International Association
Bridge Earthquake Engineering

The Fourth International Bridge Seismic Workshop, 4th IBSW

Carleton University, Ottawa, Ontario, Canada, 11-14 August 2024

TOWARDS ACHIEVING LONGEVITY IN A DIGITAL TWIN FRAMEWORK FOR INSTRUMENTED BRIDGES

C. M. Perez[(1)], K. M. Mosalam[(2)]

(1) Graduate Student Researcher, University of California, Berkeley, claudio_perez@berkeley.edu
(2) Taisei Professor of Civil Engineering & Director of the Pacific Earthquake Engineering Research (PEER) Center, Department of Civil and Environmental Engineering, University of California, Berkeley, mosalam@berkeley.edu

Abstract

As demand for digital twins continues to grow, keeping pace with the swift progress of artificial intelligence in the field presents a challenge. This paper addresses this issue by formalizing five key abstract concepts related to structural health and investigates the representation of these abstractions in production-level software frameworks. The approach aims to unite data-driven technologies with classical methods of structural mechanics to achieve seamless integration between modern parameter-based techniques and conventional finite element models. The framework is demonstrated through a full-scale implementation centered around 22 bridges in the California highway network where a myriad of both data-driven and conventional structural health monitoring procedures is employed, supporting both the scalability and adaptability in the proposed framework.

Keywords: *digital twin; physics informed; seismic hazards; structural health monitoring*

1 Introduction

The digital twin concept has emerged as a significant development in the realm of structural health monitoring that holds substantial promise for transforming infrastructure maintenance, especially for bridges, which are integral to our transportation networks. However, the path to actualizing this concept in practice is fraught with challenges. The literature, while replete with promising studies, lacks a unified understanding for the responsibilities of a digital twin, leading to disparate interpretations of its role and functionality. Moreover, the swift progression of artificial intelligence in this field poses an additional challenge, as it necessitates the development of solutions that can adapt to a rapidly evolving technological landscape.

To address these issues, this study seeks to delineate the objectives of digital twins and structural health monitoring clearly, to facilitate the coherent implementation of state-of-the-art ideas [1, 2], in a framework that aims to remain readily adaptable to the continual advances in machine learning technology. We propose a framework comprising five fundamental abstractions—Assets, Predictors, Events, Evaluations, and Metrics—that can guide the development of scalable and enduring software platforms. This framework is designed to accommodate the fluidity of artificial intelligence advancements while maintaining a steadfast focus on the core objectives of structural health monitoring. The proposed framework is compatible with the myriad procedures proposed in the literature and facilitates their robust implementation as part of a cohesive commercial-grade software platform. By offering a software platform that supports the implementation of a broad range of existing methods, we aim to foster a more cohesive and collaborative approach to advancing structural health monitoring. Through this study, we aim to contribute to the ongoing discourse on refining the digital twin concept and advancing the field of structural health monitoring.

The outline of the paper is as follows. In Section 2 the five fundamental abstractions are introduced and developed for general applications of digital twins. In Section 3 we present the Bridge Rapid Assessment Center for Extreme Events (BRACE^2) platform, a fully realized implementation of the framework that illustrates the viability and efficacy of these abstractions.

2 Abstract Framework

The central objectives of structural health monitoring are represented by five key abstractions (Fig. 1): Assets, Events, Evaluations, Predictors, and Metrics. These abstractions each embody an essential component of a complete structural health monitoring framework. An Asset is a unique physical object for which there is interest in its structural health. The term Metric is used to refer to those final quantities that are presented before a decision maker and serve as indicators of structural health or damage. These metrics are analogous to indicators like body temperature and heart rate that a medical specialist uses to diagnose a sick patient. They are computed, in real time or near real time, following Events, e.g., a major earthquake or a man-made accident, due to pre-defined triggering mechanisms and their associated thresholds. Health metrics are generated by Predictors, which either simulate the structural response or identify the structural properties. To conduct a meaningful structural health assessment, the assessor must have a sound understanding of how each metric is composed and interpreted.

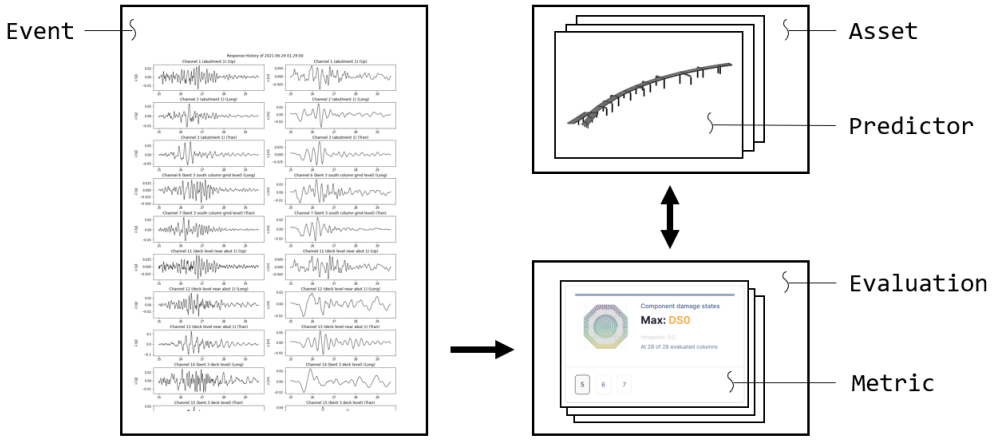
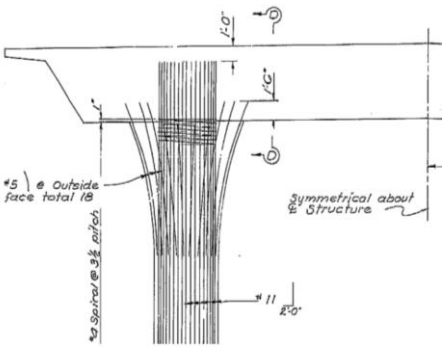


Fig. 1 - The Asset, Predictor, Event, Evaluation, and Metric represent the central abstractions of the BRACE^2 platform for bridge health monitoring.

2.1 Assets

The Asset abstraction represents the primary subject of the structural health monitoring. For example, the Painter St. overcrossing in Fig. 2(a) is an Asset identified with station number `CE89324` of the California Strong Motion Instrumentation Program (CSMIP) and structure number `04 0236` in the Federal Highway Administration's National Bridge Inventory. Naturally, an inventory of Assets will consist of a wide variety of structural designs, each with distinct vulnerabilities and strengths. It is essential that the representation of an Asset can account for these contextual details. For example, several older bridges in the Caltrans inventory, including the Painter St. overcrossing, are designed with continuous flared columns like that shown in Fig. 2(a). Structures with this design are particularly vulnerable to certain failure modes as shown in Fig. 2(b) [3–5].



(a) Obsolete continuous flare detail from the Painter St. overcrossing



(b) Column failure of the Mission Gothic Bridge following the 1994 Northridge earthquake (Photo courtesy of PEER NISEE, [6])

Fig. 2 - Example of an important feature included in the representation of an Asset.

2.2 Predictors

The Predictor abstraction represents procedures that implement a mathematical model of a physical system. This abstraction, depicted in Fig. 3, encapsulates the complexity of various modeling techniques and offers a unified interface for predicting system behavior. This generally includes any analysis procedure that an engineer can configure beforehand to be run when an Event is triggered. This covers both methods based on conventional structural analysis and methods based on data fitting. Because of the rate of active development in the field of structural health monitoring, the development of these procedures is expected to continue growing.

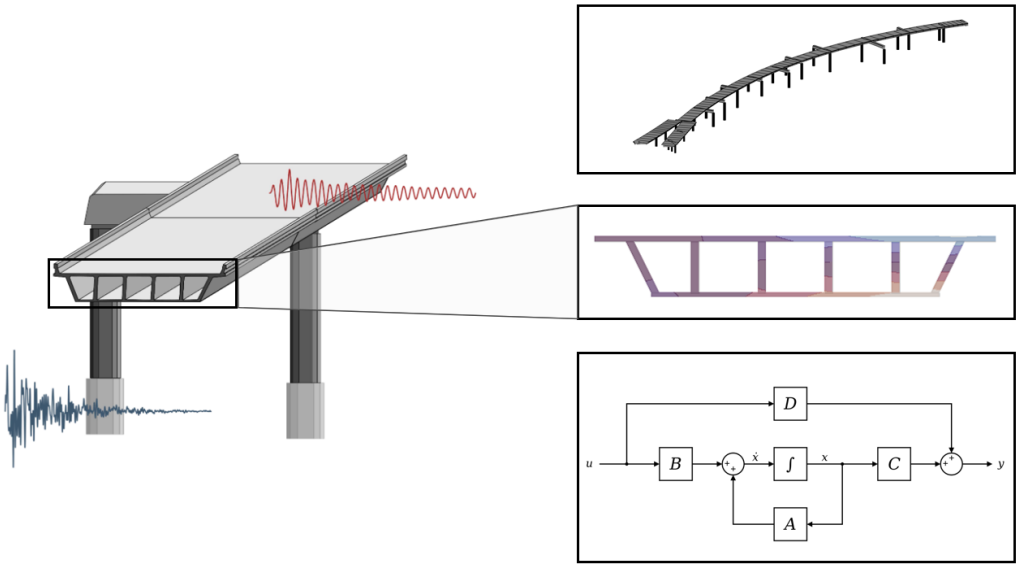


Figure 3- A depiction of an Asset (left) alongside three associated Predictors (right). The depicted predictors are (from top to bottom) a nonlinear finite element model for the dynamic equilibrium problem, a finite element model for the plane warping problem, and a state-space model for the inverse dynamic equilibrium problem.

It is useful to consider prediction methods as belonging to one of two families: Type I Predictors and Type II Predictors. In both cases, we define a Predictor through a *Physics-Driven, Data-Informed* paradigm, such that

the state of a structure as it experiences an Event is assumed to be characterized by its displacement $\mathbf{u}$ and velocity $\dot{\mathbf{u}}$, which is governed by Newton's laws of motion [7, 8]:

$$\mathbf{m}\ddot{\mathbf{u}} = \mathbf{f}^{\text{ext}} - \mathbf{f}^{\text{int}}(\mathbf{u}, \dot{\mathbf{u}}). \tag{1}$$

Here $\ddot{\mathbf{u}}$ is the acceleration associated with the state of the structure, $\mathbf{f}^{\text{int}}$ characterizes the internal resisting force produced by the structure in response to its configuration, and $\mathbf{f}^{\text{ext}}$ is the external loading due to the Event.

2.2.1 Type I: Forward Methods

In the context of Eq. (1), Type I Predictors answer the problem statement:

> If the form of $\mathbf{f}^{\text{int}}$ is *assumed* and the loading $\mathbf{f}^{\text{ext}}$ is *given*, then $\mathbf{u}$ and $\dot{\mathbf{u}}$ can be approximately *determined*.

Type I Predictors are commonly realized as approximate models for boundary value problems on infinite-dimensional (Hilbert) vector spaces by employing a *finite element* discretization to furnish $\mathbf{m}$ and $\mathbf{f}^{\text{int}}$ and an approximation to $\mathbf{f}^{\text{ext}}$ is constructed from input sensor data. Highly nonlinear geometric phenomena like buckling and collapse are efficiently represented by generalizing to the setting of infinite-dimensional manifolds [9]. Finite element models as Predictors have recently been considered in the context of digital twins as documented in [2, 10, 11]. These techniques are particularly attractive because they are readily accessible to the large workforce of civil engineers that routinely develops models of this type for problems in design and rehabilitation. However, conventional approaches to constructing such models can be extremely costly to produce at scale, as they require a qualified engineer to carefully characterize the properties of each structural component. Alternatively, Type I Predictors may be realized through artificial intelligence models that have been trained to approximate the response of a finite element model (see, e.g., [12]). Although the field is still in its infancy, this approach may be particularly important in cases where transferable models can be achieved.

It is important to note that developing a finite element model for structural health monitoring is fundamentally distinct from the more typical practice of modeling for design or rehabilitation. Many common guidelines and best practices which have been developed for these latter purposes would not generally be well suited for structural health monitoring of instrumented systems.

2.2.2 Type II: Inverse Methods

In the context of Eq. (1), Type II Predictors answer the problem statement:

> If general constraints on $\mathbf{m}$ and $\mathbf{f}^{\text{int}}$ are *assumed* and $\mathbf{f}^{\text{ext}}$ and $\ddot{\mathbf{u}}$ are *given*, then the function $\mathbf{f}^{\text{int}}$ can be approximately *determined*.

Type II Predictors are particularly amendable to data-driven methods. Data-driven methods take advantage of measured response quantities to reduce the number of modeling parameters when an analyst is faced with the challenge of defining them in Type I Predictors. This reduction can be quite significant, and with adequate data sources, can be accurately configured in just hours, as opposed to the weeks that an engineer may typically require to assemble a traditional structural analysis model of a bridge system. For example, a Type II Predictors may be realized by prescribing the domain and range of $\mathbf{f}^{\text{int}}$ and then using *system identification* methods to fit $\mathbf{f}^{\text{int}}$ to directly observed loading/excitation and acceleration response data [13, 14]. Methods of system identification use recorded response histories to characterize the system dynamics of the structure at hand. These dynamics are expressed in terms of either an input-output relationship in the frequency domain, called a transfer function, or a state space representation in the time domain, which is a set of first-order differential equations. Generally, the parameters of the dynamic equilibrium equations are used to identify characteristics of the structure. Health monitoring then entails tracking the data-fitted identity of a structure over time.

2.3 Metrics

Metrics represent the various outputs that Predictors might produce. They provide a common language for interpreting and comparing the results of different Predictors. For instance, both a system identification method and a finite element model can be used to fit modal properties to a given input and output signal. Despite these being fundamentally different Predictors, the Metric abstraction allows the modal properties they produce to be presented in a common manner or compared. Metrics are generally used for two important purposes: (1) they may be passed down to decision-makers to inform judgment calls or (2) they may be fed back to Predictors for the purpose of model updating or training [15, 16].

2.3.1 Metrics for Evaluation

When metrics are passed to decision-makers, they translate the complex mathematical models into understandable and actionable information. This could be in the form of risk assessments, performance indicators, or any other form of measurement that is relevant to the physical system being modeled. By providing a consistent interface for these outputs, the Metric abstraction ensures that the insights generated by different Predictors are comparable and comprehensible, regardless of the underlying prediction method [17–20].

2.3.2 Metrics for Updating

The second role of metrics is their use in model updating or training. In this scenario, the metrics serve as a feedback mechanism. The outputs of the Predictors are evaluated against some desired outcome or benchmark, and this evaluation is fed back into the Predictor. This feedback can be used to adjust the parameters of the Predictor, effectively 'training' it to improve its future outputs [2, 10, 21]. The Metric abstraction is instrumental in this context. It provides a common interface for these evaluations, facilitating the interchangeability of different training algorithms. For instance, consider two distinct Predictors: one employing a least squares method for system identification, and another utilizing a genetic algorithm for optimizing a finite element model [22]. Despite the fundamental differences in their methodologies, both Predictors can leverage the Metric abstraction to establish a consistent feedback loop.

The consistency provided by the Metric abstraction is crucial for several reasons. First, it simplifies the implementation of the training process. Regardless of the complexity or nature of the training algorithm, the feedback loop remains the same. This uniformity reduces the potential for errors and makes the code easier to understand and maintain. Second, it encourages experimentation with different training algorithms. By abstracting away the details of the feedback loop, the Metric abstraction allows researchers to focus on the training algorithms themselves. They can easily exchange different algorithms, compare their performance, and choose the one that best suits their needs. This flexibility is key to driving innovation and improving the accuracy and efficiency of the Predictor.

2.4 Events and Evaluations

The Event and Evaluation abstractions provide a general framework for capturing and processing data from the physical system, enabling the digital twin to reflect the state of the physical system accurately and in real time. In the context of bridge health monitoring, this generally corresponds to any situation that could place the health of a structure in jeopardy and for which an evaluation is desired, but more generally, may also be used to represent any relaying of data from the physical structure to the digital twin. The processing of an Event generally culminates in the creation of an Evaluation, which is expected to be of immediate interest to the user or the decision-maker.

2.4.1 Event Abstraction

The Event abstraction represents any occurrence that results in data being relayed from the physical system to the digital twin. An Event could be a routine data collection, a triggered sensor reading due to a significant change in the physical system, or any other situation that necessitates the transfer of data for processing and analysis. This

abstraction is designed to be broad and flexible, accommodating a wide range of scenarios and data types, and allowing the digital twin to respond to changes in the physical system promptly and accurately.

2.4.2 Evaluation Abstraction

The Evaluation abstraction represents the process of analyzing the data received from an Event. This involves invoking the appropriate Predictors, processing the data, and generating metrics that provide insights into the state and performance of the physical system. The Evaluation abstraction is designed to be adaptable and scalable, capable of handling a wide range of analysis methods and data volumes. It ensures that the digital twin can provide meaningful and actionable insights, regardless of the complexity or scale of the data.

3 Implementation: The BRACE^2 Platform

The BRACE^2 platform is a dynamic web application commissioned by the California Department of Transportation (Caltrans) to provide a link between:

1. Decision-makers who bear the responsibility for public safety and need to make informed decisions about infrastructure.
2. Practicing engineers who understand structural mechanics and are equipped to develop prediction models that can accurately forecast the behavior of structures under various conditions.
3. Researchers or specialists in the field of structural health monitoring who are actively studying and developing new metrics for health assessment.

The platform was initially deployed in a 6-month pilot phase, where digital twins were created for 22 instrumented bridges in the Caltrans inventory, which are listed in Table 1. In this period, real-time monitoring was performed on the *Hayward Hwy 580-238 Interchange* (CSMIP station `CE58658`), while the remaining 21 bridges were studied using preexisting records. Following the success of this trial deployment, development of the platform was resumed with future work expected to expand its coverage of the California highway network.

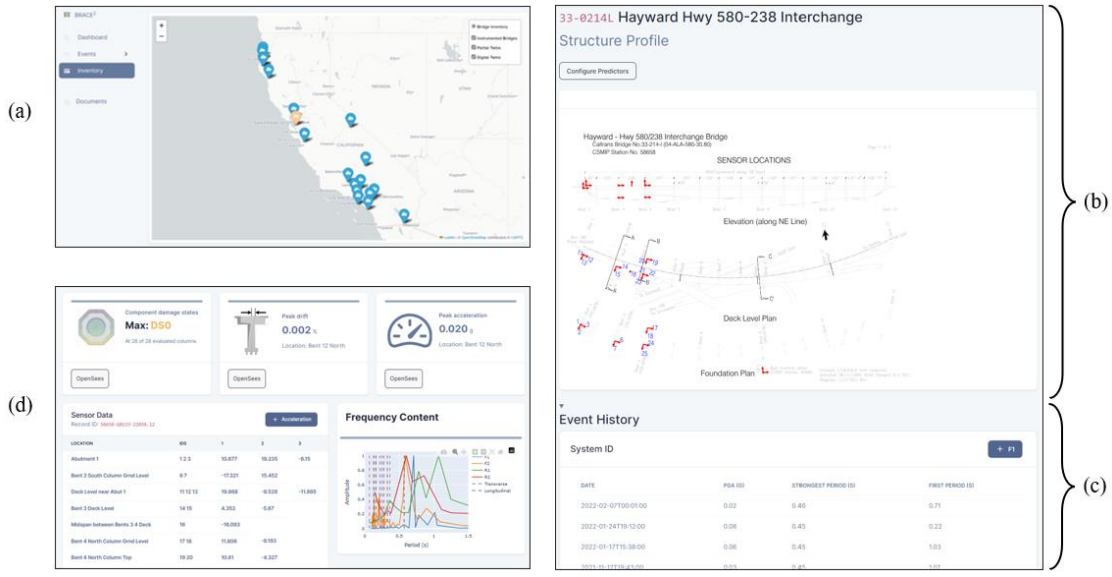


Fig. 4 - Pages from the BRACE^2 platform: (a) Dashboard, (b & c) Asset page, and (d) Evaluation page.

The implementation of the BRACE^2 platform is organized into four distinct modules. Each module implements the fundamental abstractions on the platform and has exclusive control over various tables in the database. The remainder of this section will detail this architecture, with a particular focus on the encapsulation provided by these modules and how they interact with each other. To this end we consider three key web pages generated by the platform which exemplify the merits of the abstract framework developed in Section 2 (Fig. 4):

1. The *dashboard* renders a high-level snapshot of the bridges that are managed by the platform and available for the decision-maker (Fig. 4(a)).
2. An *asset page* provides a detailed profile of the physical asset with its unique vulnerabilities and metadata (Fig. 4(b)), as well as a record of all existing evaluations (Fig. 4(c)).
3. An *evaluation page* is a report on the health of an asset, generated in response to an *event* (Fig. 4(d)).

Table 1 - Bridges monitored on the BRACE^2 platform.

CESMD*	Year	Name	Events	Sensors	Predictors
CE13705	1994	Corona - I15/Hwy91 Interchange Bridge	3	9	6
CE13795	1999	Capistrano Beach - I5/Via Calif. Bridge	5	12	3
CE14406	1981	Los Angeles - Vincent Thomas Bridge	7	26	4
CE24694	1995	Sylmar - I5/14 Interchange Bridge	2	42	2
CE24704	1994	Los Angeles - I10/La Cienega Bridge	5	15	3
CE24706	1994	Palmdale - Hwy 14/Barrel Springs Bridge	6	12	3
CE24775	1998	Grapevine - I5/Lebec Rd Bridge	7	16	3
CE33742	1996	Ridgecrest - Hwy 395/Brown Road Bridge	5	9	3
CE47315	1977	San Juan Bautista - Hwy 101/156 Overpass	10	15	3
CE68184	2002	Vallejo - Carquinez/I80 East Bridge	3	58	2
CE68185	2004	Vallejo - Carquinez/I80 West Bridge	5	103	5
CE79421	2009	Leggett - Hwy 101/Confusion Hill Bridge	5	24	6
CE89686	1996	Eureka - Samoa Channel Bridge	18	27	4
CE89708	1995	Arcata - Hwy 101/Murray Road Bridge	7	12	6
CE89735	1996	Eureka - Middle Channel Bridge	14	9	3
CE89736	1996	Eureka - Eureka Channel Bridge	13	12	4
CE89973	2001	Rio Dell - Hwy 101/Eel River Bridge	44	18	3
CE23631	2001	San Bernardino - I10/215 Interchange	9	18	4
CE54730	1995	Lake Crowley - Hwy 395 Bridge	5	9	6
CE89324	1977	Rio Dell - Hwy 101/Painter St. Overcrossing	31	20	6
CE01336	1978	Hwy8/Meloland Overpass	13	32	7
CE58658	1993	Hayward Hwy 580-238 Interchange	62	20	7
Total			**279**	**518**	**93**

*Station numbers according to the Center for Engineering Strong Motion Data (CESMD)

3.1 The `Inventory` Module

The `Inventory` module is central to the identity of the digital twin. It implements the representation of an Asset and is primarily concerned with the creation and management of entries in the `Asset` database table. To ensure that an `Asset` entry can collect all `Predictor` entries associated with it, the `Inventory` module imports the `prediction` module (see Section 3.2). To incorporate the structural details mentioned in Section 2.1 into the decision-making process, the platform represents assets with a personalized structure profile (Figs. 4(b) and (c)) which is integrated with the Federal Highway Administration's National Bridge Inventory.

3.2 The `Prediction` Module

The `prediction` module implements the Predictor and Metric abstractions. Its responsibilities include management of the `Predictor` database table and the dispatching of predictor procedures.

3.2.1 Database Management

The `Predictor` table serves as a repository for the configurations of the predictors that are invoked to populate the `Evaluation` entries for a specific `Asset`. This feature aligns with the objective of creating a dynamic software model that reflects the physical state of a structure, as it allows for the seamless integration of various predictors into the platform.

3.2.2 Procedure Dispatch

The dispatching of predictors is managed by the `Predictors` submodule, which implements the abstract `PredictorDaemon` class and its subclasses. These classes abstract away the implementation details of various predictors, providing a unified interface for their dispatch. The `PredictorDaemon` class also plays a crucial role in resource management. By providing an abstract interface for dispatching predictors, it allows the `EvaluationDaemon` class of the `Evaluation` module (see Section 3.3) to control the computational resources consumed by the predictors. This feature ensures that the platform can handle many simultaneous evaluations without overloading the system resources.

Predictors are invoked as subprocesses, and communication is performed over standard input and output streams (see, e.g., [23] for more information on these concepts). This design ensures efficient resource utilization and allows for the concurrent execution of multiple predictors. To facilitate this communication, two protocols have been developed. The protocols rely on JSON-encoded messaging in a manner that is compatible with works like [24, 25]. The flexibility afforded by this design has made it possible to integrate a wide variety of scientific computing tools into the platform to be used as Predictors. These include the `mdof` Python library for system identification [26], and the new experimental OpenSeesRT library [27], a redesigned and optimized frontend to the core OpenSees framework which has been redesigned to facilitate model updating and machine learning tasks.

3.3 The `Evaluation` Module

The `Evaluation` module is responsible for: (1) managing the `Event` and `Evaluation` database tables, (2) providing REST APIs for inter-network communication, and (3) managing the computational resources consumed by the predictors in the creation of an Evaluation.

3.3.1 Event Management

In the context of the BRACE^2 application, the Events under consideration are earthquakes. The Assets on the platform are instrumented with accelerometers that record translational motion at various locations on or near the structure. To facilitate reliable and secure communication of Event data, a secured REST API has been implemented in the platform. This feature is a critical component to enabling the Event record (Fig. 4(c)) and Evaluation (Fig. 4(d)) pages to be safely and reliably updated in real time.

3.3.2 Evaluation Management

The `Evaluation` table is a central repository that stores the results of all evaluations conducted on the platform. This table is managed by the `Evaluation` module, ensuring that the results are stored securely and can be accessed efficiently. The `Daemon` submodule is responsible for scheduling Predictors to populate `Evaluation` entries. This scheduling process is tailored to the specific properties of each `Asset` and `Event`, ensuring that the evaluations are conducted in a manner that optimizes the use of computational resources. The `EvaluationDaemon` instance is created within the `Events` module in response to the creation of a new `Event` entry. This process ensures that every Event triggers an Evaluation, facilitating real-time monitoring of the structural health.

4 Conclusion

In this study, we have presented a comprehensive framework for the realization of digital twins in the realm of structural health monitoring. Our approach is grounded in five fundamental abstractions—Assets, Predictors, Events, Evaluations, and Metrics—that provide a robust and flexible foundation for the development of scalable and enduring software platforms. We have demonstrated the practical implementation of these abstractions in the BRACE^2 platform, a full-scale software application for structural health monitoring. The BRACE^2 platform exemplifies how our approach can successfully actualize the concept of a digital twin for structural health monitoring of bridges, and potentially, a wide range of other structures. While our study focuses on the application of digital twins for bridge health monitoring, the abstractions and framework we propose are general and adaptable. They can be applied to a wide range of scenarios and data types, accommodating the diverse range of methods used in structural health monitoring.

5 References

[1] Kapteyn MG, Pretorius JVR, Willcox KE (2021): A Probabilistic Graphical Model Foundation for Enabling Predictive Digital Twins at Scale.

[2] Torzoni M, Tezzele M, Mariani S, Manzoni A, Willcox KE (2024): A digital twin framework for civil engineering structures. *Computer Methods in Applied Mechanics and Engineering*, **418**, 116584. DOI: 10.1016/j.cma.2023.116584.

[3] Sanchez AV, Seible F, Priestley MJN (1997): *Seismic performance of flared bridge columns*.

[4] Soleimani F, Mangalathu S, DesRoches R (2017): Seismic Resilience of Concrete Bridges with Flared Columns. *Procedia Engineering*, **199**, 3065–3070. DOI: 10.1016/j.proeng.2017.09.417.

[5] Caltrans (2009): *Column analysis considerations*. 2009.

[6] Aschheim MA (1994): Misssion gothic bridge.

[7] Arnold VI (1978): *Mathematical Methods of Classical Mechanics*. vol. 60. New York, NY: Springer.

[8] Courant R, Hilbert D (1989): *Methods of Mathematical Physics*. 1st ed. Wiley; 1989. DOI: 10.1002/9783527617210.

[9] Perez CM, Filippou FC (2024): On nonlinear geometric transformations of finite elements. *International Journal for Numerical Methods in Engineering*, e7506. DOI: 10.1002/nme.7506.

[10] Ghahari F, Malekghaini N, Ebrahimian H, Taciroglu E (2022): Bridge Digital Twinning Using an Output-Only Bayesian Model Updating Method and Recorded Seismic Measurements. *Sensors*, **22**(3), 1278. DOI: 10.3390/s22031278.

[11] Perez CM, Chern C, Mosalam KM (2024): *BRACE2: Bridge rapid assessment center for extreme events*, Phase I Final Report No. CA24-3703.

[12] Pang IKT, Gao Y, Mosalam KM (2024): Deep Reinforcement Learning for Structural Model Updating Using Transfer Learning Mechanism. *Computing in Civil Engineering 2023*, Corvallis, Oregon: American Society of Civil Engineers; 2024. DOI: 10.1061/9780784485231.044.

[13] Arici Y, Mosalam KM (2003): System identification of instrumented bridge systems. *Earthquake Engineering & Structural Dynamics*, **32**(7), 999–1020. DOI: 10.1002/eqe.259.

[14] Chern C, Mosalam KM (2024): Cross-Sectional Study of Physics-Informed Bridge Health Identification. *4th International Bridge Seismic Workshop*, Ottawa, Canada: 2024.

[15] Katam R, Pasupuleti VDK, Kalapatapu P (2023): A review on structural health monitoring: Past to present. *Innovative Infrastructure Solutions*, **8**(9), 248. DOI: 10.1007/s41062-023-01217-3.

[16] Avci O, Abdeljaber O, Kiranyaz S, Hussein M, Gabbouj M, Inman DJ (2021): A review of vibration-based damage detection in civil structures: From traditional methods to Machine Learning and Deep Learning applications. *Mechanical Systems and Signal Processing*, **147**, 107077. DOI: 10.1016/j.ymssp.2020.107077.

[17] Doebling SW, Farrar CR, Prime MB (1998): A Summary Review of Vibration-Based Damage Identification Methods. *The Shock and Vibration Digest*, **30**(2), 91–105. DOI: 10.1177/058310249803000201.

[18] Salawu OS (1997): Detection of structural damage through changes in frequency: A review. *Engineering Structures*, **19**(9), 718–723. DOI: 10.1016/S0141-0296(96)00149-6.

[19] Di Lorenzo D, Champaney V, Marzin JY, Farhat C, Chinesta F (2023): Physics informed and data-based augmented learning in structural health diagnosis. *Computer Methods in Applied Mechanics and Engineering*, **414**, 116186. DOI: 10.1016/j.cma.2023.116186.

[20] Muin S, Chern C, Mosalam KM (2024): Human–Machine Collaboration Framework for Bridge Health Monitoring. *Journal of Bridge Engineering*, **29**(7), 04024041. DOI: 10.1061/JBENF2.BEENG-6587.

[21] Ereiz S, Duvnjak I, Fernando Jiménez-Alonso J (2022): Review of finite element model updating methods for structural applications. *Structures*, **41**, 684–723. DOI: 10.1016/j.istruc.2022.05.041.

[22] Gomez-Cabrera A, Escamilla-Ambrosio PJ (2022): Review of Machine-Learning Techniques Applied to Structural Health Monitoring Systems for Building and Bridge Structures. *Applied Sciences*, **12**(21), 10754. DOI: 10.3390/app122110754.

[23] Silberschatz A, Galvin PB, Gagne G (2018): *Operating system concepts*. 10th edition. Hoboken, NJ: Wiley.

[24] Deierlein GG, McKenna F, Zsarnóczay A, Kijewski-Correa T, Kareem A, Elhaddad W, *et al.* (2020): A Cloud-Enabled Application Framework for Simulating Regional-Scale Impacts of Natural Hazards on the Built Environment. *Frontiers in Built Environment*, **6**, 558706. DOI: 10.3389/fbuil.2020.558706.

[25] Deierlein GG, Zsarnóczay A, editors (2021): *State of the Art in Computational Simulation for Natural Hazards Engineering*. Zenodo; 2021. DOI: 10.5281/ZENODO.4558106.

[26] Chern C, Perez CM (2024): Mdof. DOI: 10.5281/ZENODO.11660200.

[27] Perez CM, McKenna F *et al.* (2024): OpenSeesRT. DOI: 10.5281/ZENODO.10456866.