

# Automatic generation of ISO 19650 compliant templates based on standard construction contracts using a microservices approach.

Bower T.A, Rawdin A, Zhu X, Li H\*  
Cardiff University, United Kingdom  
[LiH@cardiff.ac.uk](mailto:LiH@cardiff.ac.uk) (Corresponding author)\*

**Abstract.** This study aims to establish a framework for automatically generating evidence for ISO 19650 certification. The study starts with an investigation of the challenges organisations face in compliance with BIM standards ISO 19650, the key areas of interest identified in relation to this are an organisation's ability to understand what their information requirements are. Once requirements have been identified, they are translated into format which is both machine and human readable. Extraction of text from existing project documentation is also investigated, proposing a microservice-based solution which formats and produces documents which meet the standards for information management requirements.

## 1. Introduction

The concept of Building Information Modelling (BIM) as a process-based information management framework is increasingly being adopted worldwide. There are many standards available defining BIM processes including the recent ISO 19650 series, of which there are currently 4 published parts (ISO, 2018a, 2018b, 2018c, 2020). This standard series is applicable to assets of all sizes and covers the lifecycle management of information from conception to demolition or re-purpose. The concept and principles of information management are designed in terms of BIM maturity, with stage 3 BIM requiring progression towards database and query-based environments.

This study begins with exploration of the concepts and principles of information management, which is divided into specifying, requesting, and delivering information. The collaboration of actors and how they work together is also a key aspect of the standards along with the production of standard methods and procedures.

In line with data-driven BIM stage 3 principles, this work aims to explore the concept of microservices for the purpose of assisting organisations in the AEC industry follow the information management guidance proposed by ISO 19650. This study starts with a requirements analysis of the ISO 19650 series to identify key challenges. This work then explains a framework for document information requirement schemas based on the analysis results, which goes on to inform a proposal for a microservice approach to project data collection and document generation. This study concludes with a summary of the research findings, along with discussions around of limitations of the proposed framework and future steps for improvement.

## 2. Requirements Analysis - Ethnographic Interviews with Industry

Information requirements and standards of information requirements are a key factor in asset management. BIM is the lifecycle management of asset information relating to not only how the asset is managed during its operational phase but also during the project delivery phase. This involves the process of information management and the data required to deliver and manage it. There are several information requirements during both phases. During interviews

held with 23 individuals including asset owners, operators and maintainers along with architects, engineers and building contractors throughout Wales from 2017 – 2020, many issues were raised in relation to building information modelling and how to meet the information requirements. From the asset owner’s perspective, there are many challenges in implementing BIM to existing assets (Abdirad and Dossick, 2020) in contrast to implementing BIM in new assets. Facility management systems must be able to capture data that is both relevant and delivered at the correct time to the appropriate actors.

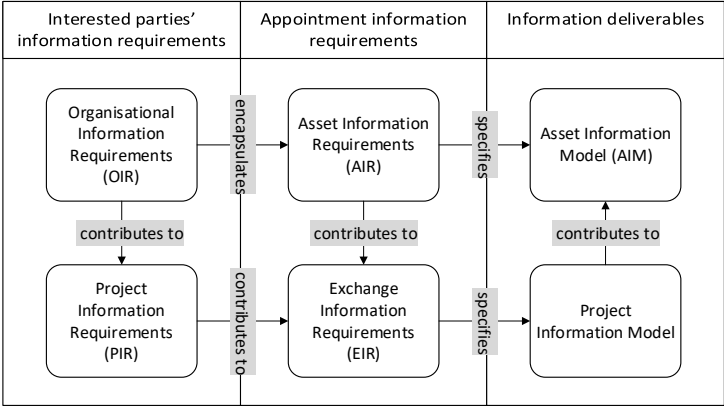


Figure 1: Hierarchy of information requirements as set out in ISO 19650-1 (ISO, 2018a)

Information requirements are specified in ISO 19650 (ISO, 2018a) as relating to one of four areas. The first; Organisation Information Requirements are high-level requirements describing information required for an organisation to run effectively. The second, Project Information Requirements are again a high-level information requirement which allows for a project to request information to answer questions. These questions are usually asked at key decision points of a project which in the UK align with various stages of RIBA scheme of works (Royal Institute of British Architects, 2020). The third requirement, Asset Information Requirements relate to information required about a particular asset during its lifecycle. The final requirement is the Exchange Information Requirements and allows an asset owner to specify how the exchange of information requirements is will occur. This requirement is responded to in the form of an execution plan by one or many of the lead parties termed Lead Appointed Party in the standard. The relationship of these requirements can be seen in Figure 1.

All the participants stated that whilst they are aware of the requirements within the ISO19650 standard, there are issues related to how the information requirements can be linked together from a practical implementation. During the interviews and case studies, open ended questions were used to prevent biased answers. Closed questions can cause issues when conducting interviews with individuals and study groups leading to bias in the responses (Nuno and St. John, 2014). The results of the interviews were collated and analysed using NVivo (QSR International, 2020). Of note amongst the interviewees and study groups was the responses given to the question “Tell me about your experience with information requirements and exchange and employer information requirements”. The responses from the interviewees aligned with each other in their responses. That is, the clients and professionals both had overall negative experiences. The clients' perspectives centred around two key themes: 1) They were unsure how to generate them. 2) They were unsure how they aligned with each other. From the professionals' experiences, the results centred around 1) The quality of information requirements 2) Clients did not understand the role of information requirements surrounding neither PAS1192 nor ISO19650.

From the interviews held with the parties, several key research questions emerged: 1) What are the challenges organisations face in collecting, storing, and reproducing the information

requirements? 2) What are the main requirements for ISO19650 compliant documentation? 3) What are the challenges organisations face in collecting, storing and reproducing the information requirements? 4) How can the requirements be addressed? And can generation of some requirements be automated?

### 3. Information Requirements Schema for ISO 19650 Documentation

Information requirements according to ISO 29481-1 (ISO, 2016) can be formed from defining processes that take place within an organisation. For this research, the key goal is to relate the information requirements together to allow for the flow of information from the Organisation Information Requirements through to the Exchange Information Requirements. All information requirements should be formed, requested, and responded to with a specific purpose. Previous work in this area (Heaton, Parlikad and Schooling, 2019) looked at forming function information requirements as a link between Organisation Information Requirements and Asset Information Requirements. It does not however look at how to link all information requirements between each other. Information requirements can also be formed by following the information schema as defined in ISO 29481-1 (Figure 2).

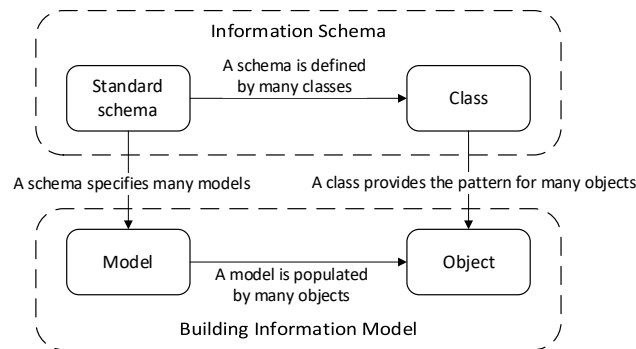


Figure 2, Development of information requirements (ISO, 2016)

For the work undertaken in this research, a simplified schema has been developed which uses activities undertaken at an organisation level to build information requirements that can be linked together using what is defined as information activity reasons. As an example: A local authority has an education department which has many schools. These schools undertake many activities which all require information. At the local authority level, they also undertake activities which require information. These activity-based information requirements have what are called information reasons. These information reasons are used as a link between the remaining information requirements and can be used to connect to questions in project information requirements as well as link them to a specific information delivery point within a defined schema plan of works such as RIBA within the UK or HOAI protocol in Germany.

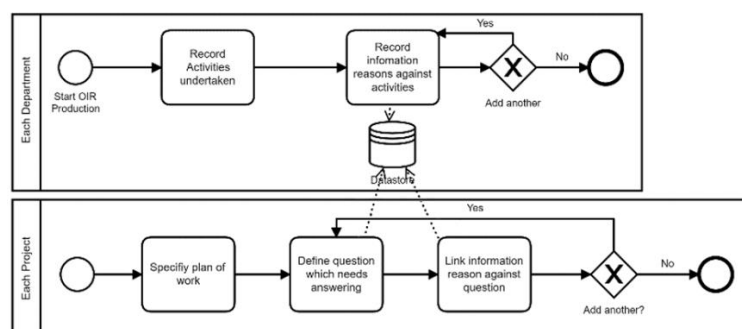


Figure 3: High-level data capture process for activity-based information requirements

The high-level information requirements data schema can be seen in Figure 3. This shows how each part of the information requirements are linked together along with the data schema for information requirements. From these high-level information requirements, a data schema was constructed for individual models based upon IFC. IFC is designated as an OpenBIM concept for data modelling. Some aspects were not able to be mapped against IFC and for this reason, an extension for the schema has been proposed which includes elements for questions and answers along with information reasons.

#### 4. Container-Based Microservice Architecture

In the context of moving towards UK BIM stage 3 and data-driven environments, there is an increasing need to explore flexible, lightweight, connected web services for management of information. Modularity and interoperability are key considerations to make when designing reusable infrastructure and several authors have made contributions to this idea for BIM applications. Previously studied use-cases for containerised microservice architectures in BIM include linked-data applications (Ferguson, Vardeman and Nabrzyski, 2016), Internet of Things (IoT) infrastructure for supporting building performance management (Kang, Lin and Zhang, 2018). For scalability, multiple nodes can be orchestrated for parallelisation of resource intensive tasks (Fahad and Bus, 2018).

Modularity can be achieved by isolating operations, assigning them suitable endpoints for data access. For example, one processing service can be used by multiple clients. There are several options available for deploying isolated web services such as virtual machines, cloud platforms, Openstack, Kubernetes, and Docker. The latter has been chosen for this work due to its relative simplicity for configuration and installation, and performance advantages over virtual machines which come from the ability for containers to share common resources (Chung *et al.*, 2016). Groups of images can be assembled and linked together using docker-compose files, allowing for simple and consistent installation and configuration of web services.

The broader aim of this project is to create a multi-standards BIM compliance checking environment, and eventually developing ‘meta’ standards for BIM compliance. The focus of this study is around BS EN ISO 19650, with a particular focus on project certification.

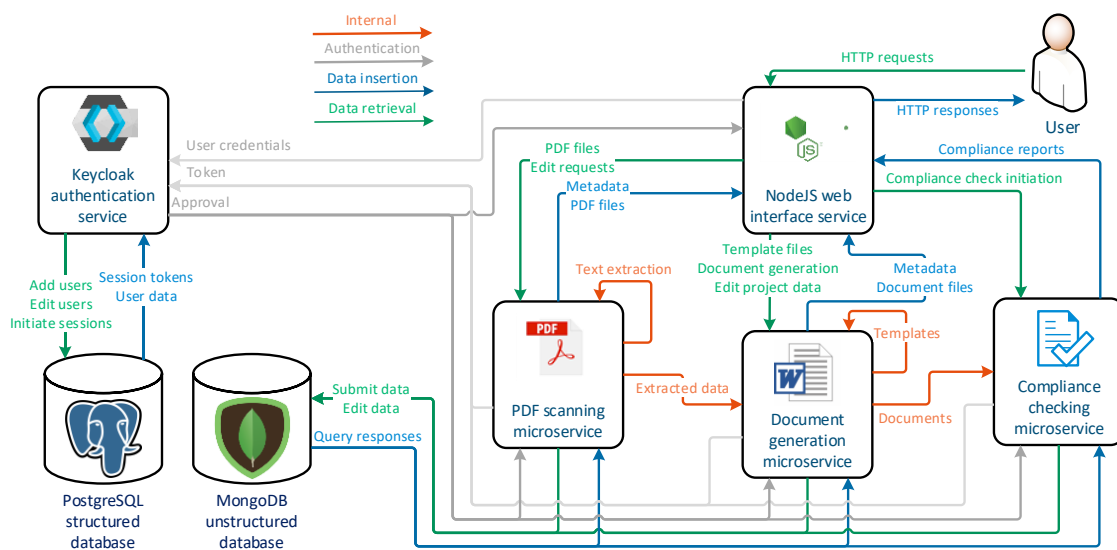


Figure 4: Overall system architecture for container-based infrastructure which incorporates automatic data extraction, document generation and compliance checking. Compliance checking is part of ongoing developments of this project

To address research question 4, this study proposes using microservice architecture with multiple services connected through Application Programming Interfaces (APIs) (Figure 4). The aim is to start with project documents with standardised structures such as contracts, and automatically produce documentation pursuant to ISO 19650 certification, for the purpose of performing in-house checks on the documents before final submission to a certification body.

### 4.1 Flask API Microservices

The key processing elements of the system used in this study are undertaken by two Flask microservices. Flask is a web framework for Python which runs as a lightweight web server. The Python library Flask Restplus is used as a wrapper for the Flask microservices. This allows concise definition of RESTful API interfaces, with automatic documentation of the API routes. REST (or Representable State Transfer) is a framework for structuring API endpoints. For a given resource (or URL), there are typically a limited number of requests available on individual items or groups of items, allowing adding, editing, deleting, viewing of resources. Data for the resources is stored in MongoDB collections, with user and project identifiers attached to all data to ensure data isolation between individuals and projects. Data is accessed in the Flask microservices using the Python library PyMongo.

### 4.2 NodeJS - Express Frontend Microservice

The frontend web service for this project is built using Express; a framework for NodeJS applications. NodeJS allows rendering of dynamic pages to present content from the database on the frontend web interface. Routes are defined for each resource, on the frontend, this typically takes the form of GET routes for rendering pages, or POST requests for submitting form data. For each route, the relevant requests can then be made to microservice API routes.

Specific organisational or project requirements are not necessarily known and cannot be fixed beforehand. Therefore, to embed flexibility into the system, HTML forms have been produced using a dynamic form generation JavaScript library called JSON Form (jsonform, 2020). Figure 5 shows a complex HTML5 form produced from two JSON schemas supplied using this library.

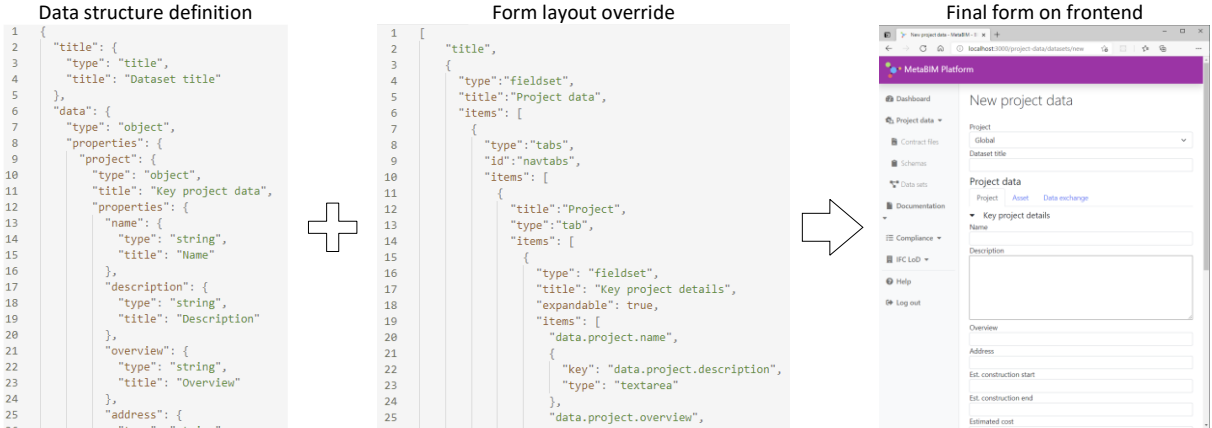


Figure 5: Form schema definition producing HTML5 friendly forms, overriding default options to create advanced layouts such as tabs and expandable fieldsets

Producing forms in this way allows the forms to easily be changed by those with even limited programming experience. Theoretically, this concept could be expanded to have a 'meta' form, generating the required form schemas and overrides. This will be considered in ongoing work.

### 4.3 Automatic Contract Scanning

The literature surrounding automatic extraction of document data is mature, and there are several approaches available. Generic methods exist for automatically converting semi-structured PDF documents into structured blocks of text (Chao and Fan, 2004). This can be taken a step further to extract and automatically classify blocks of text, for example in extracting known sections from research literature (Ramakrishnan *et al.*, 2012). Extraction from PDF files is less trivial than that of DOCX or HTML data due to its layout-based definition. Consideration needs to be made for size, spacing and alignment of characters and lines (Bast and Korzen, 2017). For example, detection of headings is can be performed through analysis of several thresholds including fonts, size and case (Budhiraja and Mago, 2020).

The contract scanning microservice performs tasks relating to extracting data from PDF files. It is wrapped in a Flask Python environment with a REST API. Within this API there are three main resources: Files, Pages, and Extraction Schemas. Files represent PDF files, and their associated metadata. The File route allows upload and download of files through the API.

The Pages resource represents the page text extracted from the PDF file. A Page resource is created by sending a POST request to the API with the File identifier, to initiate conversion. There are several options available to perform page text extraction, each performing with different accuracies (Bast and Korzen, 2017). In this study, PdfMiner is used, where each page is extracted and stored as a string in a JSON array. It is available as a Python library and performed well in a comparative review by Bast and Korzen (2017) which studied metrics such as missing or additional lines, words, or characters. The primary method for extracting data from contracts in this study is through text markers (Figure 6), where identifiable phrases in the contract are selected as markers denoting the positions of key values to be extracted.

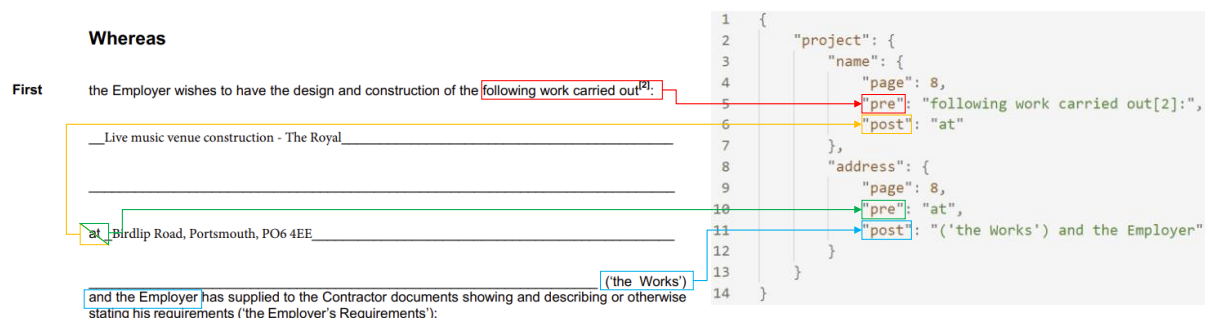


Figure 6: Extraction from JCT Design and Build 2016 contract with content, with mark-up denoting locations of fields

The text is extracted from the page string using REGEX (regular expressions) to search for two strings with a wildcard between. The expression which takes the place of the wildcard character is returned as the field value.

Punctuation which can appear in contract PDF files and can interfere with REGEX searches files has been stripped from both the page text and from the search markers. Alternatively, this REGEX issue could be resolved by converting the strings to escaped characters. To allow working with flexible, and deeply nested structures, a recursive object traversing function is used to navigate objects of any complexity. The function also allows for values to be manually specified, rather than scanned. This is convenient for addressing organisation specific requirements, or contract text which does not change.

### 4.4 Document generation algorithm

After the completion of contract scanning and extraction of the information related to the project, documents that fulfil the requirements of ISO 19650 can be generated automatically.

There are several available approaches for generating documents from templates. The most obvious being Microsoft Word’s built-in Mail Merge feature. In its default form, Mail Merge can be used for flat templates only, extracting data from relational or nested objects is not possible without custom modification through writing of macro subroutines.

There are libraries available for Python which allow creation of dynamic documents. Python-docx (Canny, 2013) is one such library, which allows creation of new documents and modification of existing documents. JSON dictionaries containing the project data can be manipulated in Python and written to a document using a template written purely in code. This approach is unlikely to be suitable for BIM project stakeholders, as it requires understanding of Python to implement and customise templates. A second library, python-docx-template (Lapouyade, 2019), builds on python-docx to create templates suitable for use with complex JSON data structures. Tags are written into documents using double braces, and sub-objects can be rendered using dot separators (Figure 7), and repeated data is rendered using loops (Figure 8).

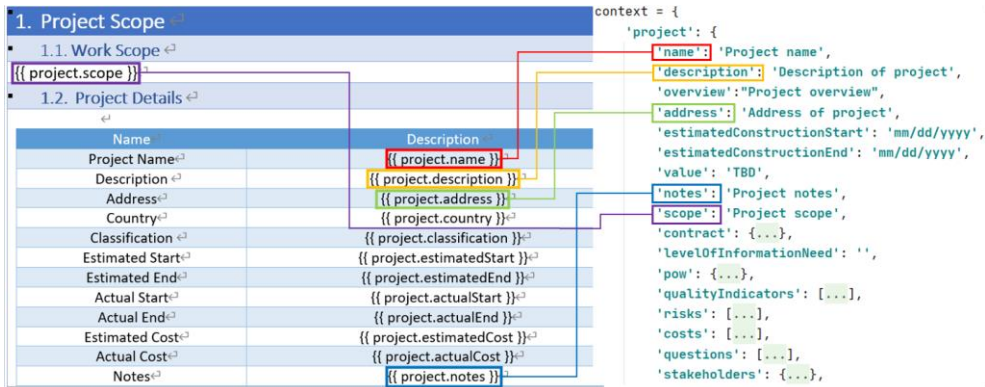


Figure 7: Association between tags in templates and key of the extracted data

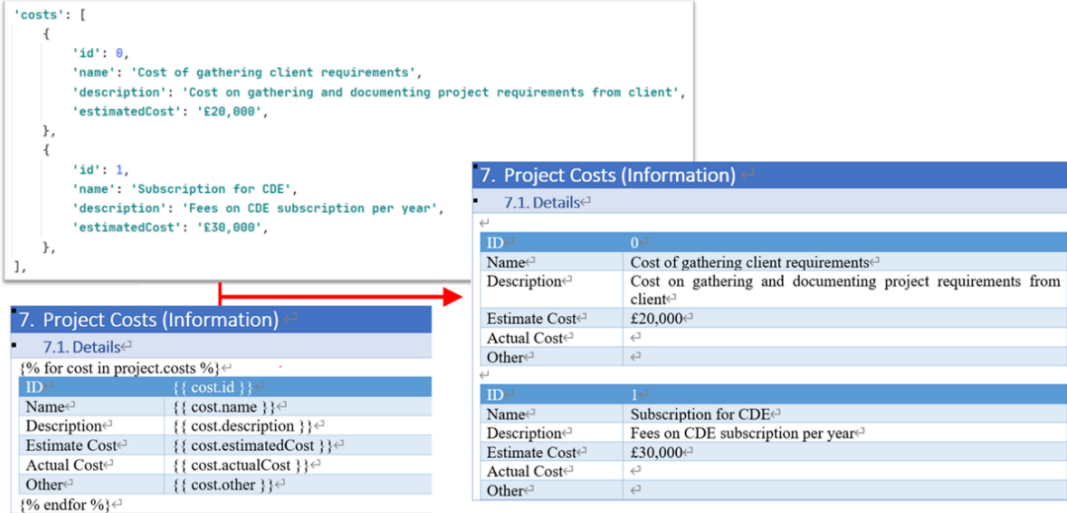


Figure 8: Generating repeated data from JSON list using a for loop in the template

Using a similar approach to the contract scanning microservice, this document generation algorithm is built into a microservice based on Flask and Flask RestPlus. The API allows uploading of templates, and creation of documents.

An important consideration for ISO 19650 compliance is the naming of documents. This is considered in the document generation engine by allowing the user to define naming conventions which extract pieces of information from the JSON schema. The naming

convention, as specified in ISO 19650-1 (ISO, 2018a) and the UK National Annex to ISO 19650-2 (ISO, 2018b), is implemented into the document generation API, where the fields, delimiters, field lengths, and blank character are defined using the JSON form JavaScript library, and sent to the document API and stored in the Mongo database. As the document is generated and downloaded by the user, the field names are extracted from the project JSON dictionary and the file name is assembled using the naming convention. Metadata is also added to the document using the python-docx library, allowing metadata fields such as author, status, revision to be specified from the web frontend.

In this study, templates and data structures for project information requirements, asset information requirements, and exchange information requirements were produced. Full templates for these three documents were produced using key project data extracted from the contracts as a basis. Additional project data is entered into the database using flexible web forms.

## **5. Discussion and Conclusions**

The first key objective of this study is to identify the key challenges faced within organisations and how technology could automatically produce the required documentation for ISO 19650 certification. The results of the surveys show that while organisations understand that they require information to comply with the required standard, they are unsure as to the method or suitable formats required to generate them. The organisations interviewed for this research were conducted over a period of 3 years for the Wales region within the UK. Although this may not be a representative picture of the whole of the UK, it shows that although BIM has been around for many years, there remain issues surrounding organisations' understanding of the BIM process and incomplete perception of BIM as a 3D modelling concept.

The development of a data schema which captures these high-level information requirements and transforms them into a machine and human readable format enables organisations to automatically comply with standards. The use of activity-based information requirements also allows generation of information requirements which can be linked together. This prevents organisations having from silos of unrelated information which make it more difficult to construct any required documents that potentially rely on related data.

In this study, a microservice-based architecture is proposed which addresses automated information requirements documentation authoring for ISO 19650 certification. Information extracted from contracts can be enriched with user supplied data using web forms. For JCT contracts, much of the data can be specified directly rather than extracted through markers, as many of the requirements are set out as static content. This approach allows organisations to produce consistent documentation, fulfilling requirements for ISO 19650 certification.

For the contract scanning microservice, the entire PDF is converted to raw text. Depending on the particular use-case there is scope to modify the algorithm to only extract the portions of text which are required on demand. If the required data is very sparsely arranged in the source PDF file, this approach may be more efficient. Use of optical character recognition (OCR) could also potentially improve the framework, allowing for extraction of scanned documents using pure OCR (Bast and Korzen, 2017), probabilistic methods (Hassan and Baumgartner, 2005), or through machine learning approaches (Budhiraja and Mago, 2020), with potential for including handwriting analysis (Baldominos, Saez and Isasi, 2018).

The framework set out for dynamically creating documentation is flexible due to its ability to be nested and recursive and to use filtering and cross referencing. This allows generation of



documents from complex data structures. The output from the contract scanning microservice and the web frontend, and the required inputs for the document generation microservice are compatible in structure. These structures can readily be expanded for different use-cases, as the system itself is designed without hard-coding any data structures.

This study also demonstrates the implementation of a containerised microservice-based architecture for BIM complementary services. As the construction industry moves towards stage 3 BIM, web-based services will become more essential. The containerised system used in this study is relatively straight forward to deploy on any operating system, and the system can theoretically be scaled for use in small or large organisations. For large-scale production environments, consideration needs to be made for high levels of traffic. When accessing resources which take time to produce, for example conversion of PDF files to text data, it would be necessary to use message brokering to route all requests through. Systems such as RabbitMQ or Redis can be used to handle queued requests. Cluster orchestration can also be used to scale up the performance and availability of web-based services. Further work in this project will assess in more detail the suitability for utilising container-based systems for use in industry.

## References

- Abdirad, H. and Dossick, C. S. (2020) ‘Rebaselining Asset Data for Existing Facilities and Infrastructure’, *Journal of Computing in Civil Engineering*, 34(1). doi: 10.1061/(ASCE)CP.1943-5487.0000868.
- Baldominos, A., Saez, Y. and Isasi, P. (2018) ‘Evolutionary convolutional neural networks: An application to handwriting recognition’, *Neurocomputing*, 283, pp. 38–52. doi: 10.1016/j.neucom.2017.12.049.
- Bast, H. and Korzen, C. (2017) ‘A Benchmark and Evaluation for Text Extraction from PDF’, in *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries*. Institute of Electrical and Electronics Engineers Inc. doi: 10.1109/JCDL.2017.7991564.
- Budhiraja, S. S. and Mago, V. (2020) ‘A supervised learning approach for heading detection’, *Expert Systems*, 37(4), p. e12520. doi: 10.1111/exsy.12520.
- Canny, S. (2013) ‘python-docx’. Available at: <https://github.com/python-openxml/python-docx> (Accessed: 15 March 2021).
- Chao, H. and Fan, J. (2004) ‘Layout and Content Extraction for PDF Documents’, in *International Workshop on Document Analysis Systems*. Springer, pp. 213–224.
- Chung, M. T. *et al.* (2016) ‘Using Docker in high performance computing applications’, in *2016 IEEE 6th International Conference on Communications and Electronics, IEEE ICCE 2016*. Institute of Electrical and Electronics Engineers Inc., pp. 52–57. doi: 10.1109/CCE.2016.7562612.
- Fahad, M. and Bus, N. (2018) ‘Conformance Checking of IFC Models via Semantic BIM Reasoner’, in *Proceedings of the 2018 European Group for Intelligent Computing in Engineering*.
- Ferguson, H., Vardeman, C. and Nabrzyski, J. (2016) ‘Linked data platform for building cloud-based smart applications and connecting API access points with data discovery techniques’, in *Proceedings - 2016 IEEE International Conference on Big Data, Big Data 2016*. Institute of Electrical and Electronics Engineers Inc., pp. 3016–3025. doi: 10.1109/BigData.2016.7840955.
- Hassan, T. and Baumgartner, R. (2005) ‘Intelligent text extraction from PDF documents’, in *Proceedings - International Conference on Computational Intelligence for Modelling, Control and Automation, CIMCA 2005 and International Conference on Intelligent Agents, Web Technologies and Internet*, pp. 2–6. doi: 10.1109/cimca.2005.1631436.
- Heaton, J., Parlikad, A. K. and Schooling, J. (2019) ‘A Building Information Modelling approach to the alignment of organisational objectives to Asset Information Requirements’, *Automation in Construction*, 104, pp. 14–26. doi: 10.1016/j.autcon.2019.03.022.

- ISO (2016) *Building information models — Information delivery manual — Part 1: Methodology and format*.
- ISO (2018a) *ISO 19650-1:2018 - Organization and digitization of information about buildings and civil engineering works, including building information modelling (BIM) — Information management using building information modelling — Part 1: Concepts and principles*.
- ISO (2018b) *ISO 19650-2:2018 - Organization and digitization of information about buildings and civil engineering works, including building information modelling (BIM) — Information management using building information modelling — Part 2: Delivery phase of the asset*.
- ISO (2018c) *ISO 19650-3:2020 - Organization and digitization of information about buildings and civil engineering works, including building information modelling (BIM) — Information management using building information modelling — Part 3: Operational phase of the, Bs En Iso 19650-1:2018*.
- ISO (2020) *ISO 19650-5:2020 Organization and digitization of information about buildings and civil engineering works, including building information modelling (BIM) — Information management using building information modelling — Part 5: Security-minded approach to i*.
- jsonform (2020) ‘JSON Form’. Available at: <https://github.com/jsonform/jsonform> (Accessed: 17 March 2021).
- Kang, K., Lin, J. and Zhang, J. (2018) ‘BIM- and IoT-based monitoring framework for building performance management’, *Journal of Structural Integrity and Maintenance*, 3(4), pp. 254–261. doi: 10.1080/24705314.2018.1536318.
- Lapouyade, E. (2019) ‘python-docx-template’. Available at: <https://github.com/elapouya/python-docx-template> (Accessed: 15 March 2021).
- Nuno, A. and St. John, F. A. V. (2014) ‘How to ask sensitive questions in conservation: A review of specialized questioning techniques’, *Biological Conservation*, 189, pp. 5–15. doi: 10.1016/j.biocon.2014.09.047.
- QSR International (2020) ‘NVivo’.
- Ramakrishnan, C. *et al.* (2012) ‘Layout-aware text extraction from full-text PDF of scientific articles’, *Source Code for Biology and Medicine*, 7(1), pp. 1–10. Available at: <http://code.google.com/p/lapdf/text/>. (Accessed: 12 March 2021).
- Royal Institute of British Architects (2020) *RIBA Plan of Work 2020 Overview*. Available at: [www.ribaplanofwork.com](http://www.ribaplanofwork.com) (Accessed: 15 March 2021).