2000). This architecture is not meant to provide a comprehensive interoperability solution, but to provide a platform to facilitate integration and to test and prove some concepts related to interoperability and integration among the structural engineering tasks.

Figure 1a shows the IFC Architecture that consists of four layers: (1) *Resource layer*; (2) *Core layer*; (3) *Interoperability layer*; and (4) *Domain Model layer*. There are links and interfaces among these layers as shown. Figure 1b shows the architecture of the proposed Integrated Building Model. The architecture of the Integrated

Building Model consists of four parts that resemble the four layers of the IFC architecture as follows: (1) The *Building Properties Layer* consists of several property objects that defines properties of component design object. This layer resembles the *Resource Layer* of the IFC architecture; (2) The *Building Components Layer* consists of objects residing in the Central Data Base (CDB) repository of the Integrated Building Model and contains structural design object components. This layer resembles the *Core Layer* of the IFC; (3) The *Building Objects Broker* with its *Standard File Transfer* system as interface. This layer facilitates the extraction and commitment of objects from/to the CDB as well as data exchange. It acts as the *Interoperability Layer* of the IFC; and (4) The *Structural Engineering Tasks* – namely conceptual design, geometric modeling, structural analysis, detailed design and detailing of the IBM are the modules that are similar to the *Domain Modules Layer* of the IFC. Details of these layers of the IBM will be given in subsequent sections.

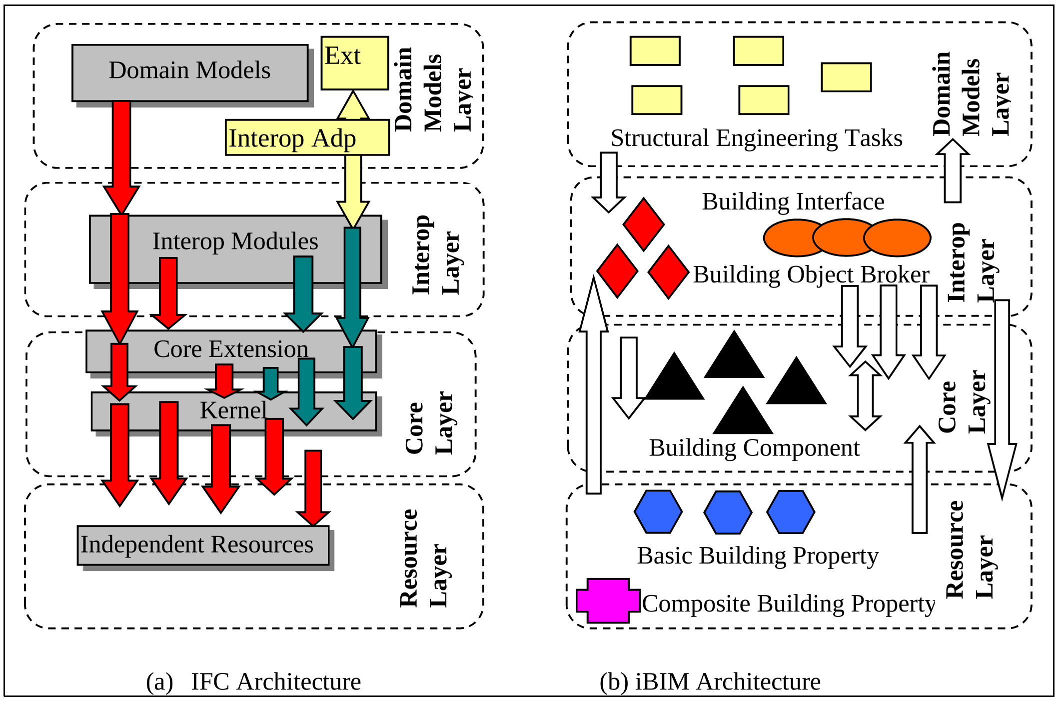

Figure 1. The IFC and the iBIM architectures

2.2 Product Model

A typical reinforced concrete building consists of several interconnected entities that make up the structural system. The entities of a typical R/C building represent (1) basic design components such as beam, column, slab, footing; (2) assembled design components which are assemblage of basic design components such as frame and floor; (3) support or restraint entities representing different restraints such as pinned, fixed, roller; (4) load entities such as dead, live, wind, earthquake and can be applied as distributed, concentrated, etc. (5) material entities; (6) section entities; and (7) connection entities. These objects are combined together to give the general framework of the product model. In an object-oriented data models, these parts can be modeled as objects using complex data types and relationships (Abdalla et al., 1995; Elayeb, 2000; Abdalla & Eltayeb, 2008).

The product model of the proposed iBIM consists of four major classes. The required data for executing the structural engineering tasks (modeling, analysis, design, detailing and presentation) are imported to and exported from these four classes. Each class has a different function and a different internal structure and they are combined together to form the overall product model of the Integrated Building Model. These classes are: (1) *Building Property* class which has several subclasses defining building element properties such as section, material, rebar, load, restraints, etc.; (2) *Building Component* class which has four subclasses defining building basic design components such as Beam, Slab, Column and Footing while; (3) *Building Broker* class and (4) *Building Inte*rface class. Figure 2 shows the *Building Property* Class subtype tree for Beam Object in EXPRESS-G. Details of several Property Classes for reinforced concrete design application were similarly developed.

Presented at the 17th International Conference on Computing in Civil and Building Engineering, June 22-24, 2018
http://programme.exordo.com/icccbe2018/delegates/presentation/334/

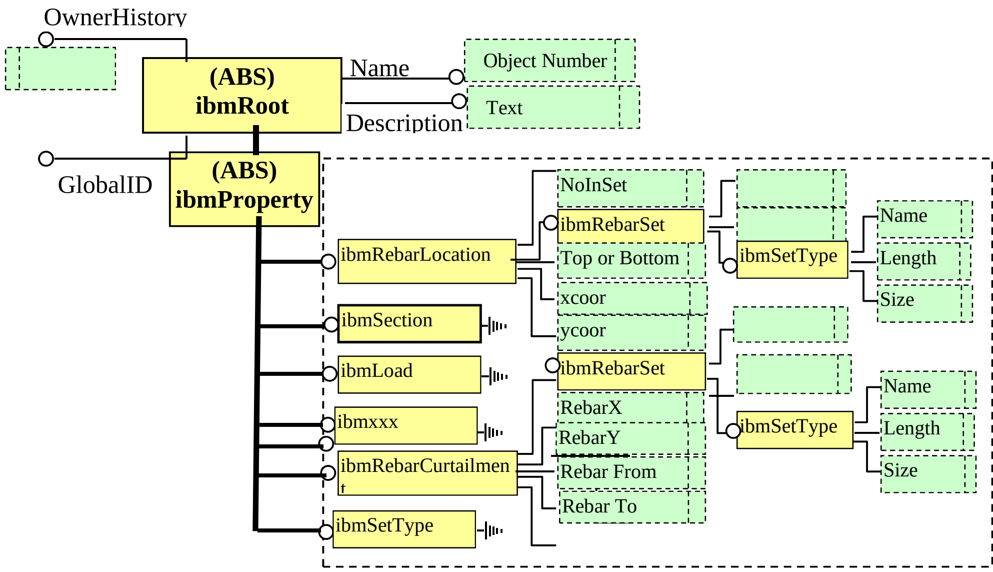

Figure 2. Definition of the Property Class subtype tree for Beam Object in EXPRESS-G

2.3 Process Model

The process model defines the interaction among the different elements of the product model defined above. The process model of the proposed iBIM involves nine steps. These steps are needed in order to execute the stages of the structural design tasks as shown in the UML sequence diagram of Figure 3. These steps must be applied sequentially as follows: (1) Initiate the process; (2) Get the building database file name for the current model. This is done by providing some fields in the *Interface* class by the user; (3) The *Interface* class sends a message to the *Property* class to execute some methods that are used for importing data from the building database; (4) The *Property* class sends a message to the *Broker* class to determine the current level of the building and activate one component at that level (e.g., beam component). This step will be repeated each time a specific level is selected by the user; (5) The *Broker* class sends a message to the *Component* class to activate the object of the current component (e.g., Beam Object); (6) The activated component object then import its local data from the component database files (beam analysis result file and beam design result file) such as component end forces, section dimensions and some global data from the property such as loading and reinforcement steel type. This step will be repeated each time a specific component is selected by the user through the *Interface* class; (7) The component object will now contain all the required data for design. Through the user interface of the component program, the designer can set or change the design options and design the current component; (8) Using the Presentation methods the user can display the results in several different ways; and (9) End or Terminate the process. More details of process communication models are given in (Abdalla 2006a, 2006b).

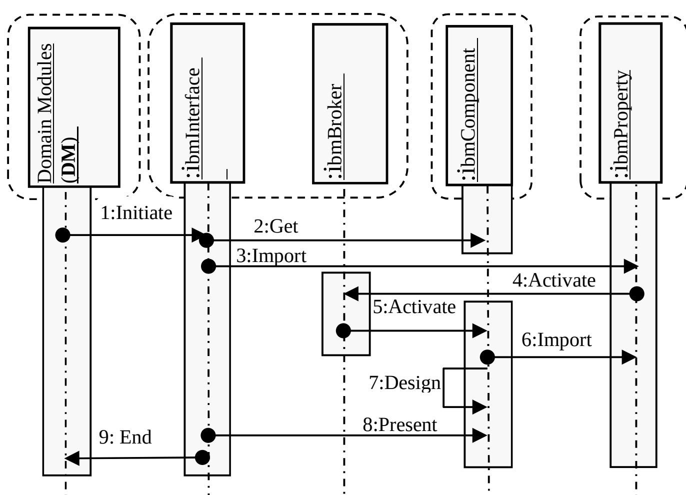

Figure 3. UML Sequence Diagram for a generic Process Model procedure

3. ELEMENTS OF THE PRODUCT MODEL

As previously indicated, the product model comprises four major classes and numerous subclasses for defining component, property, broker and interface entities. The four major classes are *Property classes*, *Component classes*, *Broker class* and *Interface class* and the data items that were implemented for the iBIM. Each object has, among other attributes, an object identifier and a type indicator. The object identifier is used to uniquely identify the object and the type indicator is used to indicate the object type.

3.1 The Property Classes

The *Building Property* class is the root class for all property objects which are the Resource Layer of the iBIM. Property classes are either Basic Property class or Composite Property class that comprises basic property classes. The objects of the Property classes are used to define specific properties and state of the component objects or building object. The Property class is a *container* that consists of several classes. It includes basic Property classes such as material, section, load as well as analysis classes for storing analysis data (maximum shear, maximum moment), design classes for storing design data and final details (steel types, design method, etc.) and presentation classes for displaying analysis results and design details. Table 1 shows a partial list of property classes as pertained to Beam component. Analogous property classes for Slab, Column and Footing components were also developed. Details of two related property classes – Rebar Location and Rebar Curtailment classes are shown in Figure. Other building property classes of Table 1 are defined in a similar fashion.

Table 1. Partial list of Basic Property Classes pertaining to Beam Component

No	Resource Object	Description
1	Section	Defines component cross section data – depth, width, etc.
2	Load	Defines load data – type , value , location , etc.
3	Rebar Type	Defines types of rebar used for flexure and shear – grade , length , size, etc
4	Rebar Location	Defines the location of the bar on the designed beam section.
5	Rebar Curtailment	Defines the details of rebar distribution along the beam.
6	BeamAnalysisResults	Defines Beam Component analysis results – end forces , pointer to loads, etc.
7	BeamDesignResults	Defines Beam Component design results – steel_set,rebar distribution, etc.

Objects of Rebar Location class are used to locate the location of rebars on the designed beam section. It has five attributes as shown in Figure 4a. Objects of Rebar Curtailment class are used in the beam object to identify the details of reinforcement (the distribution of bars along the beam length). It determines the type of bars and their positions in the cross section of the beam object, and the end positions of each bar on the longitudinal beam section. It has six attributes as shown in Figure 4b.

```
ENTITY IbmRebarLocation;
        NoInSet: INTEGER;
        RebarInSet: IbmRebarSet;
        TopOrBottom: STRING;
        XCoor: INTEGER;
        YCoor: INTEGER;
END_ENTITY;
```

(a)

```
ENTITY IbmRebarCurtailment;
        SUBCLASS OF (IbmPropertyRoot);
        RebarInSet: IbmRebarSet;
        RebarXCoor: INTEGER;
        RebarYCoor: INTEGER;
        RebarFrom: REAL;
        RebarTo: REAL;
END_ENTITY;
```

(b)

Figure 4. Structure of Rebar Location and Rebar Curtailment Classes

3.2 The Component Classes

The *Building Component* class is the root class for all structural design objects which are the backbone of the Core Layer. Its four subclasses are: (1) *Beam* class; (2) *Slab* class; and (3) *Column* class; and (4) *Footing* class. Each one of the component objects contains all the variables and relationships needed for the arithmetic operation and for modeling, analysis, design, detailing, and presentation of results. The information within the object is expressed as: (a) Data Items (variables) that represent attributes about either the physical or behavioral nature of object or a performance limitation on the object; (b) Relationships between these data items (Methods) which describe the behavior of the object.

3.3 The Broker Classes

The basic task of the *Building Broker* class is to keep track of the building components such as Beams, Slabs, Columns and Footings. It assigns the pointers to the components to locate them in the database. The *Broker* class has a data attribute called *Component Status* that has two fields – one for determining the component status and the other for the pointer.

3.4 The Interface Classes

The *Building Interface* class is used for interaction between the user and the program module, and for control of the iBIM process sequence. This class is event driven. The *Events* are executed automatically when a specific condition is satisfied. There are two types of events: (1) user events which are triggered by user actions such as mouse click; For example, most items in the Graphical User Interface contain a mouse click event that can be implemented to respond when the user clicks the item. (2) control events which are triggered by the application program actions which is triggered every time a new object is created during the run time to assign default values. For example, there is a control event that will interrupt the design process when a component violates the design constraint.

4. IMPLEMENTATION, TESTING AND VALIDATION

A computer program has been developed to test the concepts proposed in this paper. The program is named Building Structural Analysis And Design (BSAAD) (Eltayeb, 2000). This program controls a number of subprograms for design of building components (i.e., beams, columns, slabs, and footings. The program is coded using *Delphi* language (Visual Pascal) version 4.0 (Inprise, 1998). Figure 5 shows the Beam main designer screen of the developed prototype computer program.

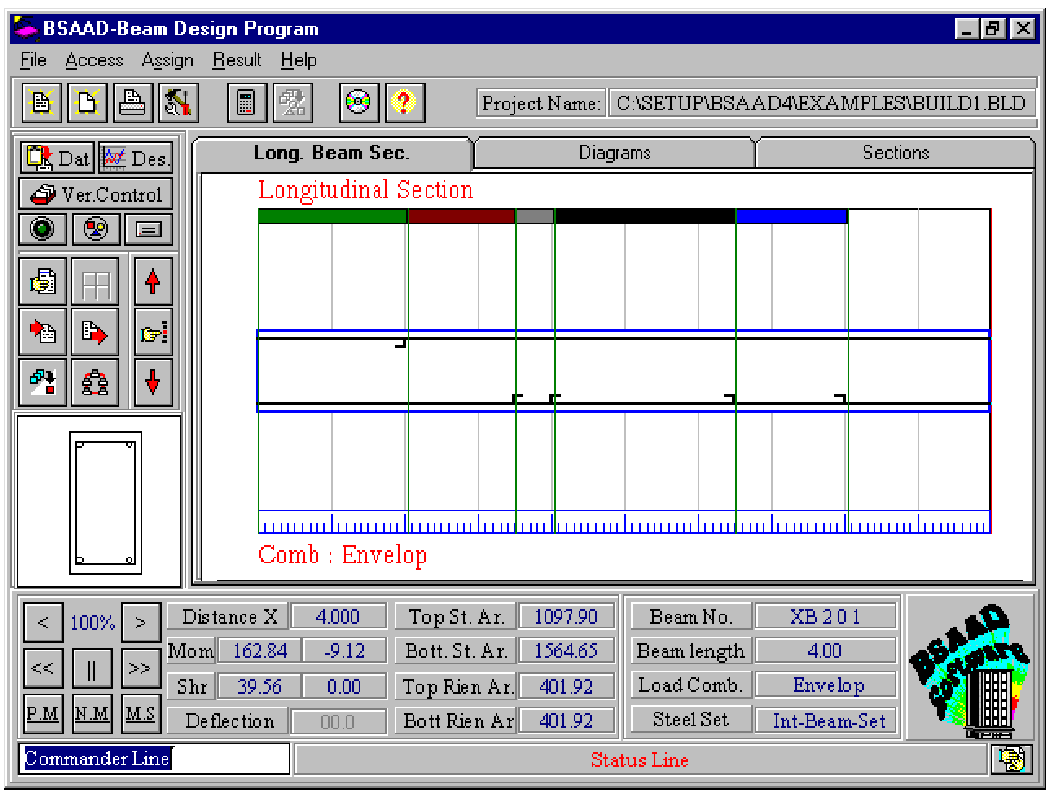

Figure 5. Beam main design screen

5. CONCLUSIONS

In this paper an Integrated Building Information Model for product and process modeling of structural engineering tasks, based on the object-oriented data model and the IFC architecture and framework, was developed. The model supports many operations for analysis and design of reinforced concrete building elements. The main idea is that a building consists of a number of floors. These floors decomposed into three components (beam, slab and column). These components form the basic element in the model which is called a *Component* class. Each class of these components contain data items and methods required for modeling, analysis, design, detailing and presentation of results. Besides the *Component* class, there are other supporting classes such as *Broker*, *Property*, and *User Interface* classes. It was demonstrated that several systems for integrating tasks of the different domains of the civil engineering discipline of the AEC industry can be designed and implemented using the concepts of the developed iBIM that made full utilization of the object-oriented data model features and the IFC architecture and framework.

REFERENCES

Abdalla, J. A. and Powell, G. H. (1995). An object design framework for structural engineering, *Engineering with computers*, 11 (4), 213-226.

Abdalla, J. A., Phan, D. H. D., Howard, H. C. (1991). Form, function and behavior in structural engineering knowledge Representation, *International Conference on the Application of Artificial Intelligence Techniques to Civil and Structural Engineering*, Oxford, UK, pp. 1-10.

Abdalla, J. A. (1991). An object-oriented architecture and concept for an integrated structural engineering system, *International Conference on the Application of Artificial Intelligence Techniques to Civil and Structural Engineering*, Oxford, UK, pp. 147-155.

Abdalla, G. A. (1989). Object-oriented principles and techniques for computer-integrated design, University of California, Berkeley.

Abdalla, J. A. and Yoon, C. J. (1991). Object-oriented finite element and graphics data-translation facility, *Journal of computing in civil engineering*, 6 (3), 302-322.

Abdalla, J. A. and Eltayeb, T. M. (2008).Version and configuration model for structural design objects: Design and implementation, *Journal of Computing in Civil Engineering*, 22 (3), 147-158.

Abdalla, J. A. and Powell, G. H. (1991). Version management needs for structural engineering design, *Engineering with computers,* 7 (3), 131-143.

Abdalla, J. A. (2006a). A communication model for structural design objects. II: Performatives and protocols, *Advances in Engineering Software*, 37 (6), 393-405.

Abdalla, J. A. (2006b). A communication model for structural design objects. I: Channels and medium, *Advances in Engineering Software*, 37 (6), 382-392.

Abdalla, J. A. and Law, K. H. (2014). A Framework for a Building Energy Model to Support Energy Performance Rating and Simulation, *Computing in Civil and Building Engineering*, Orlando, Florida, USA, pp. 227-234.

Arthaud, G. and Lebegue, E. (2007). IFC-BRIDGE V2 Data Model, *Edition fonctionnalité R7 – IAI French Chapter.*

Barak, R., Jeong, Y.-S., Sacks, R., and Eastman, C. M. (2009). Unique Requirements of Building Information Modeling for Cast-in-Place Reinforced Concrete, *Journal of Computing in Civil Engineering*, 23(2), 64-74.

Bazjanac, V, and D B Crawley (1997). The Implementation of Industry Foundation Classes in Simulation Tools for the Building Industry. *Proceedings of Building Simulation* 97, Prague, Czech Republic, 7-10 September 1997. IBPSA. Volume I, pp 203-210.

buildingSMART. (2017). *IFC2x3*. Retrieved from buildingSMART website: http://www.buildingsmart-tech.org/ifc/IFC2x3/TC1/html/index.htm, accessed on December 15, 2017.

buildingSMART. (2018). *IFC4 Release Candidate 4*. Retrieved from buildingSMART website: http://www.buildingsmart-tech.org/ifc/IFC2x4/rc4/html/index.htm, accessed on February 10, 2018.

Eastman, C., Teicholz, P., Sacks, R., and Liston, K. (2011). *BIM Handbook: A Guide to Building Information Modeling for Owners, Managers, Designers, Engineers, and Contractors* (2nd ed.). John Wiley & Sons.

Eltayeb, T. M. (2000). Computer Aided Design of Reinforced Concrete Structures: Modeling and Implementation, M.Sc. Thesis, University of Khartoum, Khartoum Sudan.

Halfawy, M. and Froese, T. (2005). Building Integrated Architecture/Engineering /Construction Systems Using Smart Objects: Methodology and Implementation, *Journal of Computing in Civil Engineering* 19(2),172-181.

Howard, H. C., Abdalla, J. A., Douglas Phan, D. H. (1992). Primitive-composite approach for structural data modeling, *Journal of Computing in Civil Engineering*, 6 (1), 19-40.

Garrett, J. H., Flood, I., Smith, I. F. C. and Soibelman, L. (2004). Information Technology in Civil Engineering – Future Trends, *Journal of Computing in Civil Engineering*, 18(3), 185-186.

Garrett, J. H. and Basten, J. (1989). An Object-Oriented model for Building Design and Construction, *Computer utilization in structural engineering*, Nelson J. K. ed. 332-341.

Kiviniemi, A., Fischer, M., Bazjanac, V., (2005). Integration of Multiple Product Models: IFC Model Servers as a Potential Solution, *Proceedings of 22nd W78 Conference on Information Technology in Construction*, Dresden, Germany, July 2005.

IAI IFC (2000). International Alliance for Interoperability, IFC Technical Guide. Industry Foundation Classes - Release 2x.

IFC2x3 (2006). *IFC 2xEdition3 Model Implementation Guide*. Model Support Group (MSG) of the IAI Thomas Liebich (lead),Yoshinobu Adachi, James Forester, Juha Hyvarinen, Kari Karstila, Jeffrey Wix.

Inprise Corporation (1998). Delphi 4.0, User Guide.

Issa, R. and Anumba, C. (2007). Computing and Information Technology (IT) Research in Civil Engineering – Self-Fulfilling or Industry Transforming, *Journal of Computing in Civil Engineering*, 21(5), 301-302.

Owolabi, A., Anumba, C.J. El-Hamalawi, A. and Harper, C. (2006). Development of an Industry Foundation Classes Assembly Viewer, *Journal of Computing in Civil Engineering*, 20(2), 121-131.

Phan, D. H., Abdalla, J. A., Howard, H. C. (1992). An integrated representation of form, function and behavior in structural engineering, *Computing in Civil Engineering and Geographic Information Systems Symposium*, Dallas, Texas, USA, pp. 394-401.

Powell, G. H., Abdalla, J. A., Sause, R. (1989). Object-oriented knowledge representations: cute things and caveats, *Computing in Civil Engineering: Computers in Engineering Practice*, Atlanta, GA, USA, pp. 1-8.

Powell, G. H. Abdalla, G. A. (1989). Object management in an integrated design system. *Computing in Civil Engineering: Computers in Engineering Practice*, Atlanta, GA, USA, pp. 556-563.

Rezgui Y., Boddy S., Wetherill M. and Cooper G. (2009). Past, present and future of information and knowledge sharing in the construction industry: Towards semantic service-based e-construction, *Computer-Aided Design*, In Press, Corrected Proof, Available online 21 June 2009.

Staub-French, S. and Fischer, M. (2000). Practical research issues in using Industry Foundation Classes for construction cost estimation. *CIFE working paper #56*, Stanford University, Stanford, CA, USA.

Succar, B. (2009). Building information modelling framework: A research and delivery foundation for industry stakeholders, *Automation in Construction*, 18(3), 357-375.

Turk, Z. (2007). Construction Informatics in European Research: Topics and Agendas, *Journal of Computing in Civil Engineering*, 21(3), 211-219.

Wix, J., Liebich, T. (2003). “IFC: Current Reality and Future Potential, *Proceedings of 10th ISPE International Conference on Concurrent Engineering: Research and Applications*, Madeira, Portugal, July 2003.

Yabuki, N. and Shitani, T. (2004). Development and integration of an IFC-based Product Model for Prestressed Concrete Bridges. *Proceedings of the fib Sylmposium*, Avignon, France, (CD-ROM), 6 pages.

Yabuki, N., Lebegue, E., Gual, J., Shitani, T. and Zhantao, L. (2006). International Collaboration for Developing the Bridge Product Model "IFC-BRIDGE". *Proceedings of the Joint International Conference on Computing and Decision Making in Civil and Building Engineering*, June 14-16, 2006, Montreal, Canada, pp.1927-1936.

Yabuki, N. (2012). Impact of ICT to civil engineering and roles of civil engineers, *Proceedings of the International Workshop on ICT in Geo-Engineering*, Kyoto, Japan, pp.3-9.

Yasaka, A. Kataoka, H., Kashima, K. Takeda, M., Usami, M., Yabuki, N., Matsumoto, N., Furukawa, S., Mogi, Y. and Adachi, Y. (2006). The Development of the Reinforced Concrete Structural Model on IFC Specification, *Proceeding of the Joint International Conference on Computing and Decision Making in Civil and Building Engineering*, June 14-16, 2006, Montreal, Canada, pp. 3116-3125.