User Centered Neuro-Fuzzy Energy Management Through Semantic-Based Optimization

Shaun K. Howell, Hendro Wicaksono, Baris Yuce, Member, IEEE, Kris McGlinn, and Yacine Rezgui

Abstract—This paper presents a cloud-based building energy management system, underpinned by semantic middleware, that integrates an enhanced sensor network with advanced analytics, accessible through an intuitive Web-based user interface. The proposed solution is described in terms of its three key layers: 1) user interface; 2) intelligence; and 3) interoperability. The system’s intelligence is derived from simulation-based optimized rules, historical sensor data mining, and a fuzzy reasoner. The solution enables interoperability through a semantic knowledge base, which also contributes intelligence through reasoning and inference abilities, and which are enhanced through intelligent rules. Finally, building energy performance monitoring is delivered alongside optimized rule suggestions and a negotiation process in a 3-D Web-based interface using WebGL. The solution has been validated in a real pilot building to illustrate the strength of the approach, where it has shown over 25% energy savings. The relevance of this paper in the field is discussed, and it is argued that the proposed solution is mature enough for testing across further buildings.

Index Terms—ANN, data mining, energy management, fuzzy logic, genetic algorithm, ontology, optimal control, semantic Web, WebGL.

I. INTRODUCTION

PUBLIC buildings have substantial proliferations of control/automation technologies and tend to experience large discrepancies between “designed” and “operational” energy use, as well as increased user comfort dissatisfaction [1], [2]. Actual energy performance can be considered as the result of a complex combination of, and interaction between, three factors: 1) intrinsic quality of the building; 2) “in use” conditions and user behavior; and 3) energy control and actuation strategy [3], [4]. Whilst altering factor 1) requires complete and costly energy retrofitting interventions, academic evidence suggests that factors 2) and 3) play a determinant role in the energy equation of a building [5]. Managing energy performance implies the ability to monitor and characterize usage patterns whilst understanding user behavior and comfort aspirations in order to devise user-centered real-time energy optimization plans. However, energy control is usually handed to smart systems that 1) do not offer flexibility in responding to unforeseen situations or needs or 2) exhibit a level of complexity that hinders their effective use by facility managers [6].

Moreover, building energy interventions have been designed without taking into account the need to negotiate energy use and desired environmental conditions [1], [6]. Building management systems (BMSs) can be seen as the interface between energy systems and users, including facility managers (FMs). On the one hand, occupants need to feel an engagement with the process of regulating their energy usage in a way that enhances their living and working experience in buildings; conversely, energy control systems should have a level of intelligence and interactivity that promote user-centered and negotiable (multiobjective) energy optimization strategies [2], [5].

Existing BMS in research and industry have shown: 1) various adoption and use problems which suggest a lack of understanding of users’ expectations in terms of levels of automation and functionality; 2) limitations in their capacity to factor in (near) real-time dynamic changing conditions, as well as addressing; and 3) often conflicting multiobjective goals, e.g., reducing energy while enhancing occupants’ comfort and working experience [2]. State-of-the-art research in BMSs involves the use of semantic-based real-time sensing tools [7]–[9] that factor in space occupancy patterns as well as user comfort feedback. However, these tools need to promote more effective energy control strategies through enhanced interoperability with existing energy modeling environments, building control systems, and operational log feeds, and deliver higher-order intelligence (through correlation and analysis of energy modeling predictions and actual use), accessible through more intuitive user-interfaces.

This paper proposes a methodology that exploits finer integration of sensing, interoperability, intelligence, and user interfaces to confer FMs the desired levels of interaction (including automation and functionality) with the BMS to address a wide range of energy scenarios. This builds on prior work [7], [10], following further experimentation with the approach, development of the underpinning software.
platform and algorithmic design, and pilot site validation, which are the focus of this paper. Following the introduction, this paper critically discusses related research, identifying gaps to be addressed by this paper. Section III illustrates the overarching methodology and the various underpinning components delivering a semantic negotiable strategy for energy management. The subsequent sections then detail each of these components, namely: semantic Web middleware (Section IV), rule-based analytics (Section V), and smart GUI (Section VI). This paper then presents the validation of the approach in a public care home building in The Netherlands. This paper discusses the proposed approach and provides concluding remarks and directions for future research.

II. BACKGROUND

A. Toward Economic, Extensible, and Integrated Retrofit BEMS Solutions

Building energy management systems (BEMSs) aim to improve the energy performance of operational buildings. They work on the principle of collecting data about the current state of a building, analyzing this data, then providing feedback to the appropriate decision maker, or reconfiguring the building automatically.

Such a system can be conceptualized in different architectural layers: a sensor layer, computational layer, and an application layer [11]. The sensor layer includes all the energy and environment monitoring devices, the computational layer analyses this data to generate knowledge and desired actions. The application layer then either acts on this automatically, or provides decision support through a user interface which may also send notifications to stimulate behavioral change and feedback. An alternative architecture is presented in [2], which includes a middleware layer between the sensor and analytics layers. This middleware connects the distributed infrastructure of sensors and actuators with the processing engine, and is responsible for handling heterogeneity and interoperability. Other architectural configurations for buildings and smart homes were observed [12], [13], each sharing a similar layered architecture.

The reduction of energy consumption through a BEMS requires it to be economic and engaging for the decision maker, and to deliver integrated, accurate, and attractive measures for energy saving. This requires the extension of the state of the art at both the analytics and interface levels. However, it must also be suitable for retrofit into buildings with existing sensor networks of heterogeneous components and be extensible, as the state of the art continues to improve, so must also innovate in the middleware BEMS layer. To this end, recent advances in each of these three layers are now reviewed in turn: 1) middleware; 2) analytics; and 3) interfaces.

B. Interoperating Legacy Systems With Advanced Analytics—The Role of Semantic Middleware

A flexible and thorough middleware solution is essential to interoperate between the existing sensor and management systems in a building and the novel analytics and visual components of a retrofitted BEMS. Whilst interoperable data exchange protocols are critical [14], and other barriers, such as data quality, integrity, and security exist [15], interoperability of data formats and meaning is a critical challenge in ICT interventions in the built environment [14]–[16]. This highlights the key challenges in energy management interoperability of both shared syntax and semantics between ICT components. These incompatibilities currently require ad-hoc mappings for effective communication and interoperation with a retrofitted BEMS.

Instead, a common vocabulary and conceptual model mitigates the effort required for software artefacts to communicate effectively in an energy management system [9], [14]–[17]. Such artefacts are referred to as semantic models, and are being developed using the Web ontology language (OWL), to facilitate the semantic Web [9], [18], [19], Internet of Things [20], and linked data [21]. These ontological semantic models standardize the description of concepts, relationships, and properties in the domain.

In the built environment domain, the openBIM IFC data model is already experiencing strong uptake [22]. This model uses a less expressive format than OWL, and its federation into OWL is an active area of research [23]. Whilst this does not sufficiently model energy management concepts, its extension toward BEMS would improve the adoption of the resulting model.

The ISES project used an OWL-DL ontology to address interoperability in an integrated lifecycle BEMS [24]. Also, the HESMOS project developed an ontology-equipped framework to integrate distributed and heterogeneous data from ICT building energy systems [25]. However, these projects do not strongly consider their alignment with existing standards, such as the IFC, and do not model occupant behavior. This gap is therefore addressed by the ontological middleware developed for the presented BEMS solution.

C. State of the Art of BEMS Rule Generation and Application

One of the biggest built environment challenges is the need for adaptive, autonomous, and replicable management solutions. Thus, several retrofit building energy management and control systems exist [26], [27], which use intelligent approaches to deal with complexity and uncertainty [28]–[30]. Whilst this can also be achieved through a semantic-based approach [31], this typically requires domain expert knowledge, although automated knowledge discovery processes are emerging [32]–[39].

Several rule generation and knowledge discovery processes exist, such as rule mining [32], combined mining [33], cooperative rules [34], neural network [35], fuzzy logic [36], fuzzy rough set [37], genetic algorithm [38], ant colony optimization [39], hybrid algorithms [7], [40], evolving fuzzy systems [41], [42], decision trees [43], [44], fuzzy classifiers [45], fuzzy pattern trees [46], and rule ripping approaches [47]. These provide a flexible method of approximating rules in a data driven way and highlight the suitability
of machine learning [40] and well-trained ANNs [27] for approximating highly nonlinear problems.

Mitra and Hayashi [40] proposed a neuro-fuzzy rule generation framework, capturing the strengths of both neural networks and fuzzy systems for use in the area of medical diagnosis. Neural networks perform well in data driven processes, providing a continuous learning ability, and fuzzy systems perform well in logic-based systems. Combining the two approaches therefore presents merits in data driven, logical systems. However, they have not compared their algorithm with other prominent rule generation algorithms.

Finally, Pal and Pal [38] proposed a self-organized rule generation process for a fuzzy controller, through a genetic algorithm. This selected the optimal number of rules without supervision, eliminating the need for expert involvement. They tested their solution on an inverted pendulum; reducing the number of rules by circa 95%, and resulting in an integral absolute time error of 0.1019. However, their approach did not optimize the membership functions of the fuzzy inference engine, which could increase the robustness of their approach.

The strengths and weaknesses of the rule generation techniques presented above vary across different types of problem, and they could be improved through multi combined approaches, such as using neural network-based optimization processes. Moreover, they could be extended with advanced partitioning techniques, such as PCA, or other fast classifiers, although PCA may not perform well with large numbers of inputs. This weakness can be avoided by also using multi regression analysis (MRA), where PCA determines the required number of classes and MRA can then determine these classes, using a regression coefficient. Therefore, hybrid processes deliver the strengths of several approaches, especially regarding data driven processes [48], although this has to be logically linked well with other methods.

### D. Toward Engaging Interface for Building Energy Monitoring and Decision Support

The final layer of the BEMS system is the application layer, which allows the FM to interact with the system’s data monitoring, analytics, and actuation capabilities. Several commercial energy monitoring tools exist, which allow the monitoring of energy consumption in a building. MonaVisa is a product which is retrofitted alongside an existing BMS [49]. This collects temperature and CO2 sensor readings and assesses these against a comfort range, generating a notification when a KPI leaves this range. These assessments are conducted at different time scales for each monitored room and are delivered through a GUI. PlugWise is an energy monitoring tool which transmits energy readings over the ZigBee protocol. This allows additional sensors to be added to monitor temperature, motion, gas, and electricity. Again, collected readings can be viewed as charts and graphs for each metered appliance over varying time scales, and overlaid onto a 2-D floorplan.

Increasingly, these sensing, analytics, and actuation services are delivered through WebApps. These aim to provide engaging interfaces with seamless cross-platform deployment. HTML5 provides a flexible and extensible means to meet the requirements of many tools, and Asynchronous JavaScript and XML (AJAX) and SPARQL queries can be used to access the underpinning knowledge. Further, WebGL facilitates 3-D visuals in HTML5 Web pages without the need for browser plugins, as HTML5 is supported natively by modern browsers. This is highly beneficial because it allows deployment across operating systems, other Web page elements can form part of the GUI, and the visuals can make use of a number of high-level communications tools, such as AJAX. The 3-D graphics software interface to WebGL is written in JavaScript, which allows the use of the document object model to manipulate the Web page, and allows the visualization to be manipulated by standard Web form controls. Finally, as this allows the seamless integration of 3-D visualizations with Web technologies, it allows the computationally expensive simulation and analytics tasks to be performed on the server side, with only the rendering of 3-D data performed by the user’s Web client.

### III. OVERVIEW OF PROPOSED SOLUTION

The research and development of a novel BEMS was undertaken through an EC FP7 project [50] and tested within a mixed mode residential care home in The Netherlands. The project aimed to produce a BEMS, which could be retrofitted into public buildings with minimal investment, to exploit an enhanced sensing infrastructure and the existing BEMS, augmented with analytics and visualization components through a semantic Web approach. This involves a semantic knowledge base, which describes the physical properties of the building as an extension of the openBIM IFC data model [18], [23], through an RDF store and SPARQL endpoint. The semantic model in the knowledge base also contextualizes the historical data stored in an MySQL database by formalizing a shared meaning. The novel analytics include the automated production of rules through simulation-based rule generation [7] and their subsequent fuzzification alongside rules from mining on historical metering data. The visualization component utilized an HTML5-based smart GUI to deliver engaging 3-D WebGL visuals alongside real-time and historical energy performance monitoring and decision support, by presenting the optimized rules as user-friendly actuation suggestions. The BEMS aimed to promote trust with FMs through a negotiation-based user-in-the-loop approach. This meant the FM was responsible for actuating the suggested changes, as this was attractive to industrial partners due to liability and legislation concerns around automated actuation. Finally, the semantic Web-based approach aimed to promote reusability and extensibility, by allowing the deployment of the BEMS in further buildings without redesign of its underlying technologies, as was tested through four other European pilot sites within the project. This paper focusses on presenting the enhanced BEMS and delivering proof of concept at the selected pilot site. The following sections therefore discuss the key components of the proposed system’s service-oriented architecture; the RDF store, SPARQL mapper, and knowledge base which constitute the semantic middleware, the data mining engine, rule engine, and fuzzy real time reasoner, which constitute the system’s analytics components, and the system’s smart GUI, as shown in Fig. 1, before a pilot site validation is presented.
The numbers H1-H5 and R1-R6 in Fig. 1. describe the two data flows involved in the proposed solution, i.e., historical and real-time data flows. Each data collected from sensors is transmitted through Web service interface into the MySQL database periodically in a certain interval resulting in a collection of historical data (H1). Through the BuildVis interface, the user can query the historical data to perform performance monitoring, for example to monitor the energy performance of certain building zone in a specified time range (H2–H3). The historical data stored in the MySQL database are then used as training data by machine learning algorithms to generate rules (H4). The resulting rules are transformed into SWRL rules and integrated into the knowledge base (H5). The rule generation is performed in a larger interval to update the knowledge, for example once in a month.

Through the Web service interface, the fuzzy reasoner collects data from the sensors in real time (R1). Then, it invokes the appropriate rules, i.e., rules with certain weights in the knowledge base that have been selected by the user (R2). The fuzzy reasoner fires the rules by setting the variables in the condition part with the values collected from the sensors (R3). Through BuildVis GUI, the user can define an energy saving goal of a certain category in his building, for example 10% energy saving for heating (R4). The knowledge base returns the suggestions containing set points values of different actuators to achieve the desired goal (R5). Subsequently, the user could set the set points corresponding to the suggested values (R6).

To summarize the relationships between the core analytics components: a genetic algorithm generates energy-saving rules, using an ANN as the cost function (as a surrogate for the thermal simulation), these rules map the current building state and actuator states to optimal actuator states for the imminent future. The rules are fuzzified and then stored in the knowledge base, and updated on a periodic basis (e.g., weekly). The rules are used by the fuzzy reasoner at runtime alongside actual sensor data, where the fuzzy reasoner recommends the best actuator state given the current observed building state and actuator states.

IV. SEMANTIC WEB MIDDLEWARE

A. Role of Semantic Middleware

As mentioned, a critical problem in retrofitting advanced analytics into existing buildings is the range of heterogeneous data sources and existing BEMS solutions encountered: such as (in our pilot case study) Priva, Controlli, and EUGENE. This was overcome through a key novelty of this paper; the knowledge base and accompanying software which served as the integration components of the proposed energy management system. It integrates heterogeneous data sources required by the system, and also provides some of the intelligence capabilities through reasoning on the rules and structures contained in the knowledge base.

Each of existing BEMS solution uses different communication protocol, for example, EUGENE uses Modbus and Priva uses BACNet. However, they provide Web service REST interface. The data are transmitted from those BEMS solution to the middleware layer through REST Web service. We developed a program to perform the mapping between the Web service schema and our knowledge base model.

The approach of a semantic middleware solution was adopted over traditional options to facilitate reuse and extensibility in the BEMS domain and the wider domains of smart cities and the Internet of Things, and to build the BEMS solution in line with the wider trend toward Web-based software. Through this approach, the proposed solution could be deployed in further buildings regardless of the proprietary data schemas and protocols used by their previously installed sensing, actuation, and BEMS infrastructure, and could be used to integrate building energy management with energy.
management at the district scale, such as, where renewables or microgrids require active and collaborative management [51].

B. IFC-BEMS Domain Ontology

The OWL was used to represent the knowledge base in order to achieve a high degree of expressiveness of the knowledge model. The knowledge domain model consists of classes representing building physical elements that are observed and analyzed in energy management activities, and building controls consisting of sensors, controllers, alarm, etc., which act as observer and controller of physical building elements. Furthermore, the knowledge model represents the human actors and their behaviors that can affect the states of building physical elements. In the knowledge model, the states are classified into simple states, for example window or room states, and complex states, which are built by relating several simple states. Energy efficiency and comfort degrees are examples of complex states. This resulted in 145 asserted classes, 43 object property slots, and 43 data property slots; the key physical and sensory classes and relationships are shown in Fig. 2.

In order to provide the possibility to reuse existing industrial standards, the knowledge domain model is aligned to IFC model, as also shown in Fig. 2. The alignment is done by defining the explicit IFC-OWL mappings that are stored in the class annotations. For example, the IFC entity IfcWindow is mapped to OWL class Window using the annotation correspondToIfcEntity. The other main IFC concepts which were reused were the physical building elements and geometries, such as doors, walls and openings, and the key extensions included descriptions of the zones, sensors, states, people, and behaviors in the domain. In total the domain ontology asserted 44 mappings to IFC concepts. This allowed an automatic IFC to OWL document conversion using SPARQL queries [23].

C. Population of Pilot Site Knowledge Base

The domain ontology model only contained classes, relations among them, and definition of their properties. In order to apply the knowledge base in a specific building, the ontology had to be populated with instances corresponding to the objects in the building that are considered essential for the energy management activities. Most current building layouts are only drawn as 2-D sketch using CAD applications, such as AutoCAD [52]. They contain only geometrical primitives, such as lines, curves, points, etc. Therefore, in order to populate the ontology, the semantic information of the sketch had to be extracted. OntoCAD is an open source tool that was developed to solve the problem. The tool clusters the geometric primitives in layers. Using the tool, we defined templates representing semantic objects, such as doors, rooms, and chairs, and select the areas in the drawing which corresponded to the to-be-generated ontology instances. The tool updated the property values of the generated instance automatically, such as the position and the perimeter. OntoCAD also allowed the validation and correction of the knowledge population, where necessary [53].

The knowledge base also embeds SWRL rules, which are generated automatically using both historical metering (generated through data mining) and simulation data. Each rule is equipped with a weight indicating the confidence of the rule. The weight has values between 0 and 1. These are used by the fuzzy reasoner to evaluate the importance of the rules. This is necessary to account for the large number of rules generated by the data mining and simulation modules. As well as these custom rules, the ontology deployment performs inference through the Jena inference module. This allows new knowledge to be produced automatically by the software from the stated axioms, resulting in inferred knowledge being used alongside explicit knowledge. For example, if a sensor is stated to be connected to a specific element, as a property of the sensor, then the software infers as a property of the building element, that the element has that sensor connected.

D. RDF Store and SPARQL Endpoint

This module is the main communication module between the knowledge base and the smart GUI. The knowledge base stores all the data about the building and its systems relevant to
DATA ENGINEERING GROUP IN TRINITY COLLEGE DUBLIN. Each based virtual platform, and maintained by the Knowledge and a Fuseki server which is running on a Linux operational system-information is also stored as an OWL file and uploaded into add additional information as sensor types and locations. This tool OntoCAD is used to identify zones in the building and into RDF and stored on the Fuseki server. The data extraction the BEMS. To enable visualization of the building floor plan, an existing 2-D DWG file is parsed and converted directly into RDF and stored on the Fuseki server. The data extraction tool OntoCAD is used to identify zones in the building and add additional information as sensor types and locations. This information is also stored as an OWL file and uploaded into a Fuseki server which is running on a Linux operational system-based virtual platform, and maintained by the Knowledge and Data Engineering Group in Trinity College Dublin [54]. Each pilot building has its own instance of Fuseki server to store the building specific knowledge base. The smart GUI queries the ontology using a combination of AJAX and SPARQL (SPARQL Protocol and RDF Query Language). When the FM selects the pilot building through the smart GUI, several SPARQL queries are made to the Fuseki server, one of which returns JSON objects which are then used to store a 2-D array of JavaScript zone objects, which describe each zone in the building. A query example is shown in Fig. 3. This would be enough to display the zones graphically (using WebGL), although as each property is returned as strings, perimeters must be parsed client side to get each point given in Fig. 3.

V. OPTIMIZED RULE-BASED ANALYTICS

In order to enhance the reasoning capabilities of the knowledge base, we integrated rules from data mining over sensor data, and rules from thermal simulation-based optimization. The rules are represented with SWRL in order to allow the integration into the knowledge base. The data mining rules are mainly used to identify inconsistent performance and to predict energy consumption in the building. Conversely, the simulation-generated rules aim to impose optimal set point configurations toward the negotiated target energy saving. Both rule types are critical to the BEMS’s capability to assist FMs in improving energy efficiency in the building. The main reason for utilizing the simulation-based rules in the proposed methodology was the complex behavior of the building environments, which could not be fully captured by rules without a simulation model and a robust intelligent solution. The following sections present the generation approaches of both rule types. Nevertheless, this paper focuses on simulation rules and only introduces data mining rules briefly, as they are described in [23].

A. Extraction of Rules Through Data Mining on Historical Metering Data

The objective of the data mining was to identify correlations between indoor and outdoor sensor data, user behaviors, and energy consumption data, and to express these as rules. The rules were then federated into SWRL rules in the knowledge base to enrich each building’s model. Reasoning on the rules generated new knowledge that can be utilized for the following goals.

1) Prediction of the energy consumption of certain user activities, building zones, and appliances.
2) Detection of energy consumption anomalies in user activities, zones, and appliances.
3) Inference of user activities in building or zones based on contextual sensor data.
4) Fault detection in appliances, based on their energy consumption.
5) Prediction of actuator states or configurations toward meeting specific comfort levels.

These intelligent capabilities were achieved through the collection and algorithmic analysis of the following relevant sensor data.

1) Indoor Sensor Data: Zone temperatures, CO₂ concentrations, and door and window states.
2) Outdoor Sensor Data: Dry-bulb temperature, precipitation rate, wind speed, brightness/luminance, and air humidity.

To allow different analyses at different aggregation levels, energy consumptions were collected using energy meters at various levels. At the appliance level, energy meters were installed at active power sockets. At the zone level, energy meters were installed at the distribution board for the target zone. At the building level, energy meters were installed in the central distribution board.

Behavioral data were then collected; mainly based on the usage of appliances and zone occupancies. That meant that if a user undertook multiple activities in a zone without changing the appliance usage, those activities were not considered as different behaviors. Key daily periods were identified, where similar behaviors were observed across days: lunch time, office hours, coffee break time, maintenance/cleaning time, and nonoffice hours.

The rules reflecting interrelationships between behavior, surroundings parameters (temperature, humidity, etc.) and energy consumption were generated through decision tree-based classification algorithms, such as C4.5 [43]. Each path in the decision tree from the root to the leaf constitutes a rule.

B. Simulation-Based Optimized Rule Generation

This system module used a 6-staged process to produce energy saving rules based on thermal simulations of the building, as shown in Fig. 4. This approach uses preprocessing to produce optimization scenarios and simulation data, and to identify sensitive variables, then trains an ANN based on this data. This ANN is then used as the cost function in a GA optimization to output actionable rules, which are then evaluated for efficacy.
1) Building Thermal Simulation and Sensitivity Analysis:
The preprocessing stage consists of scenario definition, simulated data generation, sensitivity analysis, and variable mapping. The scenario defines the objectives of the optimization and the available control variables, actors, and sensors. Thermal simulation and data generation involves thermal model development and utilization for each building. Sensitivity analysis and variable mapping then determines the most sensitive variables, and maps them with the building’s artefacts, as expressed in the knowledge base.

In this paper, a public residential care home in The Netherlands, named “the Forum,” was used as a case study, based on the scenario shown in Table I. A thermal simulation model of the building was created in DesignBuilder, as shown in Fig. 5, which includes detailed material, occupancy, and construction data.

EnergyPlus was used to produce simulated data across the permutations of the scenario’s independent variables. In the Forum building, the four actuators resulted in 32 permutations, so the annual simulation was repeated to produce 32 datasets. PCA and MRA were then used to reduce the simulation model’s 954 reported variables. The ideal reduction was determined by PCA, and then MRA was used to rank the variables’ sensitivity according to the scenario’s objectives. This process was modeled as: 1) where $F_j$ denotes either thermal energy consumption or predicted mean vote (PMV) in this case [7]

$$ F_j(\overrightarrow{\text{Var}}) = \sum_{i=1}^{\text{numvar}} \text{coef}_{ji} \text{Var}_i. \quad (1) $$

In (1), $\overrightarrow{\text{Var}}$ denotes the variables generated from the simulation, $\text{coef}_{ji}$ denotes the coefficient of variable $\text{Var}_i$ for $F_j$, and $\text{numvar}$ is the available number of variables.

The identified variables are then mapped with the existing sensors installed in the target building. Variables which cannot be mapped to sensors can inform the acquisition of additional sensors or can be excluded from subsequent stages of the process. The list of mapped sensors for the Forum building are given in Table II, and were used in the following ANN-GA rule generation.

2) ANN-Based Learning Process: ANNs predict the behavior of highly nonlinear systems, such as building energy systems [29], by conducting machine learning over training data. ANNs have been researched in energy management systems for the last two decades [55], yet they continue to perform competitively [56], and as such are still the most widely used type of data-driven model for building energy prediction [57] in research. Hence, this paper also utilizes an ANN-based learning method, where the novelty of the proposed system is the use of this traditional method in a unique way alongside GA, behavioral data mining, fuzzy rules, and ontology technologies, within an end-to-end BEMS. Following experimentation, a traditional multilayer perceptron-based ANN approach was found to perform adequately, although there is room for further investigation into deep architectures and other types of data-driven models, which could be interchanged with the 3-layer MLP used if found to perform better. The proposed ANN design used the ten variables identified previously as inputs, as well as

<table>
<thead>
<tr>
<th>Mapped Sensor</th>
<th>MRA Coefficient for Objective 1</th>
<th>MRA Coefficient for Objective 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Outdoor Air Drybulb Sensor</td>
<td>72.18</td>
<td>140.11</td>
</tr>
<tr>
<td>Wind Speed Sensor</td>
<td>102.57</td>
<td>1118.23</td>
</tr>
<tr>
<td>Wind Direction Sensor</td>
<td>1807.64</td>
<td>68.12</td>
</tr>
<tr>
<td>Direct Solar Radiation</td>
<td>66.75</td>
<td>312.90</td>
</tr>
<tr>
<td>Solar Azimuth Angle</td>
<td>7960.03</td>
<td>72.13</td>
</tr>
<tr>
<td>Solar Altitude Angle</td>
<td>417.29</td>
<td>209.34</td>
</tr>
<tr>
<td>Zone Mean Air Temperature Sensor</td>
<td>2001.26</td>
<td>9531.1</td>
</tr>
<tr>
<td>Zone Air System Sensible Heating Rate (post processed) Sensor</td>
<td>153.11</td>
<td>107.32</td>
</tr>
<tr>
<td>Zone Ideal Load Total Cooling Rate, (post processed) Sensor</td>
<td>63.06</td>
<td>577.01</td>
</tr>
<tr>
<td>Occupancy Sensor</td>
<td>303.12</td>
<td>462.22</td>
</tr>
</tbody>
</table>

![Fig. 4. Simulation-based rule generation method environmental variables.](image1)

![Fig. 5. Thermal model for forum building’s atrium zone (pilot zone).](image2)
the four actuator states at the current timestep, and time information. The outputs were then the zone’s PMV and energy consumption at the subsequent timestep, as shown in Fig. 6.

For an ANN to be effective, it must be well-trained and use an appropriate topology. To ensure this, the learning algorithm, number of hidden layers (and their number of process elements) and transfer function have to be determined robustly. In this paper, several experiments were designed and conducted to determine the optimum ANN parameters. In the experimental design, an iterative parameter tuning approach is utilized. The initial configuration set is selected as: single hidden layer with five neurons, gradient descent-based learning algorithm, tangent-sigmoid transfer functions in hidden and output layers, 0.0001 error rate, 4000 epochs for number of hidden layer, number of process elements in hidden layer, learning function, error rate and number of epochs, respectively. The next stage is changing one of the parameters while keeping others constant, if the error rate with the selected parameter is better than its constant value will be updated for further parameter selection. The best parameters were found to be: a single hidden layer of 30 neurons, using a Levenberg–Marquardt-based learning function, logarithmic sigmoid and tangent sigmoid-based transfer function in hidden and output layers. Using these parameters, the desired error rate (0.0001) was achieved at 70th epoch.

The ANN was trained with 80% of the dataset and tested on the remaining 20%, within MATLAB. The ANN architecture and training decisions are described further in [7]. This model was then used as the cost function of the GA rule production. The univariate hyperparameter search approach yielded an ANN with sufficient performance within the time and compute limitations of the work, however, further work includes optimizing the ANN design further through grid search or a similar technique.

3) GA-ANN-Based Optimized Rule Generation: The rule generation is based on finding optimized solutions for the set of control variable with related environmental variables, desired optimization level (i.e., 5%, 10%, 15%, 20%, 25%, and 30%), and time information. Once an optimum solution is found, this optimum solution and related environmental parameters, date-time info, the achieved improvement level, zone ID, and a weight based on the achieved and desired improvement in the target variable.

GA optimization was used with an ANN cost function. GA is a very popular optimization technique for complex problems [7], [30]. The proposed approach uses the actuator states alongside sensor data in the chromosome string, and uses mutation, crossover and fitness evaluation to iteratively improve the rule in a stochastic manner. The general formation of a chromosome string is shown in Fig. 7.

The proposed chromosome string includes two groups: 1) variable and 2) constant features. The variable group includes the control variables (temperature setpoint, window setpoint, blind setpoint, and shading setpoint). The constant group of the string consists of the values of the sensitive variables and time information which are denoted from $X_5$ to $X_{17}$ for month, day, hour, outdoor temperature, wind speed, wind direction, solar irradiation, solar azimuth angle, solar latitude angle, zone air temperature, zone heating rate, zone ideal total cooling rate, and occupancy, respectively.

Only the control variables ($X_1, \ldots, X_4$) are involved in the mutation and crossover operations of the GA process, and the other string elements are kept constant to determine the optimized value for the control variables. The relationship between cost function variables (inputs and output) is presented in

\[
\text{Minimize: } F_{\text{energy consumption}}(X_1, X_2, X_3 \ldots X_{17})
\]

Subject to constraints:

\[
\begin{align*}
|F_{\text{PMV}}(X_1, X_2, X_3 \ldots X_{17})| &< 1 \\
16 &\leq X_1 \leq 24 \\
0 &\leq X_2 \leq 1 \\
0 &\leq X_3 \leq 1 \\
0 &\leq X_4 \leq 1.
\end{align*}
\]

$F_{\text{Energy Consumption}}$ is the energy consumption amount based on the variation of the control variables $X_1, X_2, X_3,$ and $X_4$ while keeping other variables ($X_5, \ldots, X_{17}$) constant, and $F_{\text{PMV}}$ is constraint named as the PMV function value to keep the thermal comfort under between $-1$ and $1$.

The genetic algorithm’s crossover operation used a multipoint gene exchange within the variation groups of two parents’ chromosome strings as shown in Fig. 8. The mutation operation also acted only on the parents’ variation groups, where it selected one or more elements according to a probability value as shown in Fig. 8. Both the chromosome and the mutation operations are implemented on the $p$ and $r$ worst regions (solution sets) based on their fitness values, as shown in Fig. 9.

The algorithm used an elite selection approach, where the best $n – p$ solution were kept as original, the crossover
Fig. 8. Crossover operations in the proposed GA.

Fig. 9. Mutation operations in the proposed GA.

Fig. 10. General formation of the elitism process in the proposed GA.

Fig. 11. Example of the generated optimized rules.

operation acted on the remaining $p$ chromosome strings, and the mutation probability was $\alpha$, using the roulette technique. The mutation operation was also implemented on $r$ worst individuals with a $\beta$ probability rate. Hence, the best solutions are kept in the solution pool as shown in Fig. 10.

The primary stopping condition of the optimization was the target improvement decided by the FM. The FM negotiates an acceptable set of actuations by choosing a target, such as 30% energy reduction, then observing the optimized actuations required, and either accepting these or adjusting the target. An example of the generated rule is shown in Fig. 11, and the overall GA-ANN-based process is shown in Fig. 12.

C. Fuzzy Reasoner

The rules produced by the GA-ANN process are stored as SWRL rules, but are used by a fuzzy reasoner. Fuzzy logic is inspired by the human, approximation-based, reasoning process [58]. This process rationalizes an appropriate output from inaccurate and incomplete information. The proposed fuzzy reasoner communicates with the GUI through the mapper module, and the knowledge base through the Java expert system shell [59], as shown in Fig. 13. In this paper, a Mamdani fuzzy inference system was utilized: despite this approach’s simplicity, it was found to provide adequate performance.

Although the rules are generated automatically through machine learning, the user ultimately decides which rules should be applied. The weights are initially set automatically corresponding to the confidence of the rules, but the user is
TABLE III
RULE VARIABLES FOR THE FORUM BUILDING

<table>
<thead>
<tr>
<th>Variables in Antecedent Part (IF Antecedents)</th>
<th>Variables in Consequent Part (THEN Consequents)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1- Zone ID, 2- Rule weight, 3- Desired objective type, 4- Reduction level, 5- X5= Month information, 6- X6= Day information, 7- X7= Hour information, 8- X8= Outdoor temperature, 9- X9= Wind speed, 10- X10= Wind direction, 11- X11= Solar radiation, 12- X12= Solar azimuth angle, 13- X13= Solar altitude angle, 14- X14= Zone mean air temperature, 15- X15= Zone air sensible heating rate, 16- X16= Zone ideal total cooling rate, 17- X17= Occupancy.</td>
<td>1- X1= Temperature set point, 2- X2= Window set point, 3- X3= Shading set point, 4- X4= Light set point.</td>
</tr>
</tbody>
</table>

Fig. 14. Fuzzy membership function example.

Nine hundred fifty eight rules were generated for each objective, resulting in 3882 rules in total. The inference engine then implements the membership-based conversion given in (9), where \( \mu_{V_k} \) is the membership value of the output variable. An example of converted fuzzy rule is presented in Fig. 15. The inference engines design is based on the experts’ experiences.
Fig. 15. Example fuzzy rules presented in inference engine.

The defuzzification process then determines the selected output values for any given input set. This operation uses a rule weighting method, which increases the accuracy of the fuzzy system [65]. The weight given to each rule before the fuzzification is also included as a coefficient in the reasoning process, as shown in

$$V_{\text{crisp}} = \frac{\sum w_i \mu_Y(Y) Y}{\sum w_i \mu_Y(Y)}.$$  \hspace{1cm} (10)

The rule weight is determined by the closeness of the expected target value to its desired value, evaluated by simulation during the rule generation process. The weights are calculated according to (11), where $w_i$, $y_i$, and $\hat{y}_i$ are the $i$th rule’s percentage weight, best solution found, and expected target optimization level, respectively

$$w_i = 100 \left| \frac{y_i - \hat{y}_i}{w_i} \right|.$$  \hspace{1cm} (11)

To summarize, SWRL rules are generated through an optimized ANN-based approach (ANN-genetic algorithm), for different reduction levels, which is the basis of the theoretical rule generation process, the generated rules are then converted into fuzzy rules to create the rule base of the fuzzy inference system by inclusion of the linguistic transformations. Once a user desired level of reduction for a desired objective is received then the fuzzy inference engine is to utilize these inputs for its inference engine and to determine the most convenient outcomes in the existing post-processed SWRL rules. After determining the consequent, the rule engine searches for rules with the same actuator states and sorts them according to their weights stored in the knowledge base. The highest weighted one is selected as a response for the users.

VI. SMART GUI

This section describes the implementation and features of the front-end tool and how it accesses the different data sources to enable monitoring and visualization of the relevant static and dynamic data, and the display of suggestions to the FM. The interface has been evaluated to determine its level of usability, the resulting findings determined that over five demo objects, one of which forms the core of the evaluation presented here, the FM were supported in the task of identifying and applying suggestions [66]. The BEMS interface was implemented using modern Web languages and the bootstrap framework [6]. The interface contains three main windows; Fig. 16 shows the WebGL view of the building’s zones and Fig. 17 shows the energy monitoring and actuation suggestion window. The interface also has a menu “choose building,” so that the FM can select different buildings, if they are responsible for more than one. First, the ability to view the static properties and the historic and current energy KPIs of each building and zone under an FM’s remit is discussed. Second, the role of the GUI in presenting the knowledge from the solutions’ various analytics components, in the form of suggested actions, is presented.

A. Building Zone View and Performance Monitoring

A 3-D visualization of the building’s thermal zones was seen as a key requirement of an engaging tool, so this was enabled by converting 2-D CAD plans into semantic models in the knowledge base. As well as showing an extruded floor plan of the building, each zone is described in the knowledge base by its geometric properties, function (kitchen, atrium, etc.), ID, and its connected sensors, and these are all displayed after clicking the zone, which triggers a query of the knowledge base.

After choosing a zone and a sensor type (or multiple types), the energy monitoring interface shown in Fig. 16 allows the FM to view the current and historic performance of the zone. This is achieved through a histogram of sensed data values and a traffic light graphic which indicates the acceptability of the current performance, relative to its mean value. The historical sensor data is retrieved from the SQL database using a combination of AJAX and PHP server-side scripting. SQL was chosen due to the speed at which it can handle queries for large amounts of historical data.
The suggestions are tailored to a particular zone, which is generally a room. The tool also shows 3-D coordinates for sensors, based on industry foundation class data, or manual data entry. Incorporating IFC Cartesian locations represents ongoing work. Also, the color of each sensor’s icon represents its current state.

B. Optimized Suggestion and Negotiation Interface

To deliver the knowledge produced by the solution’s analytics, and hence support the FM in reducing the energy consumption of the building, the interface displays suggested actions as part of a negotiation interface based on the data mining, theoretical rules, and their fuzzification. Fig. 18 shows how the FM configures these criteria using drop down menus and slider bars, generated by jQuery selectors. Once the FM has selected a zone, chosen a goal type (e.g., reduce electricity consumption) and moved the slider to the target energy saving (e.g., 20%) they press the “query suggestions” button. This uses AJAX and PHP to query and return suggestions based on the rules produced by the back-end analytics and displays a number of recommended actions, such as adjusting the blinds or heating temperature set point.

Critically, the FM’s expert knowledge is then utilized to determine if the suggested actions are appropriate, as the simulated implications on the building are then displayed in the energy monitoring histograms, and the FM chooses whether or not to act on the suggestions. If they deem the savings to have negative implications on comfort, or otherwise, they can adjust the query criteria and view more suitable suggested actions. This means of control was a requirement of the solution, as FMs during the aforementioned usability evaluations indicated that they wished to have final say on whether to enact changes.

VII. RESULTS

To evaluate the performance of the developed solution, the system’s intelligence was tested in the EnergyPlus simulation environment and the full system was deployed in a real pilot building, so as to validate the entire system, including the semantic middleware and GUI components. The pilot building was a public residential care home in The Netherlands (the Forum), and the decision support capabilities of the system were tested for the building’s 3-storey atrium zone; the main energy consuming space of the building. As the Forum building is primarily an elderly care home, maintaining thermal comfort was critical whilst attempting to reduce the building’s energy consumption by using the suggested actions of the system.

Initially, the proposed solution was tested by simulating the zone’s energy consumption over a day and then repeating the simulation with the optimized energy saving rules, applied at the start of each timestep. This reduced the energy consumption from 258 to 201 kWh, whilst maintaining an absolute PMV of less than 1, which was deemed an acceptable level of occupant comfort. In contrast, the well-known rule-based systems RULE5, RULE3, and C4.5 only achieved energy consumptions of 258 kWh, 259 kWh, and 259 kWh, respectively, with the absolute PMV values increasing to 1.7, which represents greater discomfort. The generated set points and resulting energy consumption profiles from these experiments are shown in Figs. 19 and 20.

Following preliminary success, the simulation was extended to a two-month period. Using the proposed method the energy consumption was reduced from 14 600 to 11 400 kWh during the months of October and November, whereas RULE5, RULE3, and C4.5 achieved 13 500 kWh, 13 900 kWh, and 15 400 kWh, respectively, as shown in Fig. 21. Again, the proposed approach maintained an absolute PMV of less than 1.

The full retrofit BEMS solution was then deployed in the pilot building, initially for a single day and subsequently for an extended period from October 1, 2014 to January 20, 2015. In each of these tests the FM utilized the system’s decision support to receive suggested actions for energy saving, and after negotiating the severity of these, actioned them through local control systems. Based on the single day experiment, the daily energy consumption was reduced from 77 to 58 kWh as illustrated in Fig. 22. Over the two month period, the total energy consumption reduced from 7500 to 5600 kWh, when adjusted for degree day temperature correction, as shown in
In this paper, the state of the art and previous research was discussed within each of the conceptual layers of a retrofit BEMS. A novel BEMS was then introduced and the components and methodology of each of its layers were discussed in turn. First, the semantic middleware layer was introduced as a key novelty, and its benefits of interoperating a building’s devices and systems in an extensible, replicable and affordable manner was explained. The methodology of instantiating a domain ontology aligned with international standards was presented through the use of OntoCAD to populate an extended version of the IFC data model. Second, the solution’s intelligence was explained as a combination of intelligent rule generation techniques and a fuzzy reasoner. The combined use of rules generated through data mining and simulation-based optimization through SWRL ontology integration was shown. Finally, the GUI of the solution was explored; its interactions with the back-end to present zone-based performance monitoring and optimized rule suggestions were explained. Also, the client-side software decisions of WebGL and HTML5 were discussed as a means to enable cross platform deployment without requiring additional user downloads, whilst still providing a 3-D interface and many developer benefits toward further maturing the solution. Through a simple traffic light graphic, FMs can determine the zones requiring attention, and the pop ups alert the FM when a new energy saving suggestion is made. This type of feature would be ideal for mobile integration, so that FMs can be alerted in the field.

The solution was tested within both simulated and real buildings, with encouraging results in both cases. Both cases showed significant energy savings over both a single day and a period of several winter months, with the real building displaying circa 25% energy savings on average. Whilst these results are highly positive and serve as a proof of concept, further work is now required to demonstrate the solution’s replicability across other buildings. Other features which are of interest for development include the use of a wizard to help the FM with tasks, and providing multilingual support to allow deployment across countries; as driving FM engagement with the tool through an attractive and intuitive interface is a key contribution of the work.

Whilst the individual components used in the proposed system delivered sufficient performance, key ongoing work includes further optimization of each. For example, the ANN model implemented could be interchanged with a more advanced deep learning model, and its hyperparameters could be further optimized via a dense grid search or similar.

Given the successful deployment of the solution and the key novelties identified, this paper demonstrates the potential of a cloud-based approach to a retrofit BEMS solution by using semantic middleware as a system integration component alongside a human–computer negotiation process, advanced AI and an engaging user interface. The BEMS presented can therefore act as a reference point for similar solutions in terms of the energy saving potential, upfront investment reduction through system integration, and logistics and liability issue mitigation regarding AI control of building systems.

**VIII. DISCUSSION AND CONCLUSION**

This paper has presented a retrofit BEMS capable of delivering energy savings through analytics across existing data sources and actuators in a building, by using semantic middleware to integrate heterogeneous devices within a cloud-based, service-oriented architecture. As well as the novelty of the semantic approach, the solution represents a step change by encouraging the use of AI by FMs, by respecting the FM’s role in the decision process and using an engaging GUI, and the solution has been successfully deployed in a public building in The Netherlands.
ACKNOWLEDGMENT

The authors would like to thank the indirect contributions of the research partners, especially CETMA for their contribution toward the fuzzy aspects of the KnowholEM project and Kiril Tonev for his role in the software implementation of the interoperability layer.

REFERENCES


