

Standard-based Data Interoperability of the Building Information Model in Cloud

Kereshmeh Afsari, Ph.D.
Purdue University
West Lafayette, Indiana

During model-based collaborations in the Architecture, Engineering, and Construction (AEC) industry, files are largely used for exchanging Building Information Model (BIM). But, web as the enabler of cloud computing has provided the opportunities for a network-based data exchange as oppose to a file-based data transfer. Current Cloud-BIM interoperability solutions do not provide a loosely-coupled network-based system with the flexibility to reduce dependencies among collaborating Cloud-BIM applications. Thus, the need for a new interoperability framework is identified in the prior study and major components of the cloud interoperability are specified to improve model-based collaboration during the design and construction of buildings. This paper outlines the components of the new Cloud-BIM interoperability framework known as BIM Synapse that supports a network-based BIM data exchange for loosely-coupled collaboration in cloud. In addition, implementation of major layers of BIM Synapse are specified and the dataflow in this framework is explained. BIM Synapse enables a Cloud-based collaborative process allowing Cloud-BIM services to communicate through their standardized Application Programming Interfaces (APIs). Since BIM Synapse uses established standards, it provides a common understanding of BIM data. Also, implementation of BIM Synapse in Cloud-BIM can enable the integration of Internet of Things (IoT) with BIM models.

Key Words: Cloud, BIM, Interoperability, IFC, REST API

Introduction

In the Architecture, Engineering, and Construction (AEC) industry, model-based collaboration within Building Information Modeling (BIM) is mainly based on the exchange of files. These files are either in vendor specific file formats or in neutral format using Industry Foundation Classes (IFC) as open BIM standard. However, web technologies have enabled Cloud-based BIM services and provided an opportunity to exchange non-file based data via the web and over the networks. A network-based data exchange can specifically enable the integration of the Internet of Things (IoT) that requires network connectivity and provision of resources through the Web of Things (WoT) (Wilde, 2007) with the BIM process. In fact, cloud computing provides real-time access to a pool of data, on-demand access to computing resources (e.g. storage, servers) and applications, higher performance, and potentially better interoperability (NIST, 2014; Jadeja & Modi, 2012). Thus, Cloud can be an effective enabler to address current BIM challenges. Importantly, Cloud-based BIM is believed to be a cost-effective alternative to the current state of Building Information Modeling and data sharing (Juan & Zheng, 2014; Wong et al., 2014, Mahmudu et a., 2013). Since potential values of Cloud-based applications such as efficiency and low-cost have been clearly recognized, Cloud-BIM is believed to create a promising direction in BIM development. Cloud technologies are still new but it is anticipated that Cloud-BIM will change the AEC industry and its collaborations (Wong et al., 2014).

Previous study by the author (Afsari et al., 2016a) has explained current status of BIM interoperability indicating challenges and limitations of current BIM data exchange and interoperability. Also, the author has identified in a previous study (Afsari et al., 2016b) that current Cloud-BIM interoperability solutions do not provide a loosely-coupled system with the flexibility to reduce dependencies among collaborating Cloud applications. In addition, previous work by the author (Afsari et al., 2017a) highlighted the need for a new interoperability framework and specified how the integration of four major components of the Cloud interoperability i.e. Cloud API, data transfer protocol, data format and standards can improve model-based collaboration during the design and construction of the project. The objective of this paper is to outline the implementation of a new Cloud-BIM interoperability framework known as BIM Synapse that supports a network-based BIM data exchange for a loosely coupled collaboration in the Cloud. This study explains how BIM Synapse redefines BIM dataflow in Cloud-BIM applications by using web technologies and how it restructures current BIM dataflow.

Cloud-BIM Interoperability and BIM Synapse

During the design and construction of a building, it is common that the AEC project partners including the architects, engineers, construction team, and fabricators work on several software tools to model the building and create separate BIM models. To enable the collaboration, BIM data needs to be exchanged among software applications for different purposes such as structural analysis, detail design, clash detection, fabrication, etc. But one of the main challenges in this collaboration a BIM model exchanged from an application might not be compatible in the receiving application (Eastman et al., 2011). In order for the applications to exchange BIM data regardless of vendors and data formats, Industry Foundation Classes (IFC) schema has been specified (buildingSMART International, 2008). IFC is a standard established by International Organization for Standardization (ISO) to support a neutral data format for facilitating cross-platform BIM exchange (Eastman et al., 2011; buildingSMART International, 2008). Using the IFC data model, building components and processes can be defined in a publicly available data schema as an open standard (Eastman et al., 2011). IFC data schema is represented as an EXPRESS schema specification (Schenck & Wilson, 1994) as well as XSD schema specification (Nisbet & Liebich, 2007). EXPRESS as an ISO standard (Schenck & Wilson, 1994) uses the STEP Physical File (SPF) structure in ".ifc" format. XSD schema specification uses the XML document structure and represents the same data as represented in EXPRESS specification of IFC data model (buildingSMART International, 2008) in ".ifcXML" or ".xml" format. Alternatively, ifcJSON data schema is specified by the author in a previous study (Afsari et al., 2017b) developed based on IFC EXPRESS specification and Javascript Object Notation (JSON) data schema to standardize JSON representation of IFC schema. Since the inadequacy of XML data exchange in web services are previously recognized (Wang, 2011), ifcJSON schema and document can provide a more effective data serialization for web-based data transfer. ifcJSON encodes BIM data in ".ifcJSON" or ".json" format that can be exchanged over a network using web technologies (Afsari et al., 2017b).

In addition, U.S. National BIM Standard specifies a set of interoperable standards based on IFC specification to support building data exchange for a building's lifecycle (NIBS, 2015), develops a functional specification as Information Delivery Manual (IDM) and translates it to a technical specification based on IFC schema for software vendors to implement. The product of this implementation is a set of Model View Definitions (MVDs) to define requirements for standard data exchange for tasks during design, engineering, and construction (Eastman et al., 2011). Upon implementation of the MVD, it can be used in the industry by the project partners in model-based collaborations (Eastman et al., 2011; Venugopal et al., 2011). Accordingly, as an industry-wide open standard, IFC can perform as the basis for Cloud-BIM interoperability to ensure a common understanding of building data among BIM applications and project disciplines (Afsari et al., 2016a). But as identified by the author in a previous study (Afsari et al., 2016a), Cloud-BIM interoperability currently face challenges in the exchange of BIM models across the applications. In addition, the potentials of IFC specification and MVD definition have only been used in file-based BIM data exchange. In fact, current Cloud-BIM interoperability solutions have not fully exploit the potential of the web technologies towards a loosely coupled integration (Afsari et al., 2017). As a result, current challenges have highlighted the significance of a new framework for Cloud-BIM interoperability.

In the previous study by the author (Afsari et al., 2017), the need for a new data exchange architecture that utilizes web technologies to redefine the dataflow has been discussed and an approach to a framework for Cloud-BIM interoperability has been specified. Therefore, BIM Synapse is proposed as a framework for Cloud-BIM data interoperability. This paper specifies the components of BIM Synapse framework and provides a guide for its implementation. BIM Synapse uses IFC data in web compatible data format and specifies the design for a loosely-coupled interoperability solution with data request capability on the receiver application based on IFC specification and MVD definition. The reason behind naming of BIM Synapse is based on the notion of synapse in human nervous system. Synapse is a structure that enables the communications among neurons or nerve cells with a gap junction that passes signals (Kolb et al., 2011). It is where the communication among neurons happens and for that reason, it has similarity in function with BIM Synapse framework. BIM Synapse deploys Cloud interoperability features outlined in previous study (Afsari et al., 2017a) as well as IFC data model to address current challenges of BIM data exchange in the Cloud.

BIM Synapse Components

BIM Synapse uses web technologies as well as Cloud interoperability features to enable Cloud-BIM data exchange via web. The architecture of BIM Synapse is shown in Figure 1. In BIM Synapse, BIM data for Cloud collaboration is stored in a data-store within the sending application (e.g. architectural design application) and can be accessed through an API with a standard set of methods and terminologies. This BIM data can be interpreted by the receiving application (e.g. structural design application). The API in fact, provides an on-demand access to the pool of data on the sending application. Accordingly, data request happens only in the receiving application that becomes the client for the sending application and uses the data exchange requirements (i.e. MVD) to request for the BIM data. Therefore, in this process, data exchange requirements will follow the MVD specification. In BIM Synapse, since the receiving application initiates the data request, MVD implementation happens on the receiving application side (i.e. the client). The client uses MVD specification to request for a set of API resources that corresponds to the exchange requirements in the MVD definition. Upon receiving the data, the MVD-based resources will be mapped to native bindings.

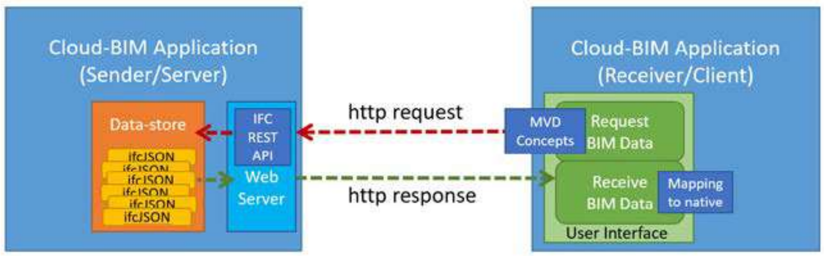

Figure 1: BIM Synapse Architecture

The architecture of BIM Synapse supports real-time data exchange based on a loosely-coupled collaboration enabled by an API following the principles of Representational state transfer (REST) web services explained in the next section. In other words, BIM Synapse will not be tying an application to a new integration system like a BIM server technology, a data interchange hub, or a Cloud integration platform. Instead, BIM Synapse allows the sending and receiving Cloud applications to exchange data directly through their RESTful APIs. Major components of BIM Synapse architecture are based on the identified features of Cloud interoperability (Afsari et al., 2017a) as follows: A RESTful API, data format, data transfer protocol, and standards as well as a data-store to store the BIM data for the REST API. BIM Synapse architecture suggests that each application API in Cloud service provider should implement a set of standardized resources stored in a data-store with an appropriate data serialization format that is retrievable over Hyper Text Transfer Protocol (HTTP).

IFC REST API

BIM Synapse implements a web API based on Representational State Transfer (REST) principles for Cloud-BIM applications. REST is an architectural style for network-based applications (Fielding, 2000) that is based on stateless client-server communication and specifies how resources should be defined and accessed through a network. In a stateful communication, data request is dependent on the state registered by the server but when avoiding application state, all state information will be contained in the interactions with a client through a server (Wilde, 2007) thus, the requests are self-contained providing simplicity, better performance and scalability. REST avoids application state, ensures that application resources are represented and identified by Uniform Resource Identifiers (URI), and sustain all state information required within the interactions between client and server (Wilde, 2007). In addition, the API for BIM Synapse follows a common data model using IFC specification as industry-wide standard data model to enable semantic interoperability.

Data Format

In BIM Synapse, the data should be serialized in a web compatible format based on common agreements and terminologies. Since IFC is the common standard for open BIM, IFC specification can provide the capabilities to capture domain knowledge as well as a common data schema. Currently ifcXML is the only certified IFC data representation that is compatible with web technologies. However, because of the limitations of XML-based documents (Wang, 2011), ifcJSON is used in BIM Synapse which can achieve higher efficiency than XML encoding. The ifcJSON is a JSON serialization of IFC data model for web-based data transfer (Afsari et al., 2017b).

Data-store

In BIM Synapse, the data-store performs as a repository to persist BIM data. The data serialized in appropriate format (i.e. ifcJSON) needs to be stored and retrieved to and from the API backend. A data-store performs as a repository for REST resources. In REST API, resources are the core and everything is modelled as resources while each resource is identified with a unique URI. A resource is an object with relationships to other resources (Fielding, 2000). Defining REST resources is a core part of making things available on the web (Wilde, 2007) which specifies the organization of the data-store.

Data Transfer Protocol

BIM Synapse uses HTTP calls (e.g. GET, POST, PUT and DELETE) for cross-platform communication. It is based on a simple request/response mechanism. In BIM Synapse the receiver of data directly deals with getting the data instead of waiting for the sending application to export data in the form of files. In fact, Cloud computing is based on TCP/IP or Transmission Control Protocol/Internet Protocol (Gong, 2010) and HTTP is a common TCP/IP protocol which World Wide Web traffic mostly uses (IBM, 2006). HTTP enables multiple connections to interact with separate elements concurrently thus, speeds up data transmission. HTTP is a simple messaging protocol but it includes the ability to search a database with a single request (Fielding et al., 1999). This allows the HTTP protocol to search the data-store and return results.

BIM Synapse Implementation

Three major components of BIM Synapse in this implementation, as shown in Figure 2, are: 1) persistence layer, 2) API layer, 3) client interaction layer. The persistence layer stores ifcJSON resources for the API, the API layer implements a RESTful API based on IFC specification to ensure a common understanding of the building data, and the integration layer in the client side enables the interactions with the API to request and receive required BIM data. List of technologies used for implementing each layer is shown in Figure 2. Also, Figure 3 diagrams the dataflow from the sender of the BIM data (i.e. server side) to the receiver of the BIM model (i.e. client side). This happens after the receiving application sends a request to a sending application. In the server side, BIM data in the form of ifcJSON resources reside in IFC REST API and are maintained in the data-store in the form of Binary ifcJSON. In the client side, ifcJSON resources are requested and received based on data requirement in MVD. Then, the ifcJSON resources are merged to form the BIM model and are translated to the native model in the receiving application.

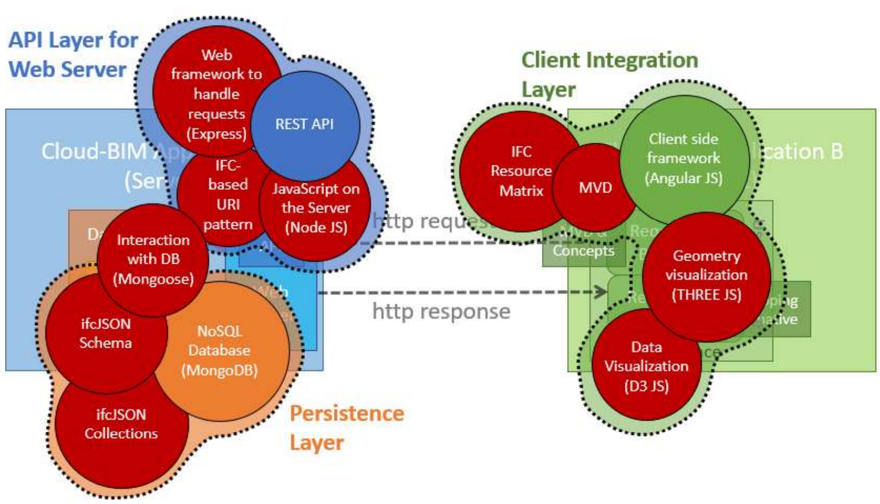

Figure 2: Overview of the technologies used in three implementation layers

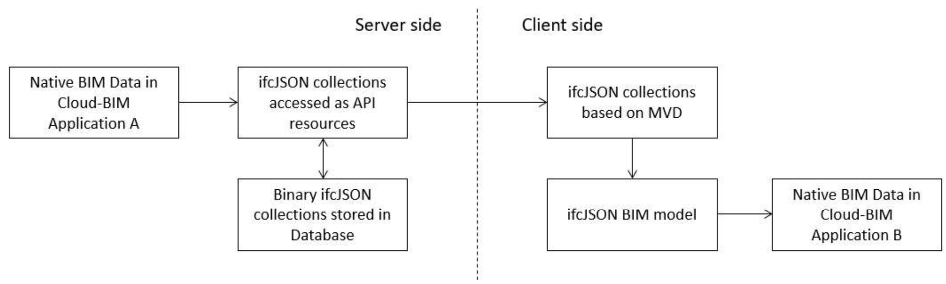

Figure 3: Dataflow from server to client native environments

The implementation guide for the three layers of BIM Synapse are specified below.

- Server-side persistence layer: To create a repository for REST resources, this study uses MongoDB which is a NoSQL database. NoSQL which stands for "Not Only SQL" is a database mechanism that stores data different from tabular relations in relational database. NoSQL databases can horizontally scale over many servers and can support a large number of operations and data while allowing data to be added dynamically (Cattell, 2010). Instead of two-dimensional table structures, NoSQL databases store documents (e.g. ifcJSON resources) and provide indexes on documents (Cattell, 2010). In this implementation of the BIM Synapse, Mongoose, as the Object Document Mapper (ODM), is used to enable the interaction between the IFC REST API and the ifcJSON resources maintained in the data-store.
- IFC REST API: IFC REST API on the server-side acts as an interface for querying and persisting data in the MongoDB database. This API for the web server in the sending application enables the collaborating Cloud-BIM application (i.e. client for this server) to request and receive required BIM data. To implement IFC REST API, Node.js and Express.js are used. Node.js is an open-source JavaScript runtime environment that enables executing scalable server applications. Also, Express.js is an open source web application framework for Node.js that handle the requests on the server side.
- Client-side Interaction: To interact with the IFC REST API, a frontend application has been created which can be used in the client side. In the client-side application (i.e. the receiver of BIM data), an integration layer should be implemented where the receiver of BIM data can connect to IFC REST API of the server Cloud-BIM application (i.e. sender of the BIM data) and request for the API resources to receive the ifcJSON data. Upon receiving required BIM data, the receiving Cloud-BIM application should translate the ifcJSON BIM model to its native binding. For the implementation of frontend application to interact with REST resources, this study uses AngularJS, a client-side framework built on JavaScript based on Model View Controller (MVC) architecture with three parts: model for maintaining data, view for displaying data, and controller that controls the interaction between model and view.

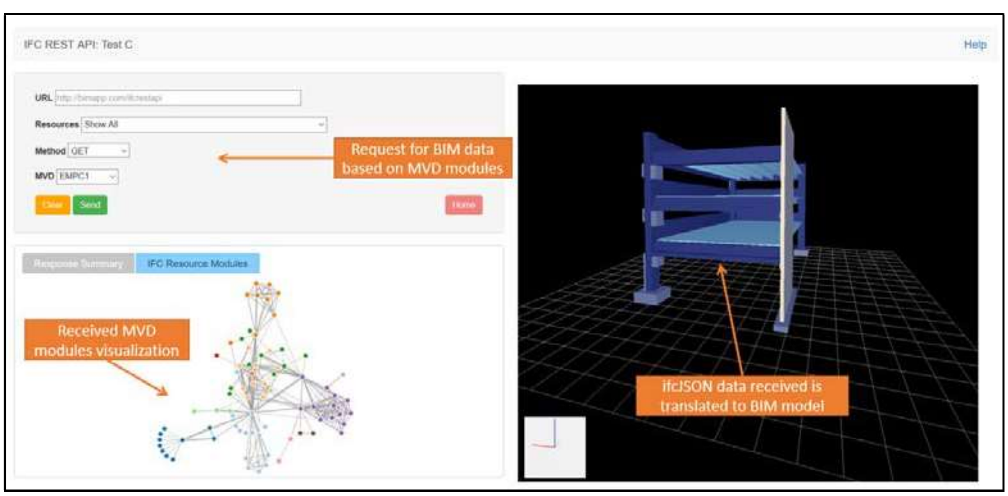

Figure 4: BIM Synapse implementation for precast concrete MVD

BIM Synapse in this study is implemented and evaluated for precast concrete using precast concrete Model View Definition based on precast concrete BIM standard (Afsari and Eastman, 2016c). Figure 4 shows BIM Synapse implementation for precast concrete exchange model number 1 i.e. EMPC1. EMPC1 consists of concept design layout of precast pieces and it is in fact a subset of IFC schema that consists of architectural concept model or engineering concept model passed to detailer for further preliminary precast structural and fabrication detailing (Afsari and Eastman, 2016c). In this study, the MVD for precast concrete is revised and is then modularized as REST resources. The details about REST resource implementation and the modularization technique REST resource identification based on IFC specification will be explained in a future study. The use of BIM Synapse framework enables BIM data to be exchanged over the network in web compatible data format. Instead of exchanging file-based data, BIM Synapse maintains and transfers modularized BIM data as the API resources. BIM Synapse provides BIM resources through the Web of Things (WoT) to enable the integration of IoT with BIM data that would not be possible with conventional methods of BIM data transfer.

Interoperability Standard for Cloud-BIM

As standards provide a common language between systems, data exchange schema and interoperability standards allow data to be shared between applications regardless of the application or application vendor. Established domain schema can enable the interoperability and the use cases will help understand interoperability scenario. Therefore, domain specific standards are crucial for Cloud interoperability (Lewis, 2013; Loutas, 2011). BIM Synapse framework with its IFC REST API and data structure that follows IFC specification, provides a common understanding of BIM data, resource management, and data request/response in the Cloud. BIM Synapse also integrates MVD as a subset of IFC schema to address specific exchange requirements. BIM Synapse is among the first efforts towards standardization of the AEC data in the Cloud. Cloud-BIM interoperability requires industry engagement to establish the standard and certify BIM applications. It needs the involvement of BIM software vendors to implement the framework and conduct user testing.

IFC specification can assist with semantic interoperability of building data in the Cloud and ifcJSON is anticipated to be widely used in Cloud-based Building Information Modeling solutions to improve interoperability of Cloud-BIM applications. Since BIM Synapse uses ifcJSON as its main data encoding, it has the benefit of explicit semantics over conventional SPF data where the semantics are hidden. To interpret the data in an SPF file, the IFC specification is required along with the file to provide an understanding of the attributes. The ifcJSON document can capture, store, and aggregates semantics of data since it provides data with explicit representation of attributes or properties (Afsari et al., 2017b). It is anticipated that ifcJSON will be of interest to web application development and Cloud computing community to replace ifcXML in most cases. Therefore, the ifcJSON implementation as well as REST resource management in BIM Synapse, needs to be developed fully to be added as a buildingSMART standard. In BIM Synapse, RESTful IFC API is based on a set of REST API design patterns while it also uses ifcJSON for resource representation. This makes the resources compatible with REST as it follows JSON data model and at the same time, it aligns the resources with the AEC standards since it follows IFC specification. The identification of API resources and the resource schema itself is based on IFC concepts and MVDs which details will be explained in a future study. In fact, BIM Synapse integrates MVD as a subset of IFC schema to address specific exchange requirements. In other words, using BIM Synapse, the user can request for BIM data in the client side based on MVD to send a GET request to IFC REST API. Each MVD contains a set of concepts and requesting a BIM model based on an MVD will return all ifcJSON resources that corresponds to the MVD concepts.

Discussion and Conclusion

This paper outlines the implementation of a new Cloud-BIM interoperability framework known as BIM Synapse. In fact, this study contributes to standardization of BIM interoperability in Cloud and to understanding a new approach to a standard-based BIM interoperability in Cloud. BIM Synapse uses a network-based BIM data transmission and can address: (1) what data to request in the receiving application by deploying IFC MVD specification, (2) how to encode and store BIM data to be transmitted properly by deploying web technologies, and (3) how to utilize received BIM data by using IFC data model. In this architecture, BIM data for Cloud collaboration is stored in a data-store within sending application and can be accessed through a standardized API. BIM data can be interpreted by a collaborating application as the receiving application. This API in fact, provides an on-demand access to the pool of data on sending application (Fielding, 2000). Accordingly, data request happens directly in the receiving application that uses a set of data exchange requirements based on MVD specification to request for the BIM data.

The dataflow introduced in BIM Synapse framework can address current challenges of BIM data exchange in the Cloud by providing a loosely coupled network-based data interoperability approach (Afsari et al, 2017a). The dataflow for Cloud-BIM data exchange based on BIM Synapse utilizes true potentials of the web technologies. Standard APIs make Cloud-based application more accessible (Lewis, 2013; Loutas, 2011) and BIM Synapse dataflow can perform as an enabler for a collaborative process allowing Cloud-BIM services to communicate through their standardized APIs. Thus, BIM Synapse can address challenges of current BIM interoperability. Also, the use of RESTful IFC API in BIM Synapse makes the framework highly scalable and simple (Fielding, 2000).

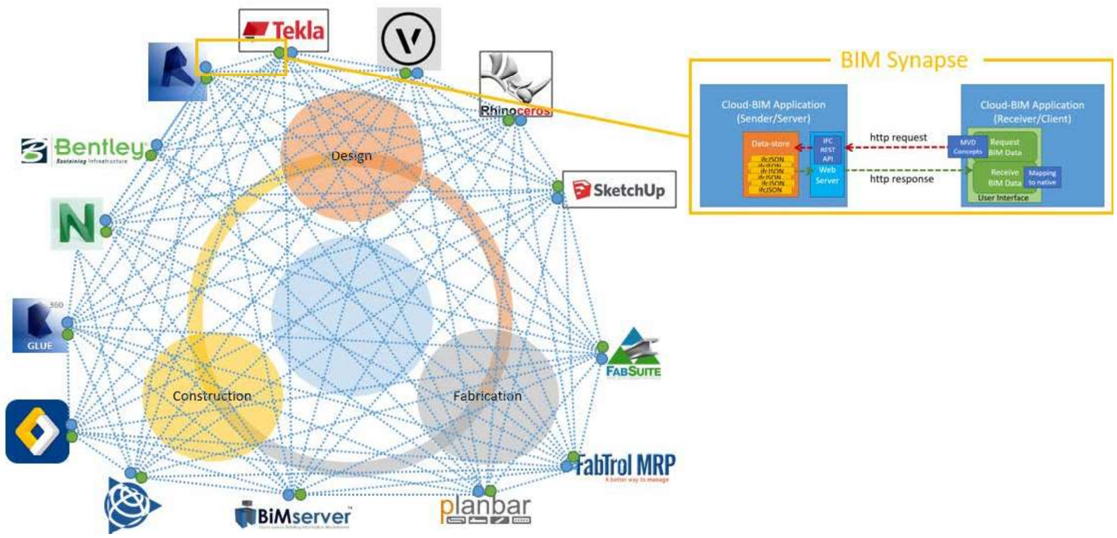

Figure 5: An ecosystem of the BIM Synapses during the project lifecycle

Overall, BIM Synapse framework emphasizes on a loosely coupled collaboration and uses standards to design and implement the framework. BIM Synapse can capture domain knowledge by using MVD specification to retrieve REST resources. Since BIM Synapse uses HTTP as the main data transfer protocol, it creates a network-based data transmission. Moreover, BIM Synapse can enable the integration of IoT with BIM model. IoT requires network connectivity and provision of resources in the application layer on top of the network connectivity (Wilde, 2007). The IFC REST API in BIM Synapse makes BIM resources accessible through standard web-based interactions and therefore, it enables the integration of BIM data with IoT technologies. Implementation of BIM Synapse framework will transform Cloud-BIM interoperability to an ecosystem of synapses, shown in Figure 5, that will reshape collaborations in the AEC industry and will enable the communication capabilities required for IoT. Future research involving the validation of the implemented framework will outline the design of the IFC REST API and will also recommend required revisions to the IFC specification to fully address Cloud-BIM interoperability.

Acknowledgement

The author would like to thank Professor Chuck Eastman, Dr. Daniel Castro, and Dr. Dennis Shelden for their helpful recommendations in this study.

References

Afsari K., Eastman C., Shelden D. (2017a). Building Information Modeling data interoperability for Cloud-based collaboration: Limitations and opportunities. *International Journal of Architectural Computing*.

Afsari K., Eastman C., Castro D. (2017b). JavaScript Object Notation (JSON) data serialization for IFC schema in web-based BIM data exchange. *Automation in Construction*. 77(1). 24-51

Afsari K., Eastman C., Shelden D. (2016a). Cloud-based BIM Data Transmission: Current Status and Challenges. Proceedings of the 33rd International Symposium on Automation and Robotics in Construction (ISARC), Auburn, Alabama.

Afsari K., Eastman C., Shelden D. (2016b). Data Transmission Opportunities for Collaborative Cloud-Based Building Information Modeling. Proceedings of SIGraDi 2016, XX Congress of the Iberoamerican Society of Digital Graphics, 9-11, November, Buenos Aires, Argentina.

Afsari K. and Eastman C. (2016c). Consolidated Exchange Models for Implementing Precast Concrete Model View Definition. Proceedings of the 33rd International Symposium on Automation and Robotics in Construction (ISARC), Auburn, Alabama.

buildingSMART International (2008). IFC Overview summary. buildingSMART International. URL http://www.buildingsmart-tech.org/specifications/ifc-overview.

Cattell R. (2010). Scalable SQL and NoSQL Data Stores. *ACM SIGMOD Record*, 39(4), 12-27.

Eastman C., Panushev I., Sacks R., Venugopal M., Aram S., See R. and Yagmur E. (2011). A Guide for Development and Preparation of a National BIM Exchnage Standard. Charles Pankow Foundation.

Fielding R. (2000). Architectural styles and the design of network-based software architectures, University of California, Irvine.

Fielding R., Gettys J., Mogul J., Frystyk H., Masinter L., Leach P., Berners-Lee T. (1999). Hypertext Transfer Protocol -- HTTP/1.1, Network Working Group.

Gong C., Liu J., Zhang Q., Chen H. and Gong Z. (2010). The Characteristics of Cloud Computing. Proceedings of the IEEE: 39th International Conference on Parallel Processing Workshops, San Diego, CA.

IBM (2006). TCP/IP Tutorial and Technical Overview. International Technical Support Organization.

Jadeja Y. and Modi K. (2012). Cloud Computing - Concepts, Architecture and Challenges. Proceedings of the international Conference on Computing Electronics and Electrical Technologies, Nagercoil, India.

Juan D. and Zheng Q. (2014). Cloud and Open BIM-Based Building Information Interoperability Research. *Journal of Service Science and Management*, 7, 47-56.

Kolb, B., Muhammad, A., & Gibb, R. (2011) Searching for factors underlying cerebral plasticity in the normal and injured brain, Journal of Communication Disorders.

Lewis G. (2013). The Role of Standards in Cloud Computing Interoperability. Proceedings of the IEEE: 46th International Conference on System Sciences, Hawaii.

Loutas N., Kamateri E. and Konstantinos T. (2011). Cloud computing interoperability: the state of play. Proceedings of the IEEE: Third International Conference on Cloud Computing Technology and Science (CloudCom), Athens, Greece.

NIBS (2015). National BIM Standard- United States. Version 3. National Institute of Building Sciences.

NIST (2014). US Government Cloud Computing Technology Roadmap, Volume I and II. National Institute of Standards and Technology.

Nisbet N. and Liebich T. (2007). ifcXML Implementation Guide. International Alliance for Interoperability.

Schenck D. and Wilson P. (1994). Information Modeling: The EXPRESS Way, Oxford University Press.

Venugopal M., Eastman C., Sacks R. and Teizer J. (2011). Improving the Robustness of Model Exchanges Using Product Modeling Concepts for IFC Schema. *Computing in Civil Engineering*, 611-618.

Wang G. (2011). Improving Data Transmission in Web Applications via the Translation between XML and JSON. Proceedings of the IEEE: Third International Conference on Communications and Mobile Computing, Qingdao, China.

Wilde E. (2007). Putting Things to REST. UCB iSchool Report, Berkeley.

Wong J., Wang X., Li H. , Chan G. and Li H. (2014). A Review of Cloud-Based BIM Technology in the Construction Sector. Journal of Information Technology in Construction, 19, Special Issue: BIM Cloud-Based Technology in the AEC Sector: Present Status and Future Trends, 281-291.