University of Waterloo
Faculty of Engineering
Department of Civil and Environmental Engineering

Improving Interoperability in Increasingly Fragmented AEC Workflows

Self-study

Prepared by
Mapendo Ngilinga de Carvalho
20801728
mpngilin@uwaterloo.ca
2B Architectural Engineering

15 September 2020

Executive Summary

The following report investigates possible information systems for better workflows in the industry of architecture, engineering, and construction. The scope of the report encompasses the system architectures, tools, and frameworks involved in the implementation of such workflows.

As buildings become increasingly complex, teams become larger and more divers. To manage the substantial increase in data produced by these teams, a multitude of workflows have been developed across the industry. However, not all workflows are equally effective in handling this relatively new informational complexity and volume. A new paradigm of construction workflow tackling information fidelity and interoperability is crucial for the AEC industry to scale along side technological advancements.

Over the years, architecture and engineering firms have found various approaches to address this issue: proprietary ecosystems developed by software companies like Autodesk, custom software plugins created in-house, the integration of semantic web technologies, restructuring building information in the object model, and integrating cloud technologies with transaction-based protocols. By assessing the strengths and weaknesses of each method in addressing the problems latent in current workflows, we can begin to determine the characteristics of an effective AEC workflow, and architect the system of tools and technologies to successfully implement one.

An overview of the evolution of AEC workflows and of the major technological trends outside of the industry suggests that an open-source, transactional, object-based data model is the best way to approach modern workflows to maximize information fidelity and interoperability across disciplines. If feasibility of implementation and maintenance are a concern however, a locally stored model-based approach without transaction capability is recommended instead.

Table of Contents

List of Figures

List of Tables

1 Introduction

As buildings grow in complexity, so do the teams tasked with their design. One of the most notable aspects of any undertaking in the architecture, engineering, and construction (AEC) industry is the shear volume and complexity of the information produced and managed over a building's lifecycle. At any given phase in the design and construction process of a building, a multitude of experts can be collaborating simultaneously, each relying on their own specialized tools and software to model, simulate, document or visualize a part of the project. Essential to the progress and coherence of every AEC project is a workflow which allows the information from these different streams to converge seamlessly.

Building Information Modelling (BIM) is a popular workflow paradigm that arose in recent years to address the flow of building information. BIM tools allow experts from different fields to contribute to a single model of a building, by storing various properties of building elements in one central file. A BIM element may store geometric data, material properties, structural information, energy loads and more.

Though these tools improved the way we model buildings, they did little to address the issues of interoperability between specialized software and the flow of information across disciplines (Becerik-Gerber & Rice, 2010). This lack of interoperability in the AEC workflows may be one of the contributing factors in the notoriously slow rate of productivity in the construction industry (McKinsey&Company, 2017).

In an effort to address the question of interoperability in AEC workflows, recent studies have presented different data management paradigms to fundamentally transform not only the way data is exchange, but also the way it is structured and stored. Our aim will be to explore the different methods of data management that have emerged in industry and in academia to better understand the opportunities and challenges involved in improving interoperability in AEC workflows.

2 Data Management Paradigms

Ubiquitous to interoperability in any design workflow is the structure and rules that govern the data within it. Over the past few years, the AEC industry and academia have considered many different approaches to address the flow of information between the ever-increasing pool of specialized software. Encompassing these solutions are five data management paradigms that employ fundamentally different approaches to managing building information.

The earliest concept for exchanging data effectively is that of proprietary ecosystems and data standards introduced more of less at the same time as BIM. That idea was soon followed by a system involving specialized connector plugins, which began opening up more possibilities for true interoperability. At his point, academia began taking more interest in the concept of interoperability and the idea of integrating semantic web technology and object models emerged. Bringing this idea even further is the paradigm of the transactional model of information exchange.

No paradigm is perfect however, and an in-depth review of the literature surrounding these ideas allow us to better understand their respective strengths and weaknesses.

2.1 File-Based Proprietery Ecosystems & IFC

One of the first approaches to achieve interoperability between the different fields of AEC was the development of file sharing within closed software ecosystems. Autodesk emerged in 1982 and quickly became a leader in the industry by offering a large portfolio of interconnected tools for engineers, architects, and designers (Autodesk Inc., 2018). However, Autodesk achieved this level of interoperability using their own proprietary file formats (such as RVT for BIM and DWG for CAD) that could only be used within their ecosystem. As more BIM software emerged, this level of interoperability wasn't sufficient, and a more universal BIM standard had to be developed (De Meyer, Van Campenhout, & Pauwels, 2010).

The task of standardizing BIM was undertaken by buildingSmart (formerly International Alliance for Interoperability) with their development of the Industry Foundation Class (IFC) (Liebich, et al., 1996). This standard offered a schema for storing rich BIM data which allowed software developers to create IFC compliant applications to read and write to this universal data format, further improving interoperability in the industry. However, a recent study looking at the advantages and limitations of BIM software packages demonstrates that in practice, the efficacy in exchanging BIM information using the IFC file format will depend on many factors such as the software's translator functions, internal data configurations, and the range of data exchanged (Iapige De Gaetani, Mert, & Migliaccio, 2020). The same study illustrates this point by transferring a 4D BIM model across industry BIM software packages and evaluating the level of interoperability (consistency in the information across platforms) observed for different BIM elements.

Table 1. Assessments of BIM technologies tested for interoperability with the IFC standard (Iapige De Gaetani, Mert, & Migliaccio, 2020)

Importing Software	ACCA Edificius	Autodesk Navisworks	Synchro Pro	ACCA USBviewer+	Graphisoft Solibri
Name	Good	Good	Good	Good	Good
Geometry	Adequate	Adequate	Adequate	Adequate	Adequate
Material	Inadequate	Good	Good	Good	Good
Phase	No Transfer	Adequate	Adequate	Adequate	Adequate

In order to preserve data integrity in increasingly large and complex IFC files, different teams will sometimes duplicate the data and work on it independently (Vonthron, Kock, & Kong, 2018). This, along with the inherently large amount of properties stored within an IFC file, results in files so large that they can become hard to manage (Bazjanac, 2002). Since BIM models recorded in the IFC standard can only be transferred by sharing the entire document, excessive file size quickly becomes a major interoperability issue and can cost a team a lot of time and money (Iapige De Gaetani, Mert, & Migliaccio, 2020). This is one the greatest limitation of the file-based interoperability paradigm.

Despite the imperfections associated with a file-based exchange system for building information, IFC remains the most widely adopted openBIM standard today (Jiang, Jiang, Han, Wu, & Wang, 2019).

2.2 Connector Plugins

In order to avoid full BIM model file transfers, a more specific approach to data sharing emerged in industry-grade software packages. Companies such as Autodesk, Unity, and

McNeel & associates began exposing application programming interfaces (APIs) for their products, allowing programmers to create plugins that tap directly into the software application[1] [2] [3]. The second paradigm of interoperability consisted of custom plugins design to funnel specific information from on application to another. Unity Reflect (for real-time visualizations of Revit, Rhino3D, and SketchUp models in Unity[4]) and Rhino Inside (to run Rhino and Grasshopper projects in Revit or AutoCAD[5]) are some examples in the AEC industry of custom plugins to funnel only essential data from one specialized software to another.

Though these solutions provide a more streamlined approached to interoperability, there is not enough literature on the matter to support a superior speed using this technique over file-based transfer. The speed of information exchange in this manner will also greatly vary across different plugins due to the difference in their software architecture.

Additionally, this paradigm is not easily scalable, as a new connector has to be developed or a previous one modified manually for every new purpose and for every combination of software package.

---

[1] https://developer.rhino3d.com/api/

[2] https://www.revitapidocs.com/

[3] https://docs.unity3d.com/ScriptReference/

[4] https://unity.com/products/unity-reflect

[5] https://www.rhino3d.com/inside

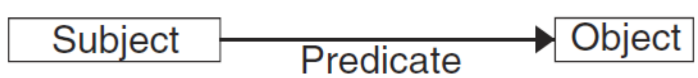
2.3 Semantic Web Technologies

The Semantic Web technology stack offers a promising alternative for maximizing interoperability with integrated data (Beetz & van Leewen, 2005). The third paradigm of interoperability attempts to use semantic web technologies to create a distributed Common Data Environment (CDE) for building information.

2.3.1 The Semantic Web

The Semantic Web can be thought off as an extension to the current world wide web. It is a development which seeks to make linked data on the internet machine-readable (Berners-Lee, Hendler, & Lassila, 2001). This is accomplished using the Resource Description Framework (RDF) standard, which links related data together in so-called triples consisting on a subject, a predicate, and an object (Werbrouck, Senthllvel, Pauwels, & Beetz, 2019). Together, these basic statements form a directed labelled graph of information with nodes representing the data and edges representing their relationships. Uniform Resource Identifiers (URIs) are then used to locate the data on the internet, allowing the information to live on different servers.

Figure 1. The triple form of an RDF statement: subject-predicate-object (Pauwels & Terkaj, 2016)



The semantic web's ability to give machines an understanding of the data and its relationship with other data on the internet stands in stark contrast with the traditional file-based indexing system of the conventional world wide web, which is why many industries have begun adopting semantic web-compatible systems of their own (Werbrouck, Senthllvel, Pauwels, & Beetz, 2019).

2.3.2 Increasing Interoperability using Semantic Web Technologies

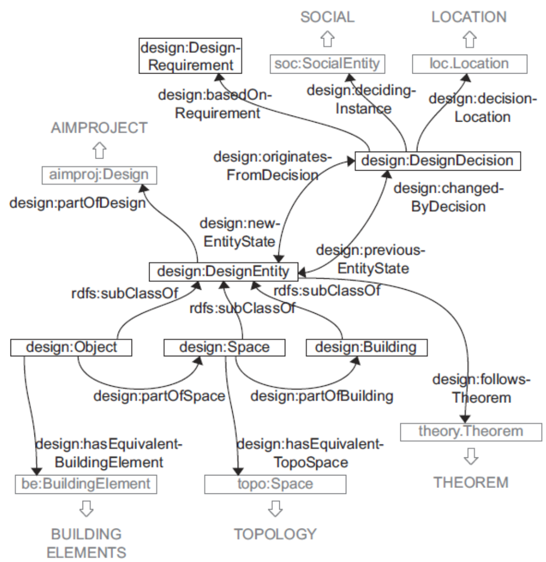
The Semantic Web have been touted as a game changer for interoperability in the AEC industry, due to its ability to link data across domains and to perform logical inferences and reasoning on building information (Pauwels & Terkaj, 2016). Specifically, a semantic representation of BIM data using the RDF syntax will contain links across disciplines and connect expert knowledge together, creating a more wholistic and complete representation of AEC projects (Werbrouck, Senthllvel, Pauwels, & Beetz, 2019). This effectively avoids the risk, time loss, and information damage associated with the duplication of information common in the traditional file-based data management paradigm.

The second advantage of linked-data over traditional domain-specific information models such as the IFC schemas is their modularity. While the IFC standards consists of a very large but fixed collection of vocabulary used to describe building information, semantic web technologies have the ability to incorporate all that vocabulary while remain modular and adapting to new topics as they emerge in the field (Werbrouck, Pauwels, Beetz, & van Berlo, 2019). With this approach, GIS, Facility Management, building regulations, and heritage data, to name a few, will be able to coexist in an interconnected web searchable and readable by both humans and machines (Werbrouck, Senthllvel, Pauwels, & Beetz, 2019).

In order to begin using semantic web technologies with BIM, a conversion between the most prominent BIM file format, IFC, and a semantic web ontology language (OWL) must occur. Several schemas have been suggested such as AIM and ifcOWL (De Meyer, Van

Campenhout, & Pauwels, 2010). These schemas have the shared goal of representing building information as connected graph for semantic web compatibility.

Figure 2: Graph Overview of the AIM ontology (De Meyer, Van Campenhout, & Pauwels, 2010)



Though the concept of integrating AEC information with semantic web technologies have received a lot of attention in academia, there remains a gap between theory and practicality (Werbrouck, Senthllvel, Pauwels, & Beetz, 2019). This is in part due to the steep learning curve associated with the use of Linked Data (Verborgh, 2018).

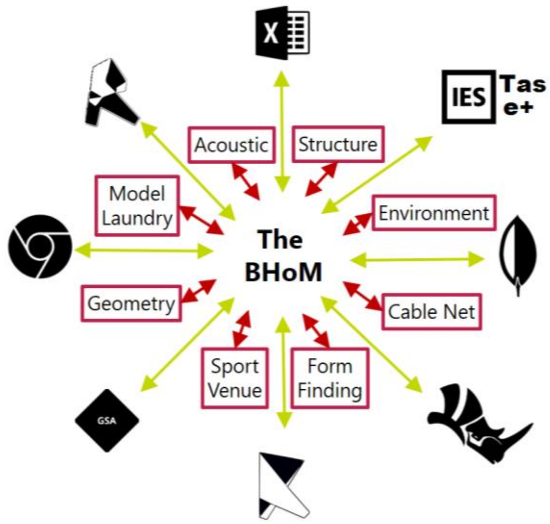
2.4 Central Object Model

Following the efforts of integrating building information with semantic web technologies, a new paradigm of data management emerged in the industry. The concept of an object model-based CDE has similar aspirations to that of semantic web technologies, but seeks to address the issue of implementation by making it more compatible with standard web technology stacks (Poinet, Stefanescu, & Papadonikolaki, Collaborative Workflows and Version Control Through Open-Source and Distributed Common Data Environment, 2020). Instead of storing linked data using RDF standards, a central object repository of a BIM model can store information of the BIM elements in classes and object of an object-oriented programming language.

One example of such model developed by directors at the engineering and design consulting firm Buro Happold[6] is the Building Habitats object Model (BHoM) (Buro Happold, 2019). Using a code base primarily written in C#, BHoM defines a set of classes and object for a common language that can be used to feed data into any BIM application, and only requires one translator plugin to be create per software package (Buro Happold, 2018). This paradigm keeps many of the elements of a semantic web ontology such as storing data and its relationships in a distributed network of servers, while also providing a more open platform for developers and designers to contribute to the code base (Buro Happold, 2018). This approach addresses the implementation issues faced by the Semantic Web and Linked Data paradigm, while keeping the idea of a Common Data Environment.

[6] https://www.burohappold.com/

Figure 3. Use Cases for The Buildings and Habitats object Model (BHoM) (Buro Happold, 2018)



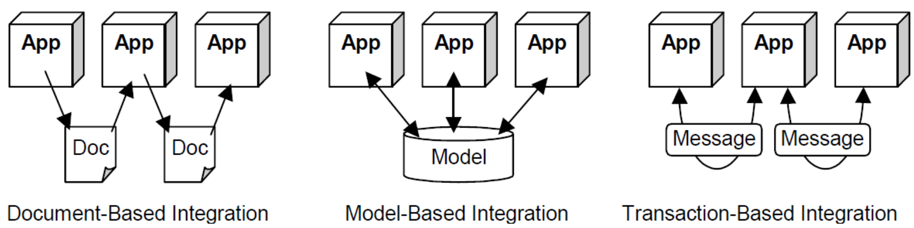
A notable factor of the BHoM is that it is completely open source. This means that the source code is available publicly, and the developers can easily contribute to it as they find new use cases or feature improvements (Buro Happold, 2018). This fosters the growth of a community dedicated to constantly improving and maintaining the code base, ensuring long-term interoperability.

While the object-based model has many advantages – especially when combined with an open-source philosophy – it exhibits a few shortcomings. Several issues remain unresolved such as the extensibility of models and version control (Froese, Yu, Liston, & Fischer, 2000). In addition, a shared model might not be favorable in every use case. Every user might not require access to the entire project and may be able to perform their duties with only a small subsection of the project data. Sharing the entire model to a large group of users may lead to preventable data corruption through human error.

2.5 Transactional Model

Transaction-based collaboration is the fifth and most recently explored data management paradigm. Where document-based exchange and object model integration seek ways to deliver all of a projects data to the users in an organized manner, a transactional approach involves the broadcasting of small data packets between software applications.

Figure 4. Alternative integration paradigms (Froese, Yu, Liston, & Fischer, 2000)



Traditionally, AEC software packages encompassed three main responsibilities (Froese, Yu, Liston, & Fischer, 2000):

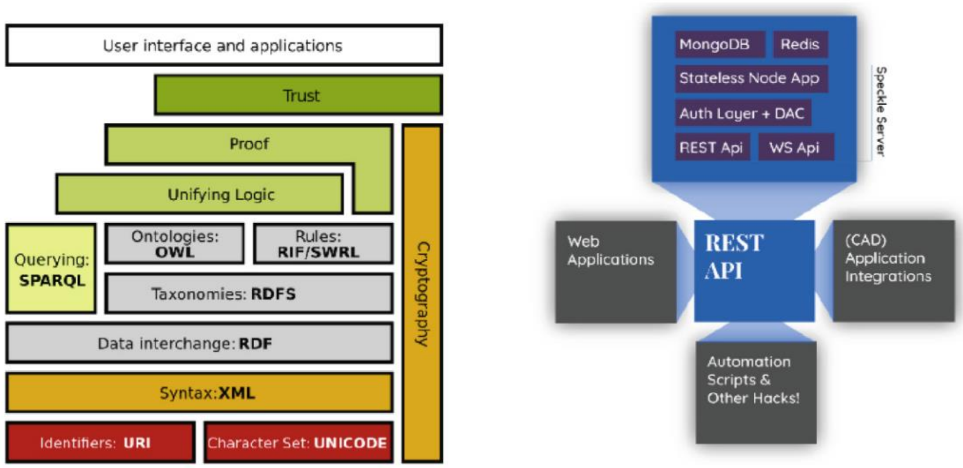
1. management and representation of domain data
2. application logic
3. information presentation (user interfaces)

While the first and second paradigms, file-based data management and custom plugins, made all these decisions for the user, the third, and forth paradigms attempted to abstract away the management and representation of data by storing them in triples or objects on distributed networks. With transactions however, messages broadcasted between software can contain changes to the domain data, new logic or functionality, and even request changes in the views or perspective of a user interface (Froese, Yu, Liston, & Fischer, 2000). Furthermore, transactions can be documented as they occur, and the history of changes can be used to revert back to previous versions of the project, effectively introducing version control to AEC workflows. This renders this approach highly flexible and adaptable to different challenges as they arise in the industry.

This approach builds on top of an object model, by orchestrating the communication between the software applications and a central object model server. Speckle[7] is a transaction based distributed CDE framework for the AEC industry born out of research conducted at the University College London that implements this exact model (Poinet, Stefanescu, & Papadonikolaki, 2020). Inspired by the modularity of the semantic web, Speckle implements similar functionality through a Representational State Transfer Application Programming Interface (REST API). It uses its own schema, the SpeckleObject, to organize building information, rather than the RDF Schema (RDF-S) used in semantic web ontologies.

[7] https://speckle.systems/

Figure 5. The Semantic Web Stack (left) and the Speckle Technology Stack (right) (Poinet, Stefanescu, & Papadonikolaki, 2020)



Like BHoM, Speckle is an open-source project, inviting many contributors from different industries to actively participate in the growth and improvement of its platform (Poinet, Stefanescu, & Papadonikolaki, 2020).

3 Conceptualizing Workflow 4.0

Digital technology is transforming the way our industries work. With the rise of the internet-of-things era, and the proliferation of data running through our machines, a new type of industry seeking to maximize on these technologies called industry 4.0 is emerging.

While industries around the world have successfully transformed their models of operation with industry 4.0 technology and are experiencing a rapid increase in their productivity rate, the construction industry lags behind (McKinsey&Company, 2017).

As AEC firms begin appreciating the advantages of industry 4.0 technologies, or when the pressures of a rapid changing world caches up to them, the question will inevitable become: how do we transform our old AEC workflows inline with the principle of industry 4.0 technologies? In other words, what does an industry 4.0-compatible workflow (a workflow 4.0) look like?

In this paper we explore not only the characteristics a workflow 4.0, but the system architecture and technology stack needed to successfully implement it.

3.1 Method

Our method will consist of determining a set of criteria relevant to the topic of interoperable workflows. These criteria will be defined based on the knowledge accrued in the literature review. Once the criteria are set, a rank from 1 to 5 will be attributed to each alternative solution, based once again on the literature reviewed in the past section and also on the author's industry knowledge.

3.2 Characteristics of an Advanced Workflow

An advanced workflow will look different from firm to firm. However, to integrate well with industry 4.0 technologies, every advanced workflow must be highly interoperable to manage not only the increase in data but also the increase in data sources and new technologies associated with them. We lay out a set of criteria based on knowledge gained in the previous section of the report to define a workflow 4.0.

- Interoperability – Ability to transfer data, logic, and interfaces across software
- Information Fidelity – Ability to preserve information integrity during transfers
- Openness – Ease of contribution by the community
- Scalability – Ability to adapt to increases in project complexity
- Feasibility – Ease of implementation

With these criteria defining a highly interoperable workflow, we must now decide which data management paradigm should govern the system architecture for workflow 4.0. After associating a weight to each criterion based on perceived value, a decision matrix can be constructed to evaluate our alternatives with rankings from 1 to 5.

Table 2. Evaluating interoperability of data management paradigms

			Paradigms				
		Weight	1	2	3	4	5
Criteria	Interoperability	0.3	1	2	4	4	5
	Fidelity	0.2	2	5	4	4	4
	Openness	0.1	1	3	2	5	5
	Scalability	0.25	3	1	5	4	5
	Feasibility	0.15	5	4	1	3	2
	Total	1.0	12	15	16	20	21
	Weighted Total		2.3	2.75	3.6	3.95	4.35

Paradigms:1 – File-based, 2 – Custom Plugins, 3 – Semantic Web, 4 – Object Model-based, 5 – Transactional

3.3 Workflow 4.0 Architecture

From the decision matrix in the previous section, we concluded that the best paradigm upon which to build a highly interoperable workflow is the transaction-based model. Knowing this, we can begin laying out the system architecture of our workflow 4.0.

3.3.1 Source of Truth

Every AEC workflow requires a source of truth that represents the official, most updated information version of the project. Since transactional workflows are building upon object models, we can utilize BHoM[8] as the framework for storing all our BIM elements and project information. This software package can either reside on a shared network in the firms physical building space, or, for a decentralized architecture, on a remote server. To connect to a decentralize common data environment, an application programming interface will have to be written in order to communicate with the BHoM effectively.

For decentralized CDE however, Speckle[9] is a more promising alternative, as it is already presented as a RESTful API. It uses the SpeckleObject as the base class for all its data, and, like BHoM, is a versatile and open-source schema that can easily scale in functionality as needed.

[8] https://bhom.xyz/

[9] https://speckle.systems/

3.3.2 Collaboration

Collaboration in transaction-based data management system are done through broadcasted messages. In order to accomplish that, firms can use web sockets to open up ports between software applications through which information can be relayed asynchronously. This can either be accomplished in tandem with a remote BHoM server, or using Speckles integrate Stream functionality.

Speckle Streams allow live unidirectional flow of data across software platforms (Speckle, 2020). For example, a Revit workset can be modelling a wall assembly along a line provided by a Grasshopper script, which pulled GIS data from an online database to determine the most optimal placement of the path. In this example, each software is a node along a stream of data, and information is flowing from senders to receivers to create a seamless, highly interoperable workflow.

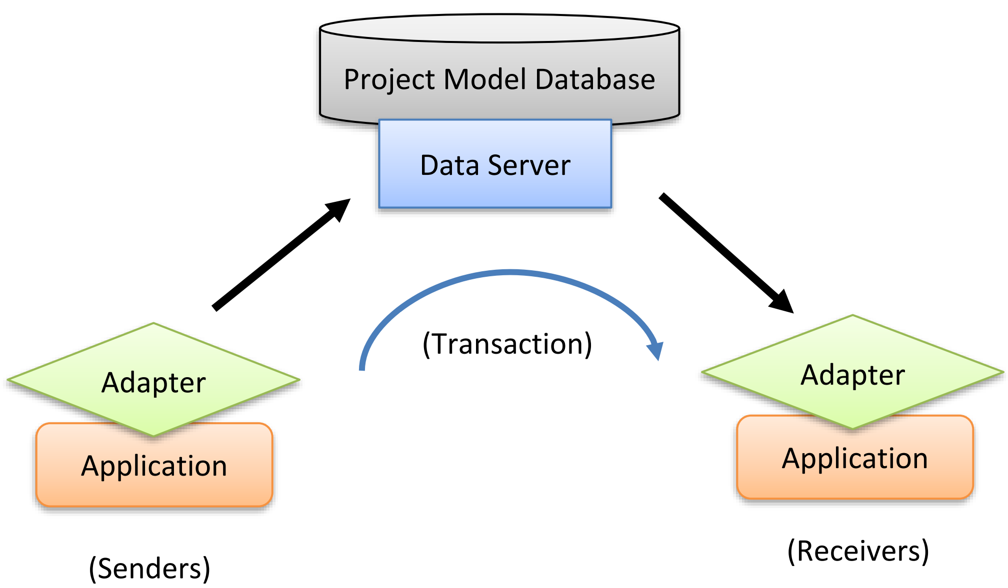
With either BHoM server or Speckle server, the transactional nature of the communication enables the use of AEC version control technologies. With BHoM, an additional software will have to be used to accomplish this task, for example 3D Repo, while with Speckle, version control functionality with a user interface is integrated in their open-source platform.

3.3.3 Community

Finally, the success of BHoM can be partly attributed to the fact that it was made open source early on (Buro Happold, 2019). Similarly, for a workflow to be effective in the sharing economy emerging from an industry 4.0 revolution, it must allow developers to contribute and improve upon it. Though this idea may be foreign in the AEC industry, it has shown great success in the software business, as exemplified by many widely used open source software such as the Linux operating system, the Blender 3D modelling software, and the python programing language to name a few.

Similar, a firm seeking to develop, and highly interoperable workflow should consider hosting the code on a public repository, welcoming contributions from members of all industries. This is crucial as an interoperable workflow will require a lot of diverse code to run the adaptors for every software package useful to the BIM industry. By open sourcing this endeavour, the community can help the workflow achieve high levels of interoperability faster, and ensure it continues to perform as new technologies reach the market.

Figure 6. Reference architecture for a distributed transaction-based workflow



3.4 Discussion

The transactional model is the best paradigm for data management in the AEC industry. By implement it with an open source, decentralized common data environment, a transactional workflow will ensure an overall greater level of interoperability, fidelity, openness, scalability and feasibility. For smaller teams however, it may be unnecessary to manage an entire server application as is required by a transaction-based data-management model. Instead, smaller firms an opt for an open-source, local object model approach, as this is the second most interoperable paradigm and is slightly easier to implement and maintain due to absence of a back-end architecture.

3.5 Conclusion

In conclusion, a firm seeking to adopt an industry 4.0-compatible workflow should adopt a transactional model, as this paradigm delivers the best results in terms of interoperability, fidelity, openness, scalability, and feasibility. For smaller firms, an object model-best workflow architecture is recommended, as it offers similar benefits while promising an easier implementation.

4 Conclusion

From the analysis in the report body, it was concluded that an open-source, distributed, common data environment combined with a transaction-based data management paradigm was the best approach to achieve highly interoperable workflow in the AEC industry.

Transactional information flow backed up with an object model has the advantage of being highly modular and flexible. This avoids the unnecessarily large file size produced by traditional file-based approaches. When offered as an open-source project, contributors from around the world can help create new functionality and improved interoperability, rendering this method highly adaptable to new technologies. In addition, the decentralized nature of transactions allows software users on different networks to collaborate, further increasing interoperability. Finally, transactions have to ability to relay not only updates to data, but also communicate logic and user interface decisions. If a transactional model is too difficult to implement, an object-model should be implemented first, as this step will facilitate the transition to a higher level of interoperability.

All in all, the advantages of a transaction-based data management paradigm can be leveraged in an industry 4.0 workflow, to drastically increase productivity in the AEC industry.

Bibliography

Autodesk Inc. (2018). *About Autodesk*. Retrieved September 7, 2020, from https://www.autodesk.ca/en/company

Bazjanac, V. (2002). Early Lessons from Deployment of IFC Compatible Software. *European Conference on Product and Process Modelling (ECPPM 2002).*

Becerik-Gerber, B., & Rice, S. (2010). The Perceived Value of Building Information Modeling in the U.S. Building Industry. *ITcon 15.*

Beetz, J., & van Leewen, J. P. (2005). An Ontology Web Language Notation of the Industry Foundation Classes. *22nd CIB W78 Conference.*

Berners-Lee, T., Hendler, J., & Lassila, O. (2001, May). The Semantic Web. *Scientific American*, pp. 1-4.

Buro Happold. (2018). *The Buildings and Habitats object Model*. Retrieved September 8, 2020, from https://bhom.xyz/

Buro Happold. (2019, November 13). *Digital Revolution in Engineering*. Retrieved September 8, 2020, from https://www.burohappold.com/articles/digital-revolution-in-engineering/

De Meyer, R., Van Campenhout, J., & Pauwels, P. (2010). Interoperability for the Design and Construction Industry through Semantic Web Technology. *International Conference on Semantic and Digital Media Technologies.*

Froese, T., Yu, K., Liston, K., & Fischer, M. (2000). System Architectures For AEC Interoperability. *Construction Information Technology.* Reykjavik.

Grilo, A., & Jardim-Goncalves, R. (2010). Value proposition on interoperability of BIM and collaborative working environments. *Automation in Construction*, 522-530.

Iapige De Gaetani, C., Mert, M., & Migliaccio, F. (2020). Interoperability Analyses of BIM Platforms for Construction Management. *Applied Sciences*(10), 15.

Jiang, S., Jiang, L., Han, Y., Wu, Z., & Wang, N. (2019). OpenBIM: An Enabling Solution for Information Interoperability. *Applied Sciences*(9), 27.

Liebich, T., Adachi, Y., Forester, J., Hyvarinen, J., Karstila, K., & Wix, J. (1996). *Industry Foundation Classes IFC2x3 Edition 3*. Retrieved September 7, 2020, from https://standards.buildingsmart.org/IFC/RELEASE/IFC2x3/FINAL/HTML/

McKinsey&Company. (2017, July 18). *Improving Construction Productivity.* Retrieved September 2, 2020, from https://www.mckinsey.com/industries/capital-projects-and-infrastructure/our-insights/improving-construction-productivity

Mendes de Farias, T., Roxin, A.-M., & Nicolle, C. (2015). Semantic Web Technology for Implementing Cost-Effective and Interoperable Building Information Modeling. *IE International Conference.* Dijon.

Ozturk, G. B. (2020). Interoperability in building information modeling for AECO/FM industry. *Automation in Contruction*.

Pauwels, P., & Terkaj, W. (2016). EXPRESS to OWL for construction industry: Towards a recommendable and usable ifcOWL ontology. *Automation in Construction*(63), 100-133.

Poinet, P. (2020). *Computational Extensibility and Mass Participation in Design.* London: UCL Press.

Poinet, P., Stefanescu, D., & Papadonikolaki, E. (2020). *Collaborative Workflows and Version Control Through Open-Source and Distributed Common Data Environment.* London: Bartlett School of Construction and Project Management.

Poirier, E. A., Forgues, D., & Staub-French, S. (2014). Dimensions of Interopterability in the AEC Industry. *Construction Research Congress.* Montreal.

Speckle. (2020, February 9). *Data Streams*. Retrieved September 8, 2020, from https://speckle.systems/docs/essentials/introduction/streams/

Thomassen, M. (2011). *BIM and Collaboration in the AEC Industry.* Aalborg: Aalborg University.

Verborgh, R. (2018, December 28). *Designing a Linked Data developer experience*. Retrieved September 8, 2020, from https://ruben.verborgh.org/blog/2018/12/28/designing-a-linked-data-developer-experience/

Vonthron, A., Kock, C., & Kong, M. (2018). Removing Duplicated Geometries in IFC Models using Rigid Body Transformation Estimation and Flyweight Design Pattern. *Visual. Eng., I*(6), 1-15.

Werbrouck, J., Pauwels, P., Beetz, J., & van Berlo, L. (2019). Towards a Decentralised Common Data Environment using Linked Building Data and the Solid Ecosystem. *36th CIB W78 Conference.* Newcastle.

Werbrouck, J., Senthllvel, M., Pauwels, P., & Beetz, J. (2019). A Checking Approach for Distributed Building Data. *31st Forum Bauinformatik Conference.* Berlin.

Whitmore, D., Papadonikolaki, E., Krystallis, I., & Locatelli, G. (2020). *Are Megaprojects Ready for the 4th Industrial Revolution.* London: ICE Publishing.

Glossary

AEC: Architecture, Engineering, Construction

BIM: Building Information Modelling

CAD: Computer Aided Design

CDE: Common Data Environment

GIS: Geometric Information System

IFC: Industry Foundation Class

Industry 4.0: A revolution in industry characterized by the rise in automation, internet-of-things technology, and machine-to-machine communication[10]

Ontology (Vocabularies): an organization of terms to describe a certain area of interest[11]

OWL: Web Ontology Language

[10] https://en.wikipedia.org/wiki/Fourth_Industrial_Revolution

[11] https://www.w3.org/standards/semanticweb/ontology