2025 European Conference on Computing in Construction

CIB W78 Conference on IT in Construction

Porto, Portugal
July 14-17, 2025

AN EVENT-DRIVEN AGENT-BASED APPROACH TO SUPPORT OPERATION AND MAINTENANCE OF COMPLEX BUILDINGS

Alessandro Carbonari, Massimo Vaccarini, Alessandra Corneli, and Berardo Naticchia
Polytechnic University of Marche, Department of Civil and Building Engineering and Architecture (DICEA), 60131 Ancona, Italy

Abstract

Current legislation about the management of public work contracts through electronic information modelling advocates structured information that enables advanced and efficient approaches for operation and maintenance of buildings. This paper suggests and showcases an agent-based system architecture, that embeds Bayesian Networks as inference engines and event-driven query agents to pinpoint failures, allowing scalability and adaptability across building types. This is another step towards the implementation of cyber-physical systems that can deal with complex domain. Finally, the proof of concept was executed in the case of event-driven maintenance applied to a middle-sized hospital.

Introduction

The implementation of building information electronic modelling required by the EU Directive 2014/24 (Directive 2014/24/EU) and incorporated in the legislation of several EU member states, is transforming the way public tenders must be managed. This is even the case of Italian legislation, which requires the use of electronic information modelling to manage contracts of public works and design contests. The main consequence is that the appointing party must define requirements within the exchange information requirement (EIR) document, called "C.I." by the legislation in force (Italian Legislation Act no. 36/2023), and the appointed party must develop and deliver a set of information containers, hopefully in the form of building information models. One of these containers is the digital maintenance plan, which is included in the detailed design and paves the way to efficient operation and maintenance protocols of an asset. Also, this framework aligns with the approach outlined by the standard (BS EN ISO 19650-1:2018), including the Project Information Model (PIM), that is the model to be built, and the Asset information Model (AIM), that models the actual asset and updates the PIM with information collected during the execution phase of the asset.

In this paper, we emphasize that the availability of structured information models as outcomes of the design and execution phases facilitates the integration of intelligent systems in an asset at the operation phase. In addition, this paper claims that management information systems can be in charge of identifying and diagnosing faults and it suggests a framework that can help facility managers and members of the maintenance staff promptly jump into action in case any symptoms have been detected by an advanced building management system.

Indeed, energy management systems usually reside on top of Building Management Systems (BMS), hence they must be implemented as a decentralized (e.g. agent-based) system to integrate BMS units that, still, too often are made of separate blocks (Schumann et al., 2014). Also, layered architectures (e.g. sensor, middleware, control layers) often are adopted to incorporate scalability to the system (Rajad et al., 2013) and pave the way for the implementation of data-driven control policies (Pour et a. 2024). These ones take advantage of context awareness provided by the structured information embedded in information models and monitoring layers included in the physical layer of a BMS.

The system architecture worked out in this paper relies on the concept of Cyber-physical systems (CPS) to overcome the separation and lack of interoperability between BMS and higher level units and creates an agent-based integrated system, which supports maintenance. It embeds edge inference capabilities to improve overall system's resilience (Nota et al., 2021).

A proof of concept has been showcased, investigating the application of this system in the case of a middle-sized hospital, currently undergoing major refurbishment.

Scientific and technical background

Building Operation and Maintenance

Building management systems (BMS) often contain legacy systems that are not well integrated with other systems (Dietrich et al., 2010). This scenario restricts the set of advanced control systems that can potentially be implemented, such as predictive control. As a consequence, integrated management solutions based on collaborative platforms must be developed to allow the management of blocks of buildings (Schumann et al., 2014). The combined management of several disciplines hinders the application of straight control logics that are implemented within only one block of the BMS. Some maintenance conflicts may rise with the regular operation of a building, hence they must be scheduled as a consequence of the way a building is operated. This raises challenges related to the definition of an integrated architecture, inspired by the Digital Twin approach (Nota et al., 2021). Hence, cyber-physical systems (CPS) can support the management of complex assets, which contributes to the overall system's resilience improvement (Nota et al., 2021). CPS is the means to realize management information systems that combines software, hardware, distributed communication and data processing to support people in their effort to manage processes and facilities (Martirano L. et al, 2019), in compliance with the standard (ISO 50001: 2018). The long-term vision considers introducing interoperable and portable frameworks for energy and information systems seamlessly adaptable on a case-by-case basis (Chiosa et al., 2024).

The path towards Construction 4.0

The wider objective of the research presented in this paper is to help realize the transition of building management systems towards the paradigm labelled as Construction 4.0, making operation value chains more effective by driving digitalization towards CPS. This entails the challenge of improving HVAC diagnostic systems to broaden their market adoption, improving their scalability and transferability (Chen et al., 2023). Indeed, it is not an easy task because socio-technical systems organized around a building asset are becoming larger and more complex. Hence, management procedures ask for a considerate integration of IT and connectivity, decentralized control and mass customization, full realization of CPS for self-organization and self-control (Pasetti Monizza et al., 2018). Thanks to the adoption of BIM, asset's information has been structured. But management systems must be able to manage scenarios that cannot be predicted, as their behaviour emerges as a result of unpredictable interactions among context-dependent and secondary effects of reactions to unexpected occurrences. Recalling that a CPS is the combination of digital and physical parts of a system, the technical design of a digital twin relies on the structure inherent in the digital part of the CPS (Klinc and Turk, 2019).

Research questions

This paper tries to answer three research questions (RQs): (i) can an agent-based layered architecture integrate a BMS with an asset information model and inference units to support the work of an asset's maintenance staff? (ii) What technical stack can realize such an agent-based architecture deployable on several types of buildings, irrespective of their specificity? (iii) How can inference engines be integrated in data-driven management information systems and be developed from expert knowledge?

The next "Materials and methods" section was conceived to answer the RQs listed above. The "Proof of concept" section will showcase how the system could be executed in the management of a middle-sized hospital subject to major refurbishment and in compliance with the latest legislation on public contracts and design contests. Conclusions and acknowledgments end the paper.

Materials and methods

The system architecture

A four-tier system architecture (Figure 1) to realize an advanced BMS was designed, including a communication backbone for data exchange through a bus. The bus, that is based on an MQTT standard over a TCP/IP network, exploits the pub-sub mechanism and the corresponding loose coupling to allow any agents to communicate with all others. This assures reliability and scalability. This choice enables cooperation among agents by requiring the adoption of a message broker for managing the bus. Further details on the bus are provided in "Coordination of agents and bus" sub-section. The BMS tier is connected to the physical world and it is in charge of data collection in one of the building automation protocols (e.g. Modbus, BACnet, KNX, Haystack, MQTT) and publication on the bus. Even if many traditional and IoT based BMSs (like Thingsboard) are able to implement gateways for a wide variety of protocols, in case a BMS does not directly support the adopted bus standard, an external gateway (e.g. implemented in NodeRed) is required to allow technical interoperability. The semantic interoperability among devices is achieved, on the edge at the gateway level, by structuring the information provided by each sensor according to the topology given by the IFC model, as illustrated in Figure 8a. The platform tier provides management functions of devices included in the physical layer and it offers services such as data storage, inference, queries and analytics. The inference unit included within this tier is in charge of performing causal inferences from data. It is made of two parts. The lower level is in charge of regular inference on the continuous data stream; the higher level is recalled just in case a failure is detected and the quality of the inference provided by the lower level must be improved.

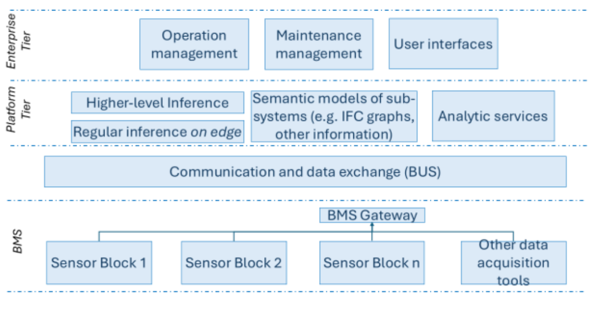


Figure 1: System architecture

The storage of semantic models (Mallak et al., 2018) will be presented in "The semantic ifc graph" section. The enterprise tier implements domain specific applications, such as operation and maintenance management. An agent-based interaction procedure among the units involved in this system architecture was conceived to enable self-organization capabilities and adaptation.

The inference tier

The inference tier includes a few inference engines, and it is in charge of filtering data pushed by the bus and of checking whether any fault calls for an action. Every agent that belongs to this layer was modelled as a BN. Indeed, BNs provide an explicit representation of causal relationships that clarify the causes of failures and can combine several sets of information into a unique inference process (Mallak, et al. 2018). They can deal with several types of knowledge, e.g. empirical and statistics (Nan Yang et al., 2020). In addition, they can implement a data-driven approach and can carry on continuous learning (Cai et al, 2018). The Bayesian Network developed in this research work, is a class of an object-oriented Bayesian Network (OOBN), because it is suitable for abstraction, it can include instance nodes as interfaces with external nodes, and it may be replicated to make up a hierarchical structure of BN fragments. Hence, it enables modularity and generalization (Korb and Nicholson, 2023). Expert knowledge was elicited through interviews (Vose, 2003), which was deemed accurate enough for the representative preliminary application object of this first research step. The interviewees were members of the maintenance staff of hospitals, that provided a narrative representation of their expert knowledge about typical faults concerning: (i) the air supply and (ii) the hydronic heating systems, that could negatively affect indoor air quality and thermal comfort. The localization of what set of components or what subsystem may be involved in a failure is in charge of the agent that queries the semantic model layer. Basically, this double layer includes both BNs that perform regular inference on edge any time a new record is provided (i.e. the "regular inference" in Figure 1), and of an higher level that performs centralized inference on-demand, in case the lower level is not able to come up with a high quality result, which will be the subject of further research.

An excerpt of the narrative regarding the air supply system that supported the development of part of the BN in Figure 2 is as follows: "*Every room is served by a loop that starts from an AHU installed on the roof. [...] In winter, outdoor air is heated the first time through a pre-heating battery. [...] The most frequent fault is that one of the heating batteries of the AHU does not work. As a result, cold air is supplied, room temperatures get colder and users complain. Another fault could be that either the supply fan or the extraction fan does not work, or both. Reasons could be that the fan belt is broken (which would impede the fan to work), or that the bearings of a fan are aging and the fan gets unbalanced, [...]* ". This textual input was translated into few nodes of the Bayesian Network depicted in Figure 2. The BN's layer no. 1 contains faults; layer no. 2 contains sensors; layer no. 3 contains symptoms (Cai et al., 2018). Sensors will be installed in the building during refurbishment works (see subsection "The case study"). The BN in Figure 2 relates faults with either the air supply system's or the heating system's levels and season. It is activated by records pushed by BMS sensors. The lowest level infer the level of indoor air quality and thermal comfort. The causal model was implemented through the Hugin™ engine. When the GUI is switched into "run mode", the bottom right corner marked in Figure 2 displays the result of data conflict analysis worked out by the engine. One of the most likely causes of a conflict is "missing evidence", for this reason this indicator suggests whether additional evidence must be collected (HuginExpert, 2020). This may be faced through the instantiation of BNs in the higher level or through additional evidence, as detailed in the "Coordination of agents and bus" subsection.

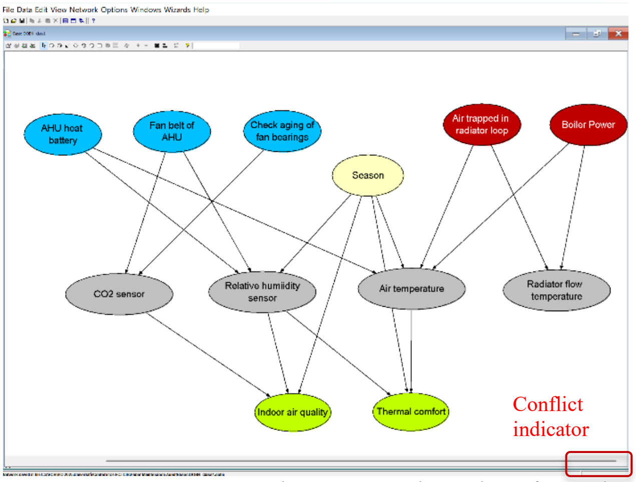


Figure 2: Bayesian Network concerning thermal comfort and indoor air quality

The semantic ifc graph

The storage of an asset information is made of a federation of IFC models for structured information and documents for unstructured information. They are stored in a database called ArangoDB, which is made of a set of collections, i.e. JSON documents. Every document in a collection can have an arbitrary number of attribute keys and values. The query language is called Arango Query Language (AQL) (ArangoDB, 2022). The stored documents are the result of a mapping process of any ifc express model into a graph, that was done by means of an in-built C++ application developed by the staff of the Dicea Department in charge of a nation-wide project (PRIN, 2017) and based on the IFC engine libraries at "https://rdf.bg/product-list/ifc-engine/", which are implemented in the platform. The application includes some options to exclude certain IFC classes, e.g. all low-level geometric representation classes, and creates two collections, one of them including entities (collection of "documents" type) and the other one representing relationships between entities (collection of "edge" type). Two documents of an IFC model can be linked by an edge document. Edge documents have additional attributes depending on the types of relationship they include, besides the required "_from" and "_to" attributes. Edges must link entities (i.e. vertices) stored in document collections. An “ifc to gltf conversion” service generates the geometry and textures from ifc files. Model versioning is managed through metadata. The geometric information associated with an IFC file is a model linked to it as a GLTF file. As a result, projects are stored in the graphDB of the collaborative platform called WeBim, along with models and views. Queries allow the cloud platform either to access data stored in the database system or to traverse a graph. The former approach includes all CRUD operations and can take advantage of “FOR” loops to search for specific combinations of parameters within any set of collections. The second type of queries starts from a vertex and follows the edges corresponding to an indicated relationship towards other vertices.

Coordination of agents and bus

The bus communication backbone supports interactions and communication among agents. As soon as an entity produces information, it delivers a message to the bus (trough MQTT protocol) along with a specific UTF-8 string made of one or more levels separated by a forwards slash, i.e. the topic. An MQTT broker is in charge of recording the topic subscriptions from all entities and to push them the corresponding messages. If the target is an agent, this event can trigger other actions.

The runtime environment used to develop and execute the agents for this paper is Node-RED. It is an asynchronous, event-driven IoT programming environment, designed for intensive real-time applications, written as a node.js application in javascript programming language. The implemented procedure involves two phases: BN initialization and BN inference updating (Figure 3). The first phase assumes that the BMS installed in a building brings information about the project, the building and the hierarchical structure of sensors in every room. Then, an agent called *BN activator* can read data about the project and building, query the graphDB to get the list of *ifcspace* entities, and associate one *space agent* to every room. Every *space agent* subscribes to the topics where data are pushed by the BMS (e.g. project/building/room*/CO_2/#) and instantiate the BN of the regular inference layer associated with every room.

The second phase is managed by the *space agent*, to (i) get sensor records pushed by the broker, (ii) update the evidence and read the inference outcome of the BN, (iii) check the conflict indicator. If this indicator is null, the inference is valid and the *space agent* can stop the execution because of no conflict, or (iv) notify the broker about an issue in the topic “system failure” in the opposite case. The broker would notify the *diagnostic agent*, which can read the payload of the latest message to read the room label and the type of failure, query the graphDB about the components of the system involved in the failure and ask the maintenance staff to react. In case of a conflict in step (iii), either additional evidence could be retrieved from the bus about another sensor node still to be instantiated in the BN on Figure 2, or another BN from the higher-inference layer can be instantiated to investigate further.

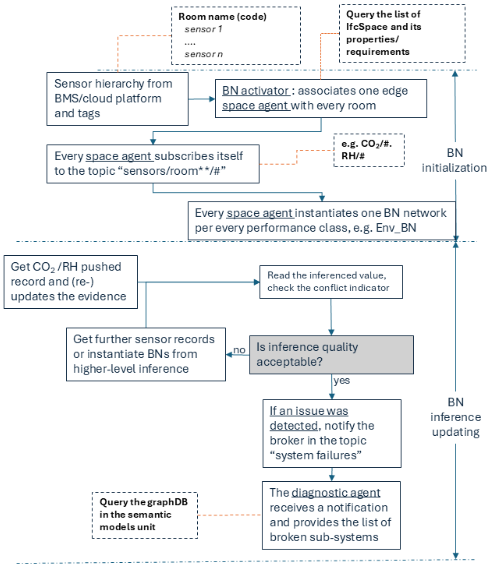


Figure 3: Conceptual scheme of the agent-based procedure implementing the self-organizing, scalable diagnostic system

Agent-based interaction procedure

The basic assumption is that every gateway of the BMS subscribes the topic concerning the sensors connected to each interested gateway, of the type “project/building/sensors/room*/#” and publishes data in the bus (Figure 4). The *BN activator* agent must instantiate one *space agent* to control every room. Information about the building is necessary to access the list of *IfcSpace* entities and get environmental requirements from the graphDB. These parameters are incorporated in the *space agent* of every room, because it must instantiate a BN, as an instance of the object-oriented BN class available in the regular inference layer. Then, the MQTT will deliver messages to the interested space agents, which can access the BN through a RESTful API and assess the outcomes (Figure 3). Every message includes a payload with the latest value recorded by any sensors (i.e. CO_2, relative humidity, air temperature), date and time, code of the space. Every *space agent* is in charge of notifying a system failure through the “system failures” topic and trigger the lowest part of the flow in Figure 3.

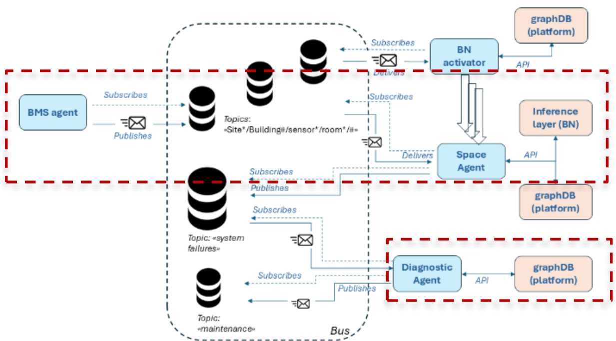


Figure 4: Interactions among agents through the bus

The overall workflow

The basic assumption to enable the logic shown in Figure 3 is that the PIM and AIM (including the as-built version of the digital maintenance plan) models are worked out during the design and execution phases, as shown in Figure 5. This includes both structured (e.g. graphDB) and unstructured information (e.g. set of guidelines, maintenance procedures and maintenance program). In the O&M phase, the system is expected to read and assess real-time data. Should there be any failures, thanks to this event-based assessment and query of information, breakdowns would be sensed and pinpointed to suggest credible responses in charge of the maintenance staff.

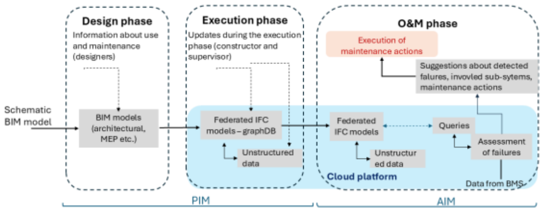


Figure 5: Overall workflow

Proof of concept

The case study

The demonstrator is a middle-sized hospital located in Pergola (a village in central Italy). It is a T-shaped, four-storey building, including two more levels underground. Every floor is 1,500 m^2 large. The hospital is undergoing major refurbishment. The design of the new project is made of several models associated to disciplines, e.g. architectural, MEP, structural, but also technical furnishing, such as hospital beds and medical devices. The model depicted in Figure 6-a is a federation of the IFC architectural model, the IFC air supply system and the IFC model of hospital beds. Picture 6-b depicts refurbishment works at the ground floor, in preparation for the installation of the air supply system.

Design of the proof of concept experiment

The digital maintenance plan concerns the technical subsystems and include both geometric and non-geometric information (Figure 7-a). Every terminal unit (e.g. an air inlet) is associated with a room, and includes properties about comfort and IAQ requirements. Sub-systems were organized hierarchically, as shown in Figure 7-b, where every cluster in the rightmost window of the editor represents a sub-system. Classification is arranged into the lists of supply and return subsystems. They are numbered progressively and according to the floor, e.g. level 1 is served by six supply and six return subsystems; those ones serving the two operating rooms are separated from the others. In every room, CO_2, humidity and temperature probes will be connected to a gateway of the BMS that controls rates of air changes and other comfort parameters. The objective of the proof of concept was to test whether the sequence of actions in charge of the several agents was triggered correctly until the assessment of a failure and the location of involved components. To simulate the role of the BMS, a dummy time series was injected into the bus, simulating a record of CO_2 above threshold, along with data collected from the remaining sensors.

(a)
(b)

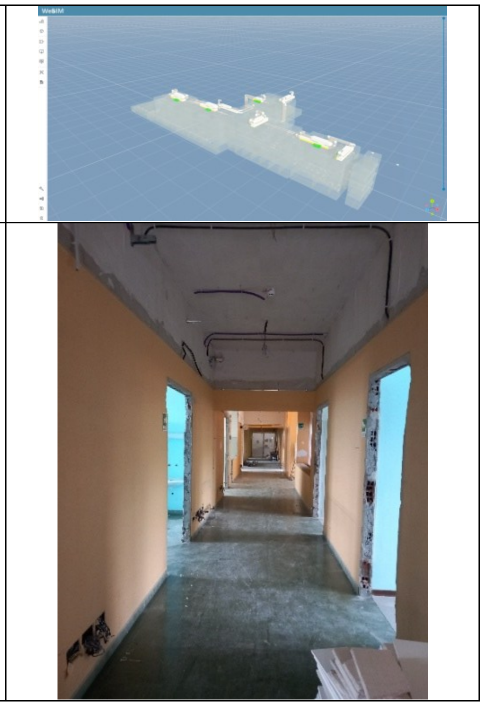


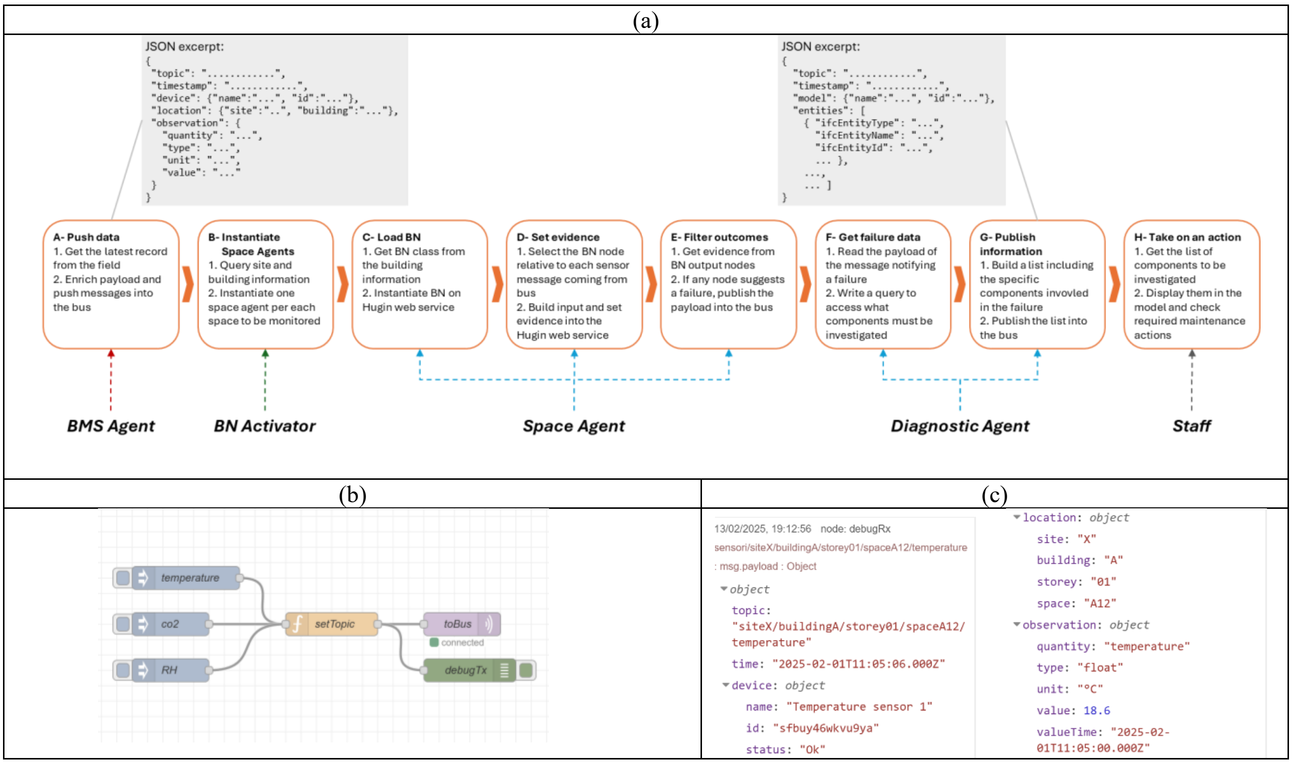
Figure 6: Federated models (a) and a picture of the jobsite (b)

(a)
(b)

Figure 7: MEP model (a) and air supply system editor (b)

Results

The proof of concept was limited to the actions of the agents framed in the red boxes in Figure 4: the *BMS agent* publishing sensor data; the *space agent* reading these records, making inferences and publishing messages to notify any possible failures; the *diagnostic agent* making queries to identify involved subsystems. Other Node-Red flows about the remaining steps have been drafted, but not executed, yet.

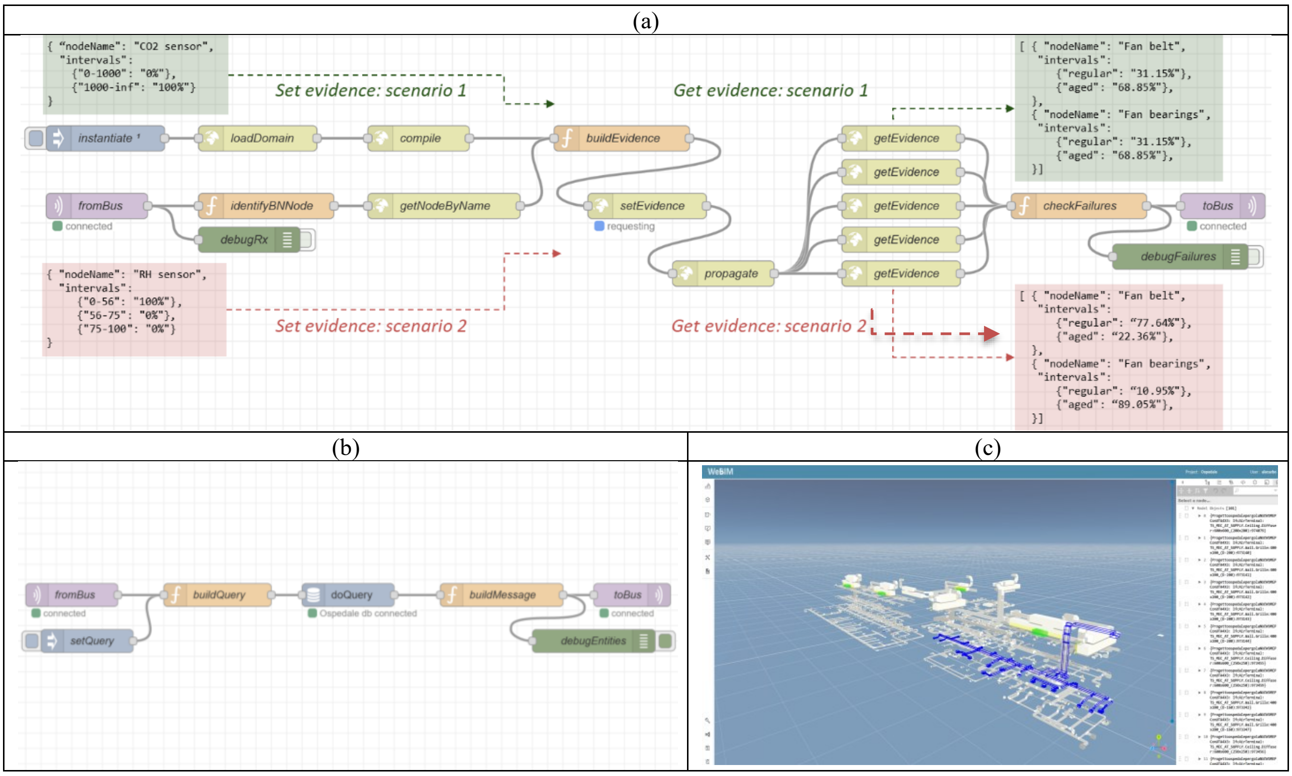



```
"topic": "............",
"timestamp": "............",
"device": {"name":"...", "id":"..."},
"location": {"site":"..", "building":"..."},
"observation": {
  "quantity": "...",
  "type": "...",
  "unit": "...",
  "value": "..."
}
```

```
"topic": "............",
"timestamp": "............",
"model": {"name":"...", "id":"..."},
"entities": [
  { "ifcEntityType": "...",
    "ifcEntityName": "...",
    "ifcEntityId": "...",
  ... },
  ...,
... ]
```

```
object
topic: "siteX/buildingA/storey01/spaceA12/temperature"
time: "2025-02-01T11:05:06.000Z"
device: object
  name: "Temperature sensor 1"
  id: "sfbuy46wkvu9ya"
  status: "Ok"
location: object
  site: "X"
  building: "A"
  storey: "01"
  space: "A12"
observation: object
  quantity: "temperature"
  type: "float"
  unit: "°C"
  value: 18.6
  valueTime: "2025-02-01T11:05:00.000Z"
```

Figure 8: Data handling and communication route for inference updating and failure identification (a); node-RED flow modelling the BMS agent (b) and content samples of the payload published into the bus (c)




```
{ "nodeName": "CO2 sensor",
  "intervals":
    {"0-1000": "0%"},
    {"1000-inf": "100%"}
}
```

```
{ "nodeName": "RH sensor",
  "intervals":
    {"0-56": "100%"},
    {"56-75": "0%"},
    {"75-100": "0%"}
}
```

```
[ { "nodeName": "Fan belt",
    "intervals":
      {"regular": "31.15%"},
      {"aged": "68.85%"},
  },
  { "nodeName": "Fan bearings",
    "intervals":
      {"regular": "31.15%"},
      {"aged": "68.85%"},
  }]
```

```
[ { "nodeName": "Fan belt",
    "intervals":
      {"regular": "77.64%"},
      {"aged": "22.36%"},
  },
  { "nodeName": "Fan bearings",
    "intervals":
      {"regular": "10.95%"},
      {"aged": "89.05%"},
  }]
```

Figure 9: NodeRed flow modelling the space agent's inference action (a) and example of Node-Red flow modelling the diagnostic agent doing a query in ArangoDB (b); list of components and their visualization as a result of the second query (c)

The *BMS agent* mentioned in Figure 4 and developed as a Node-RED flow (Figure 8-b), performs the first step of the chain of actions listed in Figure 8-a. It gets records from sensors (e.g. temperature probe), adds the topic property to the payload of the message, adds additional data about the timestamp, device properties saved in the BMS, location, type of record and recorded values. Some of these properties will support queries in later steps. The result of this step is an enriched data published to the bus (Figure 8-c). Each *space agent*, generated by the *BN activator*, subscribes to the MQTT broker for its specific topics. When a message of interest is delivered to a *space agent (*Figure 9-a.*)*, it will update evidence of the applicable BN and will trigger inferences. It exploits a Hugin web service deployed on the server of the Dicea Department and accessible by a REST API, that invokes Hugin's built-in methods. This agent performs the same steps any time a new data is delivered through the bus and received through the MQTT node "fromBus" that triggers the lower branch of the flow. The higher branch is performed once at the beginning, as soon as the BN activator instantiates the *space agent*. The BN class is identified through a specific URL that is provided by the BN activator together with the mapping between the BN nodes and the sensors. The "loadDomain" and "compile" nodes call two methods of the Hugin web service, that is a first POST to upload the domain (*domain.load*) and a second POST to compile the BN (*domain.compile*). Then, the BN can be executed. At the same time the lower branch selects what value type – hence BN's node - must be instantiated with evidence coming from sensors. The correct BN node is identified by the NodeRed's node "getNodeByName", and the evidence is built in the node "buildEvidence". This step assesses what likelihood must be set as evidence in input nodes, while the next "setEvidence" applies the *domain.select* method. Once the inference is accomplished within the next "propagation" step, inference results are retrieved by the "getEvidence" nodes. The nodes that in Figure 3 have been called "AHU heat battery", "Fan belt AHU", "Check aging of fan bearings", "Air trapped in radiator loop" and "Boiler Power", have been assessed. The "checkFailures" node filters whether any of the previous values suggest the occurrence of any failure and the last node notifies a failure in the topic "maintenance", just in case it has been detected.

The top-right box overlaid on Figure 9-a includes the JSON object including the probability values of the BN nodes "fan belt" and "fan bearings" in the first scenario. It is a consequence of a CO_2 record higher than the threshold of 1000 ppm (as reported in the top-left box overlaid in Figure 9-a). The bottom-right corner overlaid in that Figure reports the probability values of the same BN nodes, in case a second record about RH is detected, reporting an RH value between "0 and 55" in winter. This is the second scenario and it infers that "fan bearings" becomes the most likely failure. As a consequence, the *space agent* publishes a message in the "systemFailure" topic with this payload, which includes the BN node suggesting what failure and related location. This combined information will be pushed by the MQTT broker towards the *diagnostic agent*. This agent performs queries exploiting the information mentioned above along with the topology of the ifc graphDB. Thus, it suggests what the maintenance staff should investigate (Figure 9-b). The first node gets the message form the bus; the second node "buildQuery" writes an AQL query to select the location and the list of elements most likely involved in the detected failure. A connection with the graphDB (i.e. the labelled property graph reported in "The semantic ifc graph section") is established by the "doQuery" node, and the result is rearranged by the "buildMessage" node, before being published into the topic "maintenance". In this specific case, a traversal was performed, starting from the sensor entity that detected the likely failure; using the relation "IfcRelContainedInSpatialStructure" to link the "IfcSpace" inside which the sensor record was collected with the terminal unit (i.e. "IfcAirTerminal"); finding out what group of components is connected to that terminal through the "IfcRelAssignsToGroup". This relation is built by the BIM authoring SW tool (Autodesk Revit^{TM}) during the ifc export phase as a result of the hierarchy of subsystems shown in Figure 7-b. All the detected components have been included in a list and recalled and displayed (blue-coloured) inside the WeBim platform (Figure 9-c). This query marked the boundaries inside which the failure must be analysed.

Conclusions

The event-driven approach presented in this paper assumes that as-built models have been made available as a result of the design and execution phases. In addition, the latest legislation acts require that the appointing party defines requirements within an Exchange Information Requirement document, that must include all the suggestions enabling the implementation of advanced management policies in the O&M phase. The first set of requirements concerns spatial labelling of records sent from the BMS monitoring system; the second set of requirements concerns correct labelling of spaces (i.e. rooms) and the modelling of subsystems (e.g. hierarchy). As long as these inputs are combined with the availability of classes of object-oriented Bayesian Networks as inference layers and a runtime environment implementing agents, then an adaptable, self-adjustable and scalable diagnostic system can be deployed. A major limitation of this study is that the system was evaluated in just one building, lacking assessment in other buildings. As a consequence, although the approach shows potential for scalability, this still needs to be fully demonstrated in future works. Besides this topic, future research will be devoted to arrange a multi-level inference layer, where more than one BN can be combined on purpose to refine inferences, and even to enrich information access with dynamic queries associated with the type of failure.

Acknowledgments

The authors wish to thank the staff of the public institutions "Regione Marche" and "AST Pesaro Urbino" (Italy), for advising and providing the demonstrator.

References

ArangoDB Team. (2002). Graph Done Right. White Paper. ArangoDB, Inc. Available at: https://arangodb.com/resources/white-papers/graph-done-right/ . Last access: December 2024.

BS EN ISO 19650-1:2018. European Committee for Standardization. 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. December 2018.

Cai, B., Liu, Y., Hu, J., Liu, Z., Wu, S., and Ji, R. (2018) Byesian Networks in Fault Diagnosis: Practice and Application. World Scientific. Available at: https://www.worldscientific.com/. Last access: January 2025.

Chen, Z., et al. (2023) A review of data-driven fault detection and diagnostics for building HVAC systems, Applied Energy 339, 121030.

Chiosa, R., Piscitelli, M. S., Pritoni, M., Capozzoli, A. (2024) A portable application framework for energy management and information systems (EMIS) solutions using Brick semantic schema, Energy & Buildings 323, 114802.

Dietrich, D., Bruckner, D., Zucker, G. and Palensky, P. (2010) Communication and computation in buildings: A short introduction and overview. IEEE Trans. Ind. Electronics 57, no. 11, pp. 3577-3584.

Directive 2014/24/EU, of the European Parliament and of the Council of 26 February 2014 on public procurement and repealing Directive 2004/18/EC. Official Journal of the European Union, 28/03/2014, L 94/65-242.

HuginExpert (2020) HUGIN Graphical User Interface Documentation, Release 8.9. May 29, 2020. Available: https://download.hugin.com/webdocs/manuals/GUI/. Last access: Janauary 2025.

Klinc, R., Turk, Z. (2019) Construction 4.0 – digital transformation of one of the oldest industries, Economic and Business Review 21 (3), pp. 393–410.

Korb, K.B. and Nicholson, A. E. (2023) Bayesian Artificial Intelligence, Chapman and Hall, ISBN: 978-1032477657.

ISO 50001:2018/Amd1:2024 - Amendment 1: Energy management systems - Requirements with guidance for use - Amendment 1: Climate action changes. International Organization for Standardization.

Italian Legislation Act no. 36/2023 – Decreto Legislativo 31 marzo 2023, n. 36, "Codice dei contratti pubblici in attuazione dell'articolo 1 della legge 21 giugno 2022, n. 78, recante delega al Governo in materia di contratti pubblici". GU n.77 del 31-03-2023 - Suppl. Ordinario n. 12.

Mallak, A., Weber, C., Fathi, M. (2018) A Graph-based Sensor Fault Detection and Diagnosis for Demand-Controlled Ventilation Systems Extracted from a Semantic Ontology. In: Proceedings of 22nd IEEE International Conference, June 21-23, Las Palmas, Spain.

Martirano, L., Bua, F., Cristaldi, L., Grigis, G., Mongiovi, L., Polverini, S., Tironi, E. (2019) Energy Management Information Systems for Energy Efficiency. In: IEEE Transactions on Industry Applications, pp. 1-1.

Nan Yang, Guigang Zhang, Jian Wang (2020) Research on Knowledge Graph and Bayesian Network In Fault Diagnosis of Steam Turbine. 2020 Global Reliability and Prognostics and Health Management (PHM-Shanghai).

Nota, G., Peluso, D., Toro Lazo, A. (2021) The contribution of Industry 4.0 technologies to facility management. International Journal of Engineering business management, Volume 13, pp. 1–14.

Pasetti Monizza, G., Bendetti, C., Matt, D.T. (2018) Parametric and generative design techniques in mass-production environments as effective enablers of Industry 4.0 approaches in the Building Industry, Automation in Construction 92, pp. 270–285.

Pour R. A., Brenna M., Saldarini A., Zaninelli D., Longo M., Miraftabzadeh S. (2024) Development of Intelligent Models for Buildings Energy Management. In: Proceedings of 2024 IEEE International Conference on Environment and Electrical Engineering and 2024 IEEE Industrial and Commercial Power Systems Europe (EEEIC / I&CPS Europe), Rome, Italy.

Rajad Vara Prasad Y., Rajalkshmi P. (2013) Context aware building energy management system with heterogeneous wireless network architecture. In: Proceedings of the joint 2013 IFIP WMNC Conference, Dubai, United Arab Emirates.

Schumann, A., Ploennigs, J. and Bernard, G. (2014) Towards automating the deployment of energy saving approaches in buildings. BuildSys, pp. 164-167.

PRIN 2017. Project: A Distributed Digital Collaboration Framework for Small and Medium-Sized Engineering and Construction Enterprises, funded by the Italian government, Grant: Prot. 2017EY3ESB.

Vose, D. (2003) Risk analysis – A quantitative guide, John Wiley and Sons LTD, Chichester, 2nd edition.