DOI: 10.3217/978-3-99161-057-1-133

A methodology for data collection and aggregation in population-based structural health monitoring ecosystems

D. R. Lim[1, https://orcid.org/0009-0007-9068-9428], A. J. Ferguson[2, https://orcid.org/0000-0002-3794-705X], D. S. Brennan[3, https://orcid.org/0009-0001-7223-3431], D. Hester[2, https://orcid.org/0009-0007-2280-4340], R. Woods[1, https://orcid.org/0000-0001-6201-4270]

[1]School of Electronics, Electrical Engineering and Computer Science, Queen's University Belfast, Belfast BT9 5AH, U.K.
[2]School of Natural and Built Environment, Queen's University Belfast, Belfast BT9 5AG, U.K.
[3]Dynamics Research Group, School of Mechanical, Aerospace and Civil Engineering, University of Sheffield, Sheffield S1 3JD, U.K.
email: {dlim04, a.j.ferguson, d.hester, r.woods}@qub.ac.uk, d.s.brennan@sheffield.ac.uk

ABSTRACT: Population-based Structural Health Monitoring (PBSHM) provides insights based on data derived from comparing multiple structures' responses, providing a shift towards an integrated data domain. This presents significant challenges in data collection and integration of data across diverse structural populations, such as sensor systems, environmental data, and maintenance records and requires substantial engineering effort. This fragmentation of data across different formats and systems creates substantial engineering overhead when integrating new data sources, limiting the practical implementation of population-based approaches. This paper introduces a structured data flow architecture for systematic data collection and aggregation in PBSHM ecosystems by defining distinct functional components within the data collection process and enabling the structured integration of diverse data sources. The results establish a foundation for scalable PBSHM data collection, supporting the broader transition towards integrated structural health monitoring ecosystems.

KEY WORDS: Population-based Structural Health Monitoring (PBSHM); Data integration architecture; Data collection; Data aggregation

1 INTRODUCTION

Structural health monitoring (SHM) has the potential to reduce operational costs and increase infrastructure safety by augmenting existing primarily qualitative condition management processes using quantitative sensor data to track an asset's condition over time. More recently, population-based structural health monitoring (PBSHM) has been proposed to exploit the similarities between data from multiple structures to gain additional insights into their condition.

To date, the development of SHM sensing systems has largely been application-driven, such as in the Intersection Bridge 5 (IB5) [1] and the Telegraph Road Bridge [2] projects. As data acquisition (DAQ) systems were largely developed in isolation due to commercial interests, differing approaches to data measurement and transport for SHM exist. Consequently, there is no broadly agreed way to integrate these diverse, competing technical solutions into a PBSHM system. One potential approach is to develop a process that adapts all existing SHM sensing systems to support data representations compatible with current PBSHM data domains. However, this is largely impractical due to the complexity, cost, and time required to modify numerous existing systems.

To address this issue, this work proposes a methodology for the design, implementation, and operation of PBSHM data collection systems, allowing data to be merged from various sources to provide a unified view. This process is essential for analysing and making informed decisions based on comprehensive datasets in PBSHM.

A degree of data integration exists in existing SHM systems; for example, when utilising multiple vendors' sensors on a single structure. In this case, the operator may require that all the data be accessible from a single data system, therefore transforming data from multiple sources into a single database.

A common approach to transforming data makes use of an extract, transform, load (ETL) process, where the data is first extracted from the original data source, transformed into the target format and loaded into the new data store. The methodology proposed here permits the integration of sensing systems for a wide variety of structures (incl. bridges, wind turbines, masts, etc.) into a collection system for PBSHM data, that allows the aggregation and translation of data from a given structure to the shared PBSHM standard, in a defined, consistent manner. Additionally, the methodology is applicable to both existing and future systems.

Our architecture proposes a PBSHM integration pipeline that offers a flexible, modular approach to aggregating data from multiple existing SHM data sources. To demonstrate our work, we present a design study based on the existing IB5 and Telegraph Road Bridge monitoring systems, that illustrates how these may be integrated with a wider PBSHM ecosystem. This design study is presented using UML component diagrams due to its widespread use in system modelling [3].

The main contributions of this work are:

- Introduction of the concept of PBSHM integration pipelines for systematic data collection and aggregation in PBSHM ecosystems;
- An architecture for these integration pipelines, including functional definitions for the mandatory and optional components within these; and
- Demonstration of the application of the design principles of our approach to integrate two existing bridge monitoring systems into a PBSHM system.

The paper is structured as follows: Section 2 describes the current landscape of PBSHM research and identifying key challenges; Section 3 describes data pipelines that enable the integration of individual structure data collection systems; Section 4 describes the functionality in the pipelines; Section 5 details a design study that demonstrates the efficacy of the proposed architecture with conclusions given in section 6.

DOI: 10.3217/978-3-99161-057-1-133

2 BACKGROUND

To design a robust data architecture for supporting PBSHM, it is essential to first identify the benefits and requirements of PBSHM. Additionally, a thorough examination of the current landscape of SHM systems and existing data silos is necessary to pinpoint the needs and challenges for the data architecture.

2.1 PBSHM and the associated data ecosystem

As the availability of SHM data for a given structure is incomplete, a population-based SHM approach by collecting data from a group of similar structures to infer the condition of one structure. Bull et al. [4] showed that it is possible to represent the behaviour of these structures using a general form of the population that encapsulates behaviour of all structures within the population of "strongly homogeneous" or nominally identical structures. This is particularly useful for large populations that are manufactured identically and experience similar conditions, such as a farm of wind turbines where each turbine undergoes near-identical manufacturing and construction processes.

This concept, however, can be expanded to include structures that only share significant structural similarities otherwise known as homogenous populations [5], [6] with the challenge to identify those which are similar enough to transfer data between, without compromising model quality. Gosliga et al. [5] proposed an irreducible element (IE) model to represent such structures which solely captures the geometric properties of a structure, whereas a finite element (FE) model contains additional construction information. Brennan et al. [7] introduced an expanded IE model using a set of reduction rules which eliminate author ambiguity, ensuring that each is created using a consistent canonical form while maintaining all structural knowledge. Representing structures with IE models allows the creation of an attributed graph (AG) from the model, which can then be processed using a graph-matching algorithm to determine a "similarity score" between two structures.

Using this similarity score, it is possible to predict the possibility of positive data transfer across heterogeneous populations of structures. Gardner et al. [6] show that features and labels can be mapped from a source structure to a target structure, even among topologically different structures, by using IE and AG representations to extract the similarity between structures within a heterogeneous population.

Given the diverse and extensive data required for the successful realisation of PBSHM, meaningful comparisons between multiple data sources are needed. This requires the data to be standardised to allow large-scale analyses and efficient data processing. Attempts to extend open standards such as Bridge Information Management (BrIM) in Jeong et al. [8], look to address this shared-data problem within their specific regions, but extending to PBSHM is non-trivial.

As such, PBSHM has introduced a standard for its associated shared data. Brennan et al. [7] introduced a PBSHM technical ecosystem made up of the PBSHM Network, Framework, and Database. The Network is the shared data domain in which the similarity between structures is represented; the Database is the shared-data domain in which PBSHM data is stored in a common format and; the Framework is the computational domain in which all algorithms (both similarity and knowledge transfer) exist. Each domain is valid in its own right, but independent from others.

This comprehensive PBSHM ecosystem integrates data storage and software but can be expanded to accommodate larger databases and additional software modules. It may store various data categories, including sensor data, IE and FE models, reports, features, information and similarity metrics. The authors use a NoSQL database for the PBSHM ecosystem due to its increased flexibility over relational ones allowing the expansion of the database to accommodate any data that may be used in the future to develop PBSHM. Brennan et al. implemented the database using MongoDB, using a detailed "PBSHM Schema" to standardise and store the aforementioned data categories, ensuring compatibility and allowing efficient data retrieval and analysis. The flexibility provided by these choices allow for current knowledge to be embedded within the schema, however, allows for the future needs of PBSHM by enabling the adaptation of the schema to include yet-unknown data at a future date.

2.2 SHM data acquisition systems for civil infrastructure

Many SHM systems collect data for civil infrastructure and are made up of either one or multiple data acquisition systems. Various vendors can be used and provide comprehensive DAQs that manage data capture and storage, but these cannot be integrated across vendors, requiring the creation of bespoke SHM systems. Whilst this has the potential of creating a comprehensive system for an SHM structure, it may produce data that is incompatible for the purposes of PBSHM and will not easily be shared.

To highlight this, two bridges, the Intersection Bridge 5 (IB5) and the Telegraph Road Bridge have been selected, due to their extensive sensor networks and well-documented cyber-physical systems.

IB5 - IB5 is designed to continuously monitor the structural health of the bridge by recording and analysing various signals, using a variety of high-precision sensors that capture data on vibrations, strain, temperature, rotation, and other parameters. Data acquisition units collect the sensor data and convert it into a digital format suitable for real-time wireless transmission to a central server, which processes the data to detect any anomalies. The system includes an application programming interface (API) for authorized users to access the data remotely, facilitating ongoing monitoring and analysis. Additionally, power supply units ensure reliable operation of all components, even in harsh environmental conditions [1].

Telegraph Road Bridge - The Telegraph Road Bridge in Michigan employs a network of wireless sensors, including a variety of sensors that measure strain, acceleration, and temperature and are strategically placed to capture detailed data on the bridge's response to truck loading and thermal variations. Data acquisition units collect and digitize the sensor data, which is then transmitted to a central server. This bridge utilises a solar-powered wireless sensor network architecture that can also be used in hard environmental conditions [2].

Following the observations made on both systems [8], [9], [10], differences between the storage and representation between the two systems are described in Table 1.

DOI: 10.3217/978-3-99161-057-1-133

Although both monitoring systems may appear very similar (each having a physical layer, cyber-physical layer and data storage/processing capability), the systems differ in their underlying data management technologies. Both display a well-thought-out, cyber-physical architecture; however, the systems have been developed largely in isolation, with differing goals leading to a clear difference in data representations. This presents a challenge when trying to compare both structures in a PBSHM context for reasons discussed in section 2.1. Therefore, for PBSHM to be effective, it can be deduced that data integration needs to occur across the two structures.

Table 1. Comparing data representation between IB5 and Telegraph Road Bridge monitoring systems

Structure	**IB5**	**Telegraph Road Bridge**
Database Technology	PostgresSQL	Apache Cassandra
Sensor Information Storage	Database Entry	OpenBrIM
Sensor Information Schema	"ID", "Type" and "Location" fields in database	User-defined OpenBrIM Object

3 ARCHITECTURE

One of the key processes within SHM is gathering and capturing monitoring data to ultimately determine the health state of a structure. This is the same for PBSHM, but the process is compounded when considering the very nature of the population-based approach is to accumulate knowledge across multiple structures and types.

While the aforementioned PBSHM technological domains encompass knowledge when in a central ecosystem, they lack the understanding, definitions, and details of the procedures of migrating data from the SHM capture systems associated with the structures and the central system. This part focuses on proposing the missing link between the existing SHM systems – henceforth referenced as data generators – and the PBSHM technical ecosystem introduced by Brennan et al. [11].

3.1 Data Generators

A data generator is any entity that can produce data potentially valid within the PBSHM schema. As data generators can consist of existing SHM data acquisition systems, data generators can produce many different possible representations of SHM data. Therefore, to preserve this, data representation within the PBSHM data domains needs to be facilitated. This gives two possible options: a) development of a process to adapt existing data formats to representations that are supported in the existing PBSHM data domains or, b) retroactive adaption of existing SHM capture systems to support this as well as every future representation. This is significantly hindered by the need for commercial systems to remain compatible with existing solutions.

Thus, we define data pipelines which allow the transport of data from these existing, and future, data generators to the PBSHM data-domains.

3.2 PBSHM Integration Pipelines

The process of getting data from the aforementioned data generator into the PBSHM Framework is referred here to as a data pipeline. Pipelines describe the overall transmission of data from the location(s) at which the data is first introduced into the PBSHM domain to the desired end location.

3.3 Pipeline organisation

Through this data pipeline, it is important to denote the responsibility of each actor within this pipeline, by dividing it into sections. These divisions will furthermore be referred to as scopes with their own set of responsibilities, purpose and defined goals for any data requirements, transformations, and formats that may occur within its remit.

Whilst we must acknowledge that each data pipeline will be unique to the requirements of the data generator(s), there is still an abstract delimitation between each scope area. As such, this paper proposes the following aspects (see Figure 1):

Structure: The structure scope describes all the cyber-physical infrastructure required to capture information regarding a physical attribute of a structure. This scope provides an interface that allows this information to be provided to the next scope in the pipeline. Any solution implemented to this scope is specific to the set of measurements desired for a given structure.

Aggregation: The aggregation scope is defined by a generic set of processes that universally apply to every application to allow the communication of data from the structure scope to the PBSHM database and vice versa. It receives data from the previous scope and transmits this to the next.

PBSHM Network, Framework and Database: This establishes the shared data domain for which the relationships between structures, the shared domain in which PBSHM-specific algorithms and computations reside and the database where PBSHM data resides.

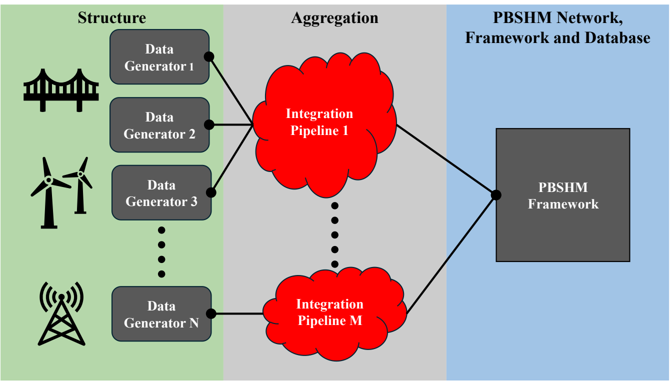

Figure 1. Pipeline flows through the aggregation scope from data generators to the PBSHM Framework

Due to the segregation of responsibility within the data pipeline, when acting within one scope, it is necessary to view the other scopes within the pipeline as black box systems. When implementing within the aggregation scope, it may not

DOI: 10.3217/978-3-99161-057-1-133

be possible to modify or understand the implementations and behaviours within the structure and PBSHM server scopes.

4 FUNCTIONALITY

To describe the functionality required in the pipeline, a roles-based approach is used. Each role dictates the functionality that is required to produce a valid PBSHM data integration system. Each role provides an interface through which data is communicated. Data flows describe the transfer of data from one role to the next. By utilising the abstract form of a roles-based approach to describe the implementation of data flows, we can define a consistent terminology within the context of the PBSHM ecosystem, allowing the methodology to apply to not just existing technologies, but future technologies. In principle, roles can be described as actors, which perform some operation on data within the pipeline.

Figure 2 outlines mandatory and optional roles. Mandatory roles (which are denoted in grey) must be implemented to create a valid data pipeline from a data generator to the PBSHM server and the optional roles (denoted in pink) describe additional functionality that can be added to the system. Arrows are used to describe the data flows where the arrowhead indicates which direction the data flow is initiated.

4.1 Mandatory Roles

Initially, there is some required functionality that must be implemented to allow the movement of data from a given data generator to the PBSHM core. It is important to note that data generators may encompass a wide variety of data i.e. channel data, feature data, reports, etc.

A pipeline may be configured as follows: Initially, data is generated by the data generator contained within the structure scope. These roles provide some interface to the aggregation layer.

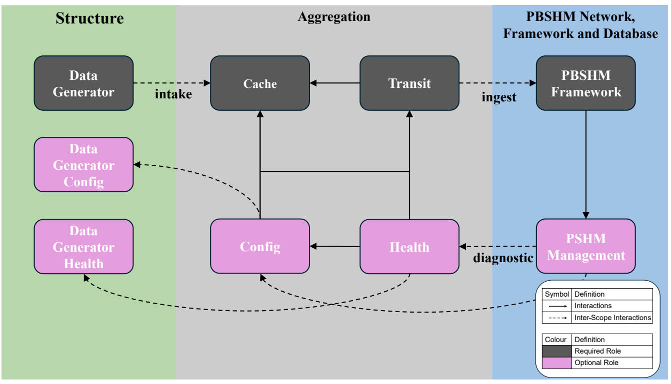

Figure 2. PBSHM data integration roles

Cache roles play a simple yet vital role in the PBSHM data integration pipeline. By utilising a buffer to store incoming data from data generators, we can aggregate multiple data streams at the cache to group the incoming data. In a PBSHM system where it can be expected to have many structures each with many data generators, it becomes infeasible to have each data generator directly interact with the PBSHM Framework.

Finally, the purpose of the transit role is to serve as the gateway between the data cache and the PBSHM Framework. Data provided by the transit role must be in the PBSHM schema format to allow compatibility with the PBSHM database. This will then perform redundancy checks, parity checks etc.

4.2 Optional Roles: Config & Health

Additionally, there is potential within the PBSHM integration pipelines to anticipate the need for reconfigurability and insight into downstream aspects of the data pipeline from the PBSHM Framework to allow effective data infrastructure management and support decision systems. Therefore, it is possible to introduce the health and config roles. The function of the health role is to provide statistics and system state information to the PBSHM framework whereas config provides an interface of the components within the aggregation and structure scope to be modified with supported configuration options.

Both health and config roles pose a significant challenge due to the unending complexity of both existing and future technologies due to both differing configurable attributes of data generators and differing statistics provided by data generators. Furthermore, cache roles can take multiple technical forms, with a further set of differing configurable attributes and properties on which insight could be desired.

For the most part, these roles are beyond the scope of this work and will be covered in more depth in future work.

5 DESIGN STUDY

The objective is to validate and demonstrate the effectiveness of the proposed data integration architecture for PBSHM by demonstrating the integrity of live data transfer from various sensor systems to a central PBSHM server. Also demonstrated is interoperability by integrating different sensor types and data formats into a cohesive pipeline. To reflect the existing landscape of SHM deployments, the design study uses the two examples given in Section 2.2. We first establish the existing methods that the systems use to access collected data.

5.1 Existing data retrieval mechanisms and assumptions

IB5 supplies a representational state transfer (REST) API (built using Fast-API) that allows researchers or stakeholders to access both stored raw and processed data. This enables end-users to access stored data of the digital twin, including raw sensor data from the physical implementation [9]. Although the cyber-physical system supporting the IB5 bridge is a comprehensive and well-designed example of developing a digital twin for a bridge, the REST API implemented for the IB5 is not publicly documented, highlighting the data integration challenges in the current SHM landscape.

DOI: 10.3217/978-3-99161-057-1-133

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="TimeSeriesCollection">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="TimeSeriesData" maxOccurs="unbounded">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="id" type="xs:string"/>
              <xs:element name="data">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="DataPoint" maxOccurs="unbounded">
                      <xs:complexType>
                        <xs:sequence>
                          <xs:element name="timestamp" type="xs:dateTime"/>
                          <xs:element name="value" type="xs:float"/>
                        </xs:sequence>
                      </xs:complexType>
                    </xs:element>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Figure 3. Assumed XSD of the API response from the IB5 monitoring system

As such, some assumptions have been made about the structure of their API queries and the resultant data that can be retrieved. The following fields are presumed necessary for the query: (1) Sensor IDs, and (2) Time window. It is assumed that the data is returned in an XML document is returned in the structure dictated by an XML schema definition (XSD) presented in Figure 3 where the values returned are:

- Sensor ID(s)
- An array of:
    - Timestamp
    - Value from sensor

An example of an object returned by the assumed API can be seen in Figure 4.

```xml
<TimeSeriesCollection>
    <TimeSeriesData>
        <id>sensor-1</id>
        <data>
            <DataPoint>
                <timestamp>2025-03-28T15:51:18Z</timestamp>
                <value>123.45</value>
            </DataPoint>
            <DataPoint>
                <timestamp>2025-03-28T15:52:18Z</timestamp>
                <value>125.67</value>
            </DataPoint>
        </data>
    </TimeSeriesData>
    <TimeSeriesData>
        <id>sensor-2</id>
        <data>
            <DataPoint>
                <timestamp>2025-03-28T15:51:18Z</timestamp>
                <value>223.45</value>
            </DataPoint>
            <DataPoint>
                <timestamp>2025-03-28T15:52:18Z</timestamp>
                <value>225.67</value>
            </DataPoint>
        </data>
    </TimeSeriesData>
</TimeSeriesCollection>
```

Figure 4. Example XML file based on the XSD of Figure 3

The monitoring system of the Telegraph Road also provides its own API to retrieve data. Its bridge information repository framework contains an Apache Cassandra database which provides the Cassandra Driver API for retrieving data. This used in conjunction with its "data mapper" (which maps the time series data stored in the database with the BrIM sensor information) provides the data necessary by the PBSHM schema via the "Sensor data retrieval" service which provides a REST API. This allows the system to be integrated into the PBSHM ecosystem. Literature on the Telegraph Road Bridge both details the structure of the API queries and provides examples of data returned [10].

5.2 The proposed data integration system

Using the information supplied about the two systems' APIs, and details inferred in section 5.1, it is possible to propose an integration system based on the architecture and roles detailed in sections 3 and 4. These roles are set out in Table 2.

This proposed integration system is defined in terms of a UML component diagram (shown in Figure 5). In this case, each component is a distinct software service. Relationships between the components which show how they interact, are drawn between each component. These components are grouped to detail in which deployment environment these would be implemented. Most have been implemented in the "PBSHM Integration Cloud Server"; however, an additional component has been added to the "PBSHM Integration Cloud Server" to show an additional module that could be added to the PBSHM Framework that would allow the insertion of records to the PBSHM Database over HTTP(S).

An important observation from Table 2 is the designation of the IB5 and TRB fetchers as data generators. Although these are part of the structure scope, in this case, they have been implemented in the "PBSHM Integration Cloud Server". As data generators must be the actors of any data transaction to the cache (Figure 2), it is necessary to add a component that fetches from the existing APIs provided by both bridges and subsequently pushes the results to their respective RabbitMQ caches. To avoid changing the existing back-ends of both structures, these are implemented in the "PBSHM Integration Cloud Server". However, this is an example of a deployment environment implementing roles across two scopes.

Table 2. Identifying how each role is fulfilled by components in the proposed PBSHM data integration system for IB5 and Telegraph Road Bridge

Role	IB5 Component	TRB Component
Data Generator	IB5 Fetcher	TRB Fetcher
Cache	IB5 Queue (RabbitMQ)	TRB Queue (RabbitMQ)
Transit	PBSHM Framework Loader	PBSHM Framework Loader
PBSHM Core	REST API	REST API

Within the proposed data integration system, most of the components could be implemented by a multitude of technologies. However, where new data transfer components are required, RabbitMQ has been chosen as the technology to represent to reception of these data transfers.

 DOI: 10.3217/978-3-99161-057-1-133

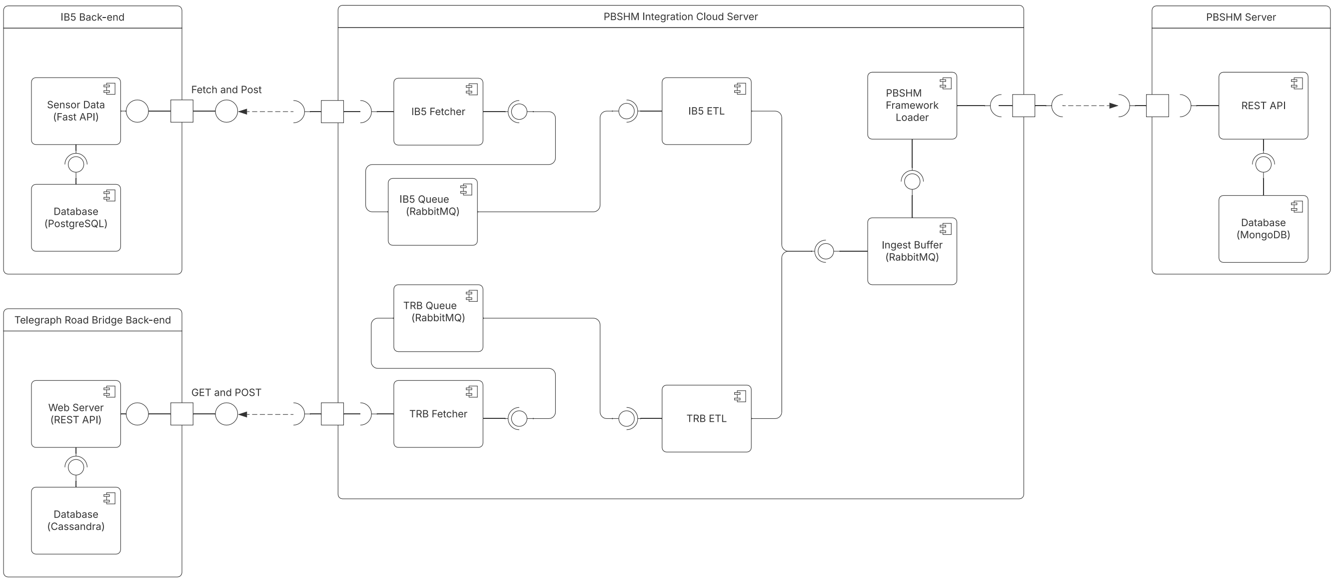

Figure 5. Universal markup language diagram showing the proposed population-based structural health monitoring data integration system for the Intersection Bridge 5 and Telegraph Road Bridge

Whilst these components could most likely be replaced with similar technologies or even a more traditional data transfer and storage method, RabbitMQ has been chosen as it both dictates the protocol in which messages are sent/read and provides a data store in the form of the message queue. Furthermore, it is open-source and well-documented, allowing ease of use without complications surrounding licensing [12], [13].

As stated, no changes have been made to the functionality provided by either the IB5 or Telegraph Road back-ends. Therefore, to bring data into the “PBSHM Integration Cloud Server”, two “Fetcher” components (each for their respective monitoring system) have been added which periodically sends requests to their respective APIs to check whether new data has been added to the databases and then fetches and loads it onto a RabbitMQ, therefore fulfilling the role of the data generator. An example application provided in [10] documents how automated data retrieval could be implemented for the Telegraph Road Bridge.

Once the sensor data from each bridge has been loaded into their respective message queues, sensor data from both bridges is then extracted from their message queues into ETL components. The purpose of these ETL components is to transform the data in the message queues from their respective data formats and schemas into a JSON format that follows the PBSHM schema. Whilst the data from both bridges is undergoing similar processes to be transformed into PBSHM schema, the underlying technologies will take significantly different methods to undertake this due to the difference in the data format and structure (schema) of the data generators. The key differences being:

- In IB5, data is assumed to be returned in XML format whereas Telegraph Road returns data in JSON format.
- It is assumed that IB5 returns data in a similar schema to that shown in Figure 3 whereas Telegraph Road produces data in its schema (examples given in [10]).

Once the data from either structure has been transformed into PBSHM Schema, this can then be loaded onto another RabbitMQ labelled the “Ingest Buffer”.

By transforming the data into the PBSHM schema data integrity and interoperability are ensured as the schema provides a unified way to represent sensor data and contains information about the origin of the data.

From this point, as the data from both bridges is in the same structure and format it can be manipulated by the same components through the rest of the pipeline. The data is then extracted by the ingest buffer and loaded onto “PBSHM Server” via a REST API with an HTTP(S) request at which point the “PBSHM Server” API will load the data onto the database.

6 CONCLUDING REMARKS

This paper presents a methodology for the design, implementation and operation of PBSHM data collection systems. By defining mandatory roles for data transfer from structure to PBSHM Network, Framework and Database scope and demonstrating their application to real-world, bridge monitoring systems, it is shown how diverse existing SHM data acquisition systems can be integrated into a PBSHM ecosystem, whilst maintaining data integrity and interoperability as provided by the standard PBSHM schema. Future work will focus on refining the optional components within the aggregation scope, particularly the config and health role which present significant challenges due to the variability in configurable attributes and operational performance. Robust reference implementations of the aggregation scope components will also be developed to provide practical guidance for SHM system operators seeking to integrate their monitoring systems with PBSHM systems, thereby accelerating the adoption of PBSHM. Furthermore, work will include the development of software modules for the PBSHM framework that allow for the use of external software tools by

DOI: 10.3217/978-3-99161-057-1-133

providing interfaces to the data collected as defined by security policies.

ACKNOWLEDGMENTS

The authors gratefully acknowledge the support of the UK Engineering and Physical Sciences Research Council (EPSRC) through the ROSEHIPS project (Grant EP/W005816/1). For the purpose of open access, the authors have applied a Creative Commons Attribution (CC BY) licence to any Author Accepted Manuscript version arising.

REFERENCES

[1] P. R. A. Fidler, F. Huseynov, M. Bravo-Haro, V. Vilde, J. M. Schooling, and C. R. Middleton, 'Augmenting an existing railway bridge monitoring system with additional sensors to create a bridge weigh-in-motion system and digital twin', Jul. 2022, doi: 10.17863/CAM.86738.

[2] S. M. O'Connor, Zhang ,Yilan, Lynch ,Jerome P., Ettouney ,Mohammed M., and P. O. and Jansson, 'Long-term performance assessment of the Telegraph Road Bridge using a permanent wireless monitoring system and automated statistical process control analytics', *Structure and Infrastructure Engineering*, vol. 13, no. 5, pp. 604–624, May 2017, doi: 10.1080/15732479.2016.1171883.

[3] M. Ozkaya, 'Do the informal & formal software modeling notations satisfy practitioners for software architecture modeling?', *Information and Software Technology*, vol. 95, pp. 15–33, Mar. 2018, doi: 10.1016/j.infsof.2017.10.008.

[4] L. A. Bull *et al.*, 'Foundations of population-based SHM, Part I: Homogeneous populations and forms', *Mechanical Systems and Signal Processing*, vol. 148, p. 107141, Feb. 2021, doi: 10.1016/j.ymssp.2020.107141.

[5] J. Gosliga, P. A. Gardner, L. A. Bull, N. Dervilis, and K. Worden, 'Foundations of Population-based SHM, Part II: Heterogeneous populations – Graphs, networks, and communities', *Mechanical Systems and Signal Processing*, vol. 148, p. 107144, Feb. 2021, doi: 10.1016/j.ymssp.2020.107144.

[6] P. Gardner, L. A. Bull, J. Gosliga, N. Dervilis, and K. Worden, 'Foundations of population-based SHM, Part III: Heterogeneous populations – Mapping and transfer', *Mechanical Systems and Signal Processing*, vol. 149, p. 107142, Feb. 2021, doi: 10.1016/j.ymssp.2020.107142.

[7] D. S. Brennan, J. Gosliga, E. J. Cross, and K. Worden, 'Foundations of population-based SHM, Part V: Network, framework and database', *Mechanical Systems and Signal Processing*, vol. 223, p. 111602, Jan. 2025, doi: 10.1016/j.ymssp.2024.111602.

[8] S. Jeong, R. Hou, J. P. Lynch, H. Sohn, and K. H. Law, 'An information modeling framework for bridge monitoring', *Advances in Engineering Software*, vol. 114, pp. 11–31, Dec. 2017, doi: 10.1016/j.advengsoft.2017.05.009.

[9] D. Gürdür Broo, M. Bravo-Haro, and J. Schooling, 'Design and implementation of a smart infrastructure digital twin', *Automation in Construction*, vol. 136, p. 104171, Apr. 2022, doi: 10.1016/j.autcon.2022.104171.

[10] S. Jeong, Hou ,Rui, Lynch ,Jerome P., Sohn ,Hoon, and K. H. and Law, 'A scalable cloud-based cyberinfrastructure platform for bridge monitoring', *Structure and Infrastructure Engineering*, vol. 15, no. 1, pp. 82–102, Jan. 2019, doi: 10.1080/15732479.2018.1500617.

[11] D. S. Brennan, T. J. Rogers, E. J. Cross, and K. Worden, 'On calculating structural similarity metrics in population-based structural health monitoring', *Data-Centric Engineering*, pp. 1–24, Aug. 2024, doi: 10.1017/dce.2024.45.

[12] M. M. P and S. Dixit, 'Distributing Messages Using Rabbitmq with Advanced Message Exchanges', *International Journal of Research Studies in Computer Science and Engineering*, vol. 6, no. 2, pp. 24–28, 2019.

[13] G. Fu, Y. Zhang, and G. Yu, 'A Fair Comparison of Message Queuing Systems', *IEEE Access*, vol. 9, pp. 421–432, 2021, doi: 10.1109/ACCESS.2020.3046503.