energies

Article

Human Perception and Building Automation Systems

Daniel Ramsauer [1,*], Max Dorfmann [2], Hilda Tellioğlu [2] and Wolfgang Kastner [1]

[1] Institute of Computer Engineering, Automation Systems, TU Wien, 1040 Vienna, Austria; wolfgang.kastner@tuwien.ac.at
[2] Institute of Visual Computing and Human-Centered Technology, Artifact-Based Computing & User Research, TU Wien, 1040 Vienna, Austria; max.dorfmann@tuwien.ac.at (M.D.); hilda.tellioglu@tuwien.ac.at (H.T.)
* Correspondence: daniel.ramsauer@tuwien.ac.at

Abstract: Building automation is concerned with closed- and open-loop control of building services such as heating, cooling, ventilation and air conditioning, lighting and shading. The ultimate goal is to reduce energy consumption while providing comfort for the occupants. However, ensuring human comfort is a complex affair. In case of dissatisfaction, users need to inform the building operators about apparently badly adjusted setpoints. Then, service units of the facility management have to manually analyze how to improve the situation. Due to the complex characteristics of human perception and derived feedback, this can become a troublesome and time-consuming task. This paper describes the main results of our investigations to improve occupant comfort in office buildings using environmental information monitored by a Wireless Sensor Network (WSN) and human perception collected from a feedback tool. A joint information base aligned with static data from building information modeling integrates the information gathered. Reasoning on these data sources allows adjustments of the Building Automation System (BAS) to automatically enhance the tenant's comfort or suggest necessary adjustments for facility managers. Communication between the different system components is handled via Message Queuing Telemetry Transport (MQTT). A real-world field study shows the potential of the developed approach, proves its feasibility, and demonstrates the functionality of the feedback tool.

Keywords: building automation; occupant feedback; building information modeling; building management; prototyping

check for updates

Citation: Ramsauer, D.; Dorfmann, M.; Tellioğlu, H.; Kastner, W. Human Perception and Building Automation Systems. *Energies* **2022**, *15*, 1745. https://doi.org/10.3390/en15051745

Academic Editors: Anastasios Dounis and George S. Stavrakakis

Received: 6 January 2022
Accepted: 21 February 2022
Published: 25 February 2022

1. Introduction

In the last few decades, there has been a growing interest in integrating building management into increasingly complex buildings, which is due to the rising performance requirements demanded by users and facility owners. The requirements stem from various business activities and allocation of different types of space, demanding more efficient use of facilities provided by the building. Further, reducing the running costs of a building and ensuring proper maintenance of the building's equipment, alongside environmental considerations, enable a high-performing facility [1,2]. Therefore, Facility Management (FM) plays an important role in enhancing the overall performance of an organization's building. Per the definition of the International Facility Management Association (IFMA), FM is "the practice of coordinating the physical workplace with the people and work of the organization. It integrates the principles of business administration, architecture and the behavioral and engineering sciences" [3]. Information and Communication Technology (ICT) supports FM in managing the complexity of large buildings by providing tools for supplying necessary information about the facility to a facility manager. Such systems are called Computer-Aided Facility Management (CAFM) systems. Common features of CAFM systems include information and description on the physical properties as well as documenting of energy usage [4]. These systems and tools support the building management system, which mostly operates with objective measurements originating from sensors of the BAS [5,6]. Other input data originate from the feedback of tenants in form of problem reports to the facility management. However, allowing the occupants to directly adjust some environmental parameters is associated with risks [7]. For this, platforms, ticket systems or hotlines provide the means for users to give feedback in natural speech, through Web forms, or free form text fields. Analyzing such feedback is labor-intensive and the processing time is usually too long, hence this requires a more sophisticated interaction device to usefully provide feedback. By using a knowledge base specifying the requirements on the relevant information of different types of user feedback, knowledge engineering methods allow us to assess the data sets and classify problems and possible solutions [8]. In order to accomplish this, building operators and facility managers formalize their expert knowledge for deriving reusable, generic rules. In combination with monitoring data from sensors of the BAS and context information, triggering rules based on the user's feedback enables an indirect control of the BAS for increasing user comfort. Thus, research into closing this gap and providing a suitable set of rules is necessary. This further requires an appropriate feedback tool and methods to enable processing and comparison of subjective user feedback and objective monitoring data as far as possible in an automated fashion.

The paper contributes to answering the following research questions:

- Is there a uniform and straightforward method to incorporate occupant feedback into a building automation system automatically?
- What requirements for a user feedback tool have to be defined to enable gathering multi-dimensional feedback from occupants?
- What is a suitable architecture for allowing users to give context-aware feedback and enable automatic integration into a BAS?
- Which rules can be derived from occupants and facility managers in order to make suggestions to re-configure BAS and increase user comfort?

We are investigating an automatic integration of building occupant feedback and analyzing this feedback in combination with building monitoring data to improve occupant comfort and provide facility managers with information on possible re-configurations and ongoing building issues. As part of this process, a user feedback tool is designed and developed, an ontology for modeling all entities of a "Human perception and Building Automation System" (HumBAS) environment is created, and a rule-based approach for integrating feedback is developed [9]. This enables HumBAS to cover a wide range of use cases, with focus on office buildings, as they have a relatively constant and static user base, as well as to allow including open-plan offices into the project. Additionally, as work in office buildings is characterized by mental work with high demands on concentration, bad indoor conditions affect the performance of office workers [10]. In this respect, the office context provides a highly complex field worth investigating. Figure 1 provides an abstract overview of the HumBAS vision.

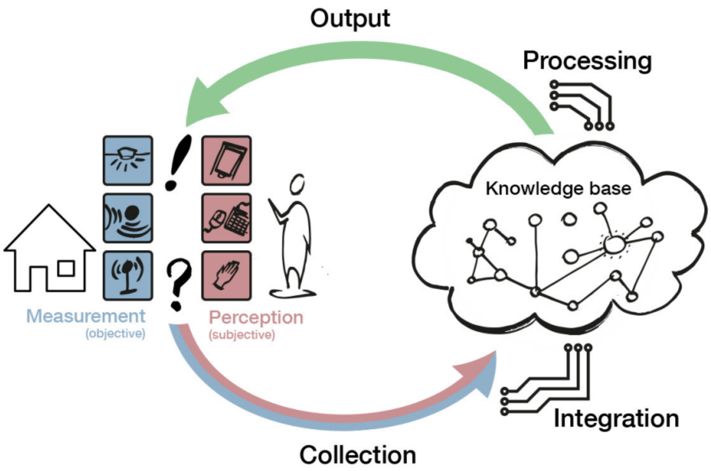

Figure 1. Human perception and Building Automation System (HumBAS) project.

The rest of this paper is structured as follows. Section 2 outlines relevant definitions used throughout the paper and provides the background on information models. Section 3 highlights relevant applications and research for capturing user feedback. Section 4 elaborates on the design approach of the feedback tool. The system architecture of Section 5 covers implementation details of the proposed approach. Afterward, Section 6 explains the deployment and evaluation strategy, while Section 7 concludes the paper and outlines the potential for future work.

2. Background

A short overview of relevant definitions and the background on information models is given, which provide the foundation for assumptions made within the rest of the paper.

2.1. User Comfort

An extensive literature survey by Frontczak and Wargocki [11] found four recurring categories across the surveyed studies: Thermal comfort, visual comfort, acoustic comfort and air quality.

- Thermal comfort is "that condition of mind which expresses satisfaction with the thermal environment" [12]. Apart from that, the standard ISO 7730 [13] defines two indices for predicting the mean thermal sensation (Predicted Mean Vote (PMV)) and the mean satisfaction with thermal conditions (Predicted Percentage Dissatisfied (PPD)) of a group of people.
- Visual comfort is defined as "a subjective condition of visual well-being induced by the visual environment" [14] and specifies a number of physical properties. These visual conditions include parameters such as illuminance, the color of light and the amount of daylight to name just a few [15].
- Navai and Veitch [16] defined acoustic comfort as "a state of contentment with acoustic conditions", whereas these conditions are mostly characterized by sound, mainly by sound pressure level and frequency. Positive acoustic comfort is primarily associated with avoiding acoustic discomfort [17].
- Similarly to acoustic comfort, indoor air quality is most often associated with the lack of discomfort as a result of odor and sensory irritation. Acceptable air quality is defined as "air in which there are no known contaminants at harmful concentrations as determined by cognizant authorities and with which a substantial majority (80% or more) of the people exposed do not express dissatisfaction" [18].

Some attempts have been made to rank the importance of the individual factors against each other, but unfortunately, as Frontczak and Wargocki [11] conclude, "little information is available on modeling how individual environmental conditions related to thermal, acoustic and visual comfort, as well as satisfaction with indoor air quality influence the overall satisfaction [with the indoor environmental quality]".

2.2. Information Models

With regards to modeling a building during its life cycle, Building Information Model (BIM) offers a method for providing a data model containing all data that emerge during planning, construction, operation and maintenance of a building. This data model is used and updated by all project participants (e.g., architects, planners, engineers, facility managers) during the whole building life cycle [3,19]. Corresponding to the term Building Information Model (BIM), the data model is known as a building information model. An important part of BIM is the building object that can be parametrized [19]. On one hand, there are component objects representing products with fixed shapes such as windows, doors, or boilers, on the other hand, there are layered objects such as walls or a roof. These objects are used within the particular building information models. Two important open BIM data exchange standards are the Industry Foundation Classes (IFC) [20] and the "Green Building XML" (gbXML) [21]. In recent years, a lot of research has been conducted on using BIM in the context of the Semantic Web. As the open specification of the IFC is the de facto standard for open BIM data, most research and work are focused on transforming IFC represented as an EXPRESS schema to an OWL ontology resulting in the ifcOWL ontology. Results of this work on ifcOWL and the conversion can be found in [21] and [22]. Another approach for modeling assets in buildings is Brick, an OWL-based open-source standard [23]. Brick provides semantic descriptions of the building's assets, as well as their relationships, and models them as a directed, labeled graph. This building model is machine-readable, which is accomplished by using the Resource Description Framework (RDF) language, allowing us to query the triple-based model using the query language SPARQL.

3. State-of-the-Art

Apart from ensuring the energy- and space-efficient use of a building, another goal of the FM is the well-being of the building's tenants. This well-being relates to the "user experience", which describes the interaction of a user with a system in relation to the quality of this interaction [24]. In order to measure this user experience, feedback from tenants can be analyzed.

Various applications and software solutions provide capabilities for capturing user feedback—specifically in the context of personal comfort estimation, modeling and optimization of indoor environments—and can be classified into three broad categories (some of which inevitably overlap), based on the technical approach taken by the authors:

- Mobile applications running on smartphones
  - Comfy [25] aims to manage the workplace by providing users with an app for booking spaces, finding colleagues and allowing them to give feedback about the workplace.
  - TherMOOstat [26,27] uses thermal feedback from users to define whether they feel comfortable or uncomfortable in order to contribute to energy savings.
  - BEES [28] collects occupancy preferences in order to calculate an optimal temperature set point and additionally takes spatial information into account.
  - FORCES [29] is a comfort voting application allowing users to give thermal feedback based on their preference for decreasing energy costs and increasing occupant satisfaction.
  - Authors of [30] propose a model to describe the personal thermal complaint behavior for controlling the personal office environment by setting thermal set points.
  - Lam et al. [31] implemented a participatory model-driven temperature set point controller that uses a mobile application for collecting votes from building occupants.
- Tangible interaction devices
  - MiniOrb [32] is a personal interaction device that documents user's comfort preference values, displays the local user group's sensor readings and the average comfort preferences of all users.
  - ThermoKiosk [33] provides users with an interaction device for expressing their thermal comfort and viewing the results for manually setting temperature values of Heating, Ventilation, and Air Conditioning (HVAC) systems.
- Wearable biosensors, often complemented by neighboring sensors
  - Comfstat [34] combines heart rate measurements from wearable biosensors, data from stationary sensors, and user-feedback collected from a mobile (smartphone) feedback tool.
  - Authors of [35] use commercial-off-the-shelf wearable fitness trackers and stationary sensors together with a mobile app for estimating individual mean thermal sensation model parameters.

Apart from these applications and tangible devices, the authors of [36] present the Web application "Yousense", which allows users to express their current feeling. Additionally, the application enables users to add contextual information, such as time and location.

Feedback recorded may not be connected to building automation-relevant comfort of users and is quite general (e.g., nice room). "Yousense" does not consider utilizing the building structure's data or feeding back control data into the BAS.

Similarly, some of the applications mentioned above have limitations regarding their comprehensiveness. Exemplary, "Comfy" only considers the BACnet protocol and does not include information from the model-based building management. Additionally, most of the applications mentioned above only include thermal information in their feedback process. The tangible devices are not connected to a BAS, hence an automatic integration of the user's feedback is not part of MiniOrb or ThermoKiosk. Using wearable biosensors for retrieving the user's comfort raises privacy concerns, as strict measures have to be taken in order to ensure the security of the user's data.

Authors of [37] present OFFICE, which aims to manage the trade-off between occupant comfort and energy costs within a smart building HVAC control by utilizing a Model Predictive Control (MPC) framework. By using the current and predicted location of users and what they want, weather data and how zones in the building react to changes, they aim to control the thermal zones. A Web application allows users to give thermal feedback ranging from "cold" to "hot" and includes a "neutral" vote, indicating the optimal condition for the user. With these features, OFFICE is able to minimize energy costs and still satisfy user comfort constraints. The OFFICE framework is one approach to utilize human-in-the-loop data for increasing comfort parameters by controlling the HVAC for thermal regulation. However, comfort does also include other parameters (see Section 2.1) apart from thermal perception. Additionally, it is unclear if and how the FM is informed about problems, for example, a defect ventilation leading to decreasing temperature within a zone. Such failures could lead to users giving incorrect feedback to the system. As given feedback is centrally stored in a database, privacy concerns regarding the user's thermal preferences arise. Overall, Winkler et al. [37] focus their work on the optimization of energy costs and thermal comfort and provide a foundation for further research.

In [38], the authors propose a framework for dealing with humans as an element of the building automation loop. By using a building information model as the core concept, they aim to allow users to give feedback by using a mobile application for enhancing and customizing operations of a building. The app should further allow observing the energy usage and occupancy level and view the need for maintenance processes. In [39], they use this framework on their campus for assessing a digital twin framework for controlling and monitoring sustainability assets. This educational building is equipped with a wide range of sensors and monitors students on daily activities, aiming to increase indoor environmental quality and energy utilization.

The study in [7] evaluated several field implementations of occupant-centric building controls. Their results revealed that most research focuses on HVAC and lighting sectors, separately, and that the duration was most often less than three months within 10 zones. With regards to user data and privacy concerns, which involve occupant's direct comfort voting as well as location privacy, the reviewed studies sometimes overlooked these aspects. Another interesting finding was that open-plan offices were rarely represented.

In recap, we assume our work is the first attempt in the literature focusing on office buildings to combine real-time occupancy, user feedback and sensor data with an information model approach and rule-based decision making for indirect BAS control, which includes thermal, visual, acoustic and air-quality parameters.

4. User-Centered Approach and Feedback Tool

In order to design a feedback tool that can collect feedback from users of an office building, a problem-oriented and user-centered approach [40] was applied. First, an in-depth analysis of the context was carried out. On the basis of these findings, personas were created with the help of which everyday actions in the work context should be better understood. This is achieved by playing through different narrative scenarios with them. Scenarios are used to refine the understanding of certain demands users place on a building.

Based on these scenarios, concrete use cases were derived that describe the interaction with and function of the system. In the following, this process is described in more detail and the results are presented.

4.1. Contextual Analysis

The contextual analysis contained a variety of methods to gain insights into the everyday use of buildings and BAS. In addition to a problem-oriented literature review, in-depth open interviews were conducted. Likewise, a series of smaller workshops with project members were carried out, in which the knowledge about, experiences with and demands on BAS were purposefully worked out. As pointed out by Frontczak and Wargocki [11], thermal comfort, visual comfort (natural and artificial lighting), acoustic comfort, and air quality are central dimensions that have a major impact on the perceived comfort in the interior. Based on this assumption, a mapping of the corresponding comfort parameters to commonly installed building automation subsystems that can influence them was made. The result is shown in Table 1.

Table 1. Mappings of comfort parameters to commonly installed BAS subsystems.

Comfort Parameter	BA Subsystem
Thermal	HVAC, shutters
Visual	Shutters, artificial lighting
Acoustic	HVAC, shutters, artificial lighting
Air	HVAC (ventilation), shutters

The in-depth interviews showed that the building users are mostly satisfied with the automated control of the BAS. Nevertheless, there are situations in which the BAS creates inappropriate conditions for certain work activities. For example, BAS controls the shutters to maximize natural lighting while not allowing direct sunlight to pass through the windows. This can lead to problems when presentations are held using video projectors if the lighting conditions in the corresponding room are too bright.

One of the most important conclusions from the in-depth interviews and the literature review is that the optimal work environment is the one that disturbs or interrupts least (see, e.g., [41] for more information about the effects of interruptions at work). Conversely, this means that the primary goal of HumBAS is not to obtain as much feedback as possible from users but to understand the act of giving feedback as a signal that something went wrong throughout the automation process. Feedback to the system is a means to an end in order to create a comfortable working environment that, optimally, automatically adapts to (potentially changing) user needs. In this respect, it is feasible not only to temporarily react to user feedback but to use this feedback to optimize the behavior of the BAS if it is facing similar future situations.

4.2. Personas

One major goal in human centered software design is to shift the view to the humans that are actually using the system that is to be designed. One tool to analyze goals and needs of the targeted user group is to create fictional characters that represent hypothetical archetypes of real users interacting with the system. Such fictional characters are called personas. Personas are usually described by a set of characteristics, goals as well as biographical and behavioral traits. Once carefully created, they serve as a vehicle to

- Understand the wishes, ideas, and goals of users.
- Identify occurring problems in the course of system interactions.
- Enable inter-subjective communication about design decisions within project teams, based on a fictional manifestation of user requirements [42].

Two personas were developed for the requirements analysis of the HumBAS feedback system. Firstly, a primary persona named Iris, which is characterized by high demands of a non-intrusive working environment. Furthermore, a secondary persona named Christoph was designed, which is primarily characterized by a focus on a trouble-free social environment and has a high level of technical understanding.

4.3. Scenarios

The personas served as basis to develop and play through several scenarios. Two different types of scenarios were used. First, a series of so-called "As-Is" scenarios that, based on the insights gained from the previous steps, address the status quo of current building use and the problems that occur in this context. Based on the "As-Is" scenarios, corresponding "Should-Be" scenarios were developed. These scenarios serve to describe the use of the system to be developed and how it is able to solve the problems that arise in the "As-Is" scenarios. This method can be used to develop concrete use cases based on a problem-oriented and user-centered approach. It should be noted that the used form of scenarios is less intended towards defining concrete interaction mechanisms, but towards the development of a conceptual design (realized through conceptual scenarios). Therefore, the interactions themselves and the technologies used to realize them are only vaguely suggested [40]. An excerpt of the developed scenarios for each persona can be seen in Table 2.

Table 2. Excerpt of scenarios for Iris and Christoph.

As-Is	Should-Be
It's 12:45 when Iris returns to her work space. Some sunbeams shine through the blending system installed on the building's facade and hit the wall opposite to Bernhard as well as Iris' face. She says: "They should configure those blinds correctly again!" Since the sun doesn't hit Bernhard nor its computer monitor, he replies: "but the light is much more comfortable." Iris grabs her notebook and sits down at the short end of the table which is fully shaded: "I cannot concentrate that way!"	After half an hour, Iris returns visibly angry to the office and tells Bernhard about her argument with her supervisor. She sits down in her office chair and realizes that the sun is shining directly in her face. "Can we close the shutter a bit? The sun shines directly in my face!" "Go ahead!", he replies. Iris opens the shutter interface on the HumBAS system and closes the shutters. The system incorporates her actions to her personal comfort model and will automate these manual actions in similar future scenarios.
Upon arriving at the office, Christoph immediately heads for a free desk in the open-plan area. "It's way too hot in here! Let's do everyone a favor and up the air conditioning!", he thinks to himself and heads straight for the wall-mounted thermostat. However, before he has a chance to even touch it, several voices can be heard in the background: "Don't touch it!", "Leave it alone!", "Go ahead! Turn it down a notch!", "No! We've just had this discussion an hour ago!". Surprised, Christoph turns around and heads for his desk. "Ok, suit yourselves! I'm only here for a couple of hours, anyways.", he says.	Upon arriving at the open-plan office, Christoph immediately heads for a free desk. "It's way too hot in here! Why don't they turn up the air conditioning?", he thinks to himself as he sends a notice to HumBAS that he's feeling uncomfortably hot. Upon receiving Christoph's feedback, HumBAS recognizes that it needs more information to decide. Up to this point, the system's assumption was that the current settings were OK for the majority of the people in the office. HumBAS selects a small number of people in the open-plan area (based on multiple criteria, e.g., whether they seem busy or not) and polls them for feedback. As the replies come in, most of them confirm that it is currently too hot. HumBAS reacts and lowers the thermostat set point.

4.4. Use Cases

Considering the findings of the previous steps, a series of use cases were defined. For simplification, a dummy system called "HumBAS system" is introduced, which is responsible for the processing and reasoning of the objective sensor data and the subjective user data generated by feedback. It thus serves as a bridge between continuously collected data in predefined areas of the building (zones) and the BAS, which controls the subsystems of the building according to a specific configuration. Since both the HumBAS system and the BAS are autonomously acting systems, they will be regarded as actors in the use cases. The use case descriptions are primarily written from the perspective of a currently active user, while all other building users are seen as passive users. The active user is regarded as the one that—whenever required by a use case—provides active feedback to the system.

Active here means that the given feedback is not the result of a request by the system but emerges from the user's needs.

Exemplary, some use cases are:

- Give sensual feedback and trigger actions.
- The HumBAS system is able to request feedback from its users.
- Every user has a personal comfort model with respect to privacy.
- Incorporate sensor data into the configuration sent to the BAS by the HumBAS system.
- Conflict resolution in case multiple users are in one open-plan office.

Some considerations have been made regarding users in an open-plan office. Above all, HumBAS treats every user equally so that no user roles (e.g., primary feedback provider) need to be defined. Additionally, the system assumes that users do not act maliciously in a way that bad indoor conditions are disclaimed to irritate occupants of the same room. Nevertheless, conflicting requests from different users may exist, since each of them reports their personal dissatisfaction. However, in case of repeated conflicting requests from different users the conflict resolution should not be solved by the HumBAS system. Instead, manual intervention from a corresponding service unit of the facility management seems necessary.

At present, communication between building users and facility management takes place via conventional communication channels such as telephone, e-mail, ticket-systems or face-to-face. Regarding the feedback channel between these two parties, communication is usually limited to reporting problems of the infrastructure. Therefore, the HumBAS system includes use cases for the FM workflow as well:

- Monitoring of data and system.
- Intervene into the feedback process, e.g., if no infrastructure is available to change the environmental conditions.

4.5. Feedback Tool Requirements

In order to design the feedback tool, several requirements have to be defined. In particular, the system should be able to capture (negative and positive) feedback from an office user and include spatial and temporal information. By using an automatic localization mechanism, feedback given shall affect environmental conditions in the spatial context where it was given, and the system has to provide feedback (feedback response). The system has to be portable and connected to the HumBAS system, while providing secured communication. Any collected data shall not be disclosed to third parties, data have to be pseudonymized and only the absolute minimum of data shall be collected. Users should be able to give feedback regarding temperature, ambient noise, ambient light, air quality and general comfort. The HumBAS system is able to request feedback and the feedback tool should be able to display this feedback request in a non-intrusive way, while there is no obligation to follow this request. The next section includes technical details regarding the feedback tool and covers its design approach.

5. System Architecture

The HumBAS system architecture comprises several elements (cf. Figure 2). Apart from the feedback tool, a WSN provides up-to-date environmental parameters and enables the feedback tool's localization. A central server, called HumBAS server, aggregates available sensor data from the WSN and the BAS and handles feedback from feedback tools by propagating it to the built-in rule-engine. As a communication protocol between these entities, MQTT was chosen. MQTT is a simple and lightweight communication protocol which is mainly used for machine-to-machine (M2M) communication and designed to fit the requirements of the Internet of Things (IoT) [43]. The protocol relies on a publisher/-subscriber model and guarantees the transmission of messages from clients to a server over the Transmission Control Protocol (TCP). There are two types of devices in a typical MQTT network, multiple MQTT clients, which exchange application messages, and an MQTT broker. A client can either be publisher, which announces messages, a subscriber, that applies for messages, or both. The MQTT broker interconnects all clients and accepts or transmits application messages which are published on predefined topics to all connected clients.

For an overall representation of the HumBAS environment, an information model of the building's environment defines all relevant entities. As configuring these various entities is prone to errors, the information model does include the necessary elements for defining configuration parameters, allowing for a very high configurability. The next sections provide details of these elements.

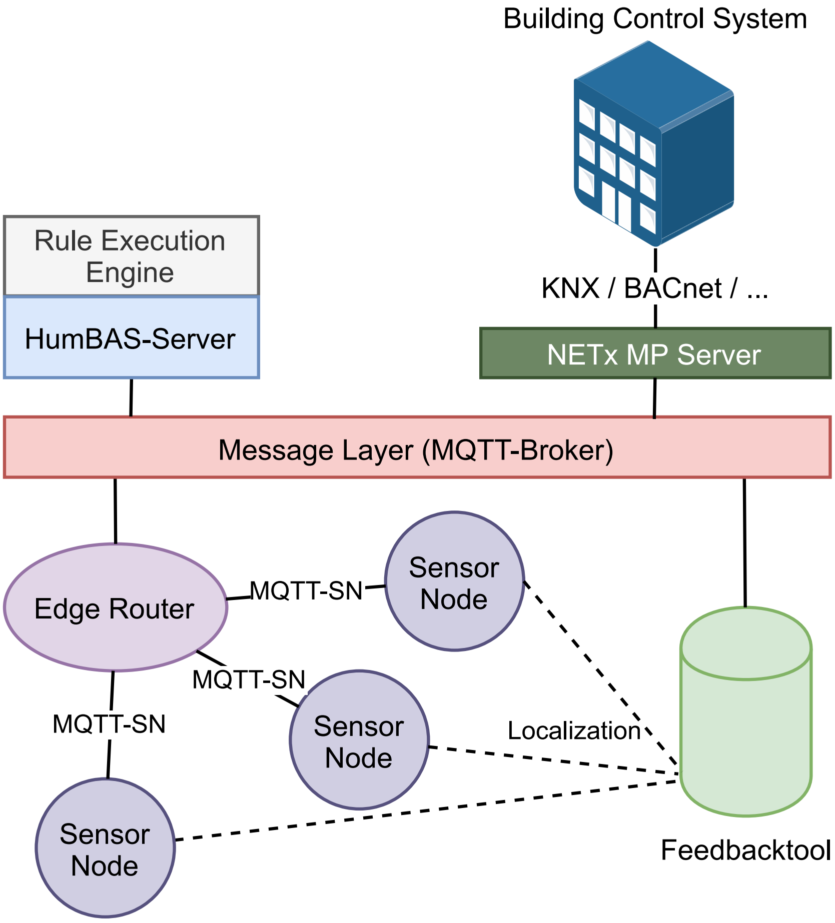

Figure 2. Overview of the HumBAS system architecture.

5.1. Information Model

In order to provide information about a building's equipment and components, a suitable approach is to use ontologies. Ontologies describe the knowledge of a specific area of expertise with the assistance of formal ordered conceptualities and their relations to each other [44]. They are mainly used to share the knowledge between different applications and services, plus they represent knowledge and provide means to discover new relationships. Most common components of ontologies are individuals, classes and relations [45–47].

The individuals, sometimes also called objects, are the basic components of ontologies and include real objects such as a developer (person), a climate system or an office room. Classes or concepts are collections of objects with similar characteristics, comparable to classes in any object-oriented programming language. A class might be a facility system, which contains all components such as heating, climate systems and entrance systems for buildings. Another example for a class is a room; individuals of this class are, for instance, an office room, a conference room and a dressing room. The last component to describe an ontology are relations. Relations are used to link objects and define the correlation between them and also to outline properties.

As HumBAS requires an information model which not only stores information about the building, the BAS, measurement data as well as user comfort requirements, the underlying ontology should support the necessary features. Overall, after considering various ontologies, the main aspects required of the HumBAS ontology are included in the Brick ontology [23], and therefore, Brick provides the foundational ontology for the HumBAS information model.

As a first step, the sensor nodes which compose the WSN, consisting of sensor nodes and edge routers, were modeled according to Web Ontology Language (OWL) and Brick. After integrating these nodes and at the same time creating the necessary additions to Brick (components for measurement data), the other necessary components were modeled. These are:

- The feedback tool;
- The HumBAS server;
- The building automation system.

As Brick does not intend to model software components, the necessary OWL definitions for the feedback tool software and the HumBAS server were integrated into the HumBAS information model. Alongside this, the first version of the software prototype was designed, which included the WSN, the feedback tool, the HumBAS server and the data exchange protocol. With regards to these components, all core entities of the HumBAS environment, namely the WSN, the feedback tool, the HumBAS server, the BAS integration and communication aspects, have overlapping configuration parameters, such as unique identifiers or network addresses. Resulting from this complexity, configuring and deploying all components correctly is an error-prone process, as small mistakes or typos could potentially cause the system to not work properly. This led to a very important addition to the information model: the modeling of configuration parameters and the generation of configuration files based on the information model.

With this, it was necessary to adapt the HumBAS information model with the addition of the following components:

- An entity which represents the configuration of various entities;
- Configuration items representing configuration parameters;
- Data exchange platform-related software components;
- The correlation between the data exchange platform and the BAS.

Additionally, each of these components requires storing corresponding parameters as text or numbers in the model and allowing for extracting them as needed. This was achieved by extending the model with configuration-specific entities that represent the configuration parameter and a relation to their values (cf. Figure 3).

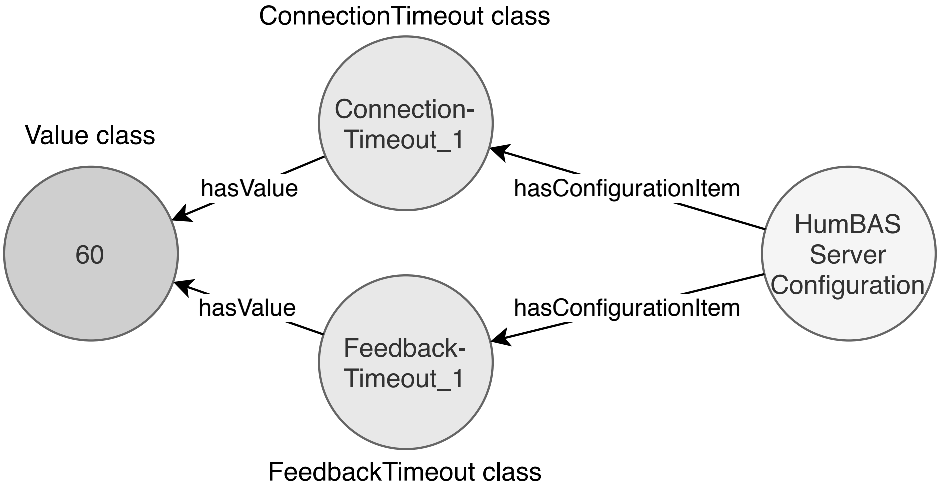

Figure 3. Value mapping.

Figure 4 illustrates the different files generated from the information model specific to a HumBAS-enabled building. Apart from the generated configuration file, which defines relevant configuration parameters of the respective entities, a few entities require more specific configuration or information. For defining the HumBAS environment, an environmental file describes all entities, which includes locations and zones as well as Boolean flags for indicating possible set points. In order to set and read any data point of the BAS, a mapping file defines the relation between each of those points and the respective interface definition. Additionally, sensor data received from the WSN requires a data conversion, as described in Section 5.3, which presumes information of the information model. Therefore, a code generator queries the relevant information of the information model and generates an executable Python program, which handles this data conversion.

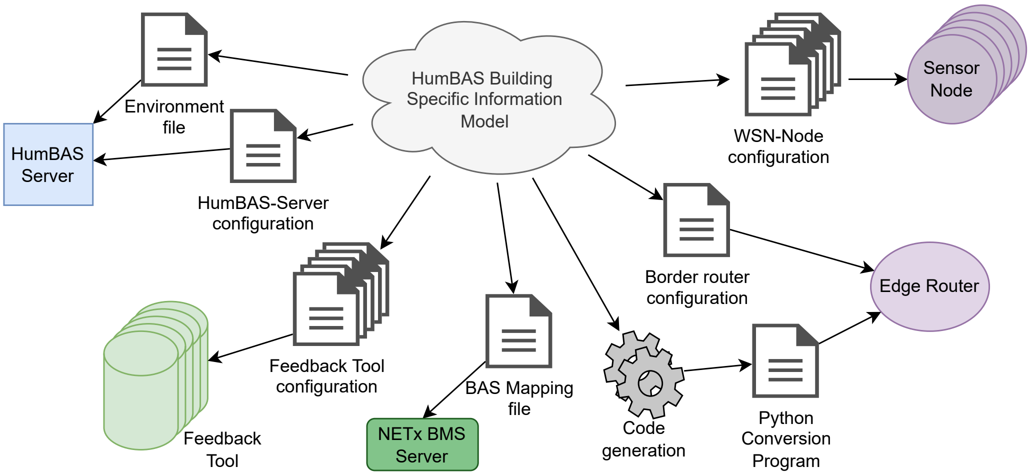
HumBAS Building Specific Information Model
Environment file
HumBAS Server
HumBAS-Server configuration
WSN-Node configuration
Sensor Node
Border router configuration
Edge Router
Feedback Tool configuration
Feedback Tool
BAS Mapping file
NETx BMS Server
Code generation
Python Conversion Program

Figure 4. Files generated based on the information model.

5.2. Feedback Tool

Based on the requirements of the feedback tool (see Section 4.5) and use cases (see Section 4.4), internal workshops were held, in the course of which a variety of ideas for interaction forms were generated. Four resulting ideas were then considered in detail and for each idea a paper prototype was realized. At the same time, interaction icons were created to symbolically represent the interaction language without resorting to the written language. After conducting interviews, the evaluation showed that an interaction language that allows users to articulate their own perception of a comfort parameter (e.g., "I am too cold") is most suitable for the HumBAS Feedback Tool. With regards to the form factor of the feedback tool, it appeared that subjects were virtually unanimous in finding a cylindrical tangible interaction tool with a rotating input mechanism the most interesting for interacting with the system.

This personal cylindrical tangible should provide the possibility to articulate the current perception of the relevant indoor climate parameters. Based on this, a series of sketches were created in which the initially rough design of the cylinder was refined. In addition, possible interaction flows for the core functionality of giving feedback as well as receiving and responding to feedback requests were conceived with the help of storyboards. Further, all interaction symbols that were declared difficult to understand were revised.

In addition to the sketches, storyboards, mock ups, and non-functional paper prototypes, the technological feasibility and the required hardware became a major focus of the design process. Based on the designed interaction flows, hardware requirements were derived, with the help of which the various interaction mechanisms can be realized. The next step was to design the functional HumBAS Feedback Tool under consideration of both aesthetic and functional aspects. The prototype was to be produced using 3D printing. In particular, the arrangement of the hardware inside the cylindrical base shape and the implementation of the rotating mechanism proved to be a challenge. In addition, the

HumBAS Feedback Tool software was developed to implement the designed input and output mechanisms and the communication with the overall system. The final design is illustrated in Figure 5.

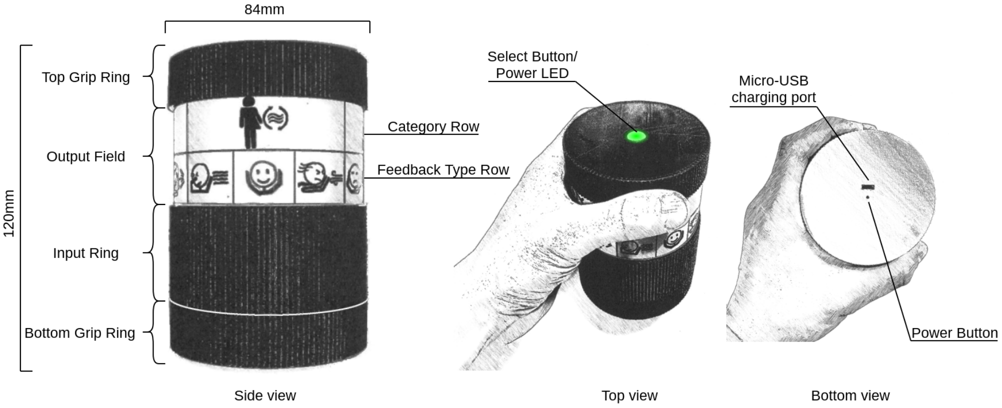

Figure 5. Description of the individual elements of the HumBAS Feedback Tool.

The core of the hardware is a Raspberry Pi Zero W, which due to its size is particularly well-suited. The Raspberry Pi's Wi-Fi module enables connectivity to the HumBAS server and the Bluetooth Low Energy (BLE) module is used for self-localization of the feedback tool. Other hardware components include a 4000 mAh LiPo Battery, a 5 V power booster for transforming 3.7 V to 5 V, a custom power on/off module, a rotary encoder, RGB LEDs and a compass and accelerometer module.

One of the core elements of the feedback tool is the input ring, whereas behind the feedback symbols are RGB LEDs indicating the position of the feedback the user will give. By turning this ring, the LEDs behind the feedback symbols rotate, which means that the current LED is turned off and the LED in turn direction will turn on. After pressing the button on top and putting the feedback tool back on a flat surface, the feedback is sent to the HumBAS server. For indicating that the user of the feedback tool is busy, turning the feedback tool upside down will switch it to the suspend mode, which means that any received request will not result in any visual indication by the feedback tool. With regards to a feedback request issued from the HumBAS server, all LEDs from the feedback tool will slowly fade in and out indicating this feedback request if not in the suspended mode.

The above-mentioned features of the feedback tool are encapsulated into one of two software components, namely the feedback tool software stack. The other software component of the feedback tool is the comfort classifier software stack.

In order to enable adapting the BAS in future situations when feedback was given, the comfort classifier incorporates given feedback by its user and uses the data to adapt a locally stored comfort model. This model consists of several smaller models differentiated by zone, hour and season. Each of these smaller models store four different environmental categories: thermal, acoustic, light and air quality. Exemplary, the thermal category stores the temperature, the humidity and the level of shutters if they are available. To achieve this, as soon as feedback is given, the current environmental parameters (e.g., temperature, brightness) are sent to the feedback tool and relevant values are integrated into a fitting model with regards to spatial and temporal context. Depending on the type of feedback, for example "too hot", the comfort classifier will adapt the corresponding model's category (i.e., thermal) accordingly. For this, each category's environmental parameter stores upper, lower and optimal values of acceptable, hence "OK", values as perceived by the feedback tool's user. Integrating new feedback therefore adapts these values, e.g., "too hot" feedback may decrease the upper limit of the temperature parameter. This enables the comfort classifier to create a model of the perceived user's comfort.

Other features include the parameterized request of this comfort model in order to adapt a zone to the necessities of the user without it performing and keeping track of statistical data. Additionally, the comfort classifier utilizes a keep-alive mechanism for informative purposes, a Structured Query Language (SQL) database for storing the comfort model and a local MQTT broker, which propagates location changes directly to the comfort classifier.

With regards to the used software stack, JAVA in combination with the Spring Framework [48] was used. The Spring Framework provides several features, such as dependency injection, data bindings, data access, in particular Java Database Connectivity (JDBC) for connecting to the local SQL database, and enables focusing on application level logic. Another important part of the implementation is the use of Maven [49], which is a software project management tool and allows for integrating various libraries automatically, such as the required MQTT client. For deploying the comfort classifier, the whole software stack is containerized into a Docker [50] image and pushed to a repository or saved locally.

Figure 6 shows the corresponding fully functional HumBAS Feedback Tool on the left and the software modules on the right.

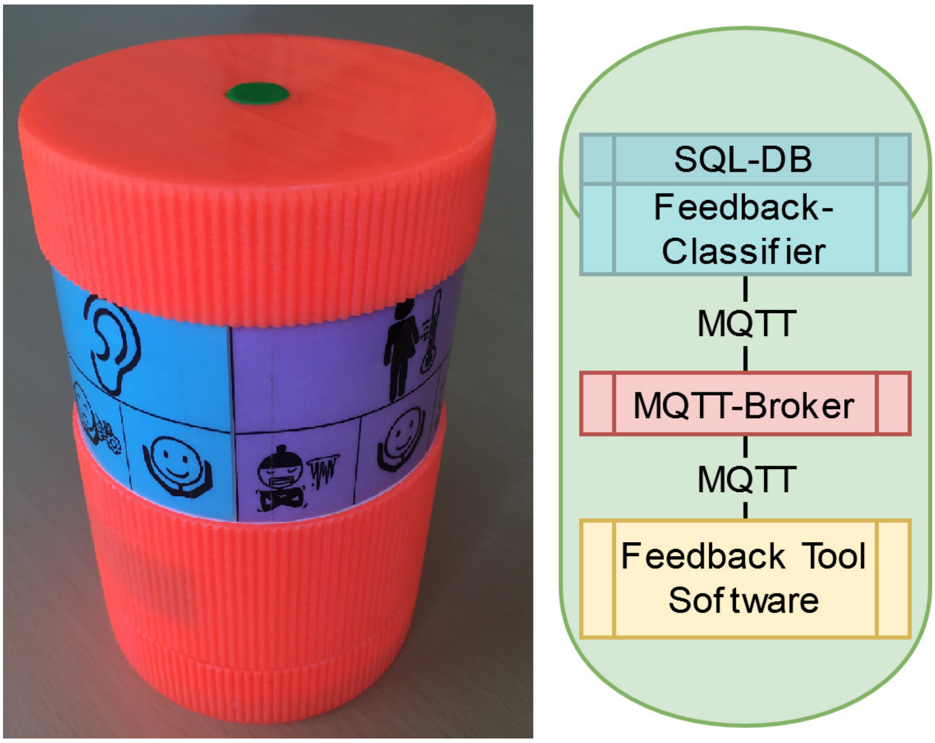

Figure 6. Prototype of the HumBAS Feedback Tool in cylindrical shape on the left and software modules on the right.

5.3. Wireless Sensor Network

The Wireless Sensor Network (WSN) is responsible for two tasks: first it delivers various sensor data to the HumBAS server; and second, it provides the localization for the feedback tool by utilizing a Bluetooth beacon. The WSN is split up into sensor nodes, which provide environmental data and the Bluetooth beacon, and the Border Router Implementation. The sensor nodes further are able to communicate with each other via OpenThread [51] and are connected to the Border Router, which acts as a gateway between the sensor nodes and the Internet. The communication protocol between these two elements is the Message Queuing Telemetry Transport for Sensor Networks (MQTT-SN), a lightweight form of MQTT, tailored for sensor networks which utilizes Transport Layer Security (TLS) encryption for security. Moreover, the Border Router provides an MQTT-SN Gateway which forwards the generated data to the application MQTT broker. Part of this data propagation involves a data conversion of forwarded messages. This is necessary as additional data, such as the sensor node's location, have to be included into the messages sent to the application MQTT broker.

With regards to the localization of the feedback tools, at least one sensor node per zone is necessary. The feedback tool uses the Received Signal Strength Indication (RSSI) for measuring the signal strength of nearby Bluetooth beacons provided by the sensor nodes. By this signal strength, a high enough precision can be achieved, so that the feedback tool can estimate its zone position inside a building.

5.4. Humbas Server

One central aspect of the HumBAS architecture is the HumBAS server. It has several features, whereas the main goal is managing all communication between users, the building automation system and the facility management. Additional features include monitoring of the location and status of every feedback tool, providing statistical data for the FM and validating feedback from users by propagating it to the built-in rule execution engine. In order to be aware of all features, locations and zones the building and the BAS provide, a so-called HumBAS environmental file is necessary. This file includes the spatial information, as well as flags which define if a manipulation of environmental parameters is possible, i.e., a flag for shutters if they can be automatically lowered and raised.

5.5. Connection to BAS

In order to send commands to the BAS, the NETx MP Server (www.netxautomation.com (accessed on 1 January 2022)) provides an interface for receiving MQTT messages encrypted via TLS. The HumBAS server accesses this interface for managing a virtual representation of the BAS in order to accurately calculate the difference in absolute and desired environmental parameters.

5.6. Implementation

With regards to the implementation of the system architecture, the entities of the HumBAS environment, namely, the HumBAS server, the rule execution engine, the WSN, the feedback tool and the NETx MP Server, utilize various software stacks.

Similar to the feedback tool's comfort classifier, the HumBAS server uses the Spring Framework [48] as foundation, Maven as software management tool [49] and Docker [50] as deployment tool. A configuration file generated by the information model includes parameters for timeouts and most importantly the definitions for the MQTT topics used throughout the HumBAS environment.

In order to evaluate feedback from users based on the rules (cf. Listing 1) provided by the FM, a rule execution engine is necessary. For this purpose, Drools [52] was chosen, an open source Business Rules Management System (BRMS). It features an inference-based rules engine for evaluating rules and has support for the Spring Framework, hence it integrates well into the HumBAS server architecture. By executing rules and receiving feedback, this rule engine may generate commands for the BAS, which the NETx MP Server receives. This server provides multiple protocol interfaces for connecting different technologies such as KNX, BACnet, Modbus and OPC allowing it to manage a range of building automation systems.

MQTT handles TLS encrypted data exchange between the entities of HumBAS by using different topics and JavaScript Object Notation (JSON)-encoded payloads containing feedback, sensor data or other information used by HumBAS.

Listing 1. Extract of rules for regulating temperature and lighting.

```
rule ''Feedback tooHot reduce room temperature''
when
$ruleData:RuleData(feedback == ''toohot'' &&
temperature > 23.0
);
then
$ruleData.setResult(''TEMP_REDUCE'');
end
rule ''Feedback toodark open shutters''
when
$ruleData:RuleData(feedback == ''toodark'' &&
shuttersClosedFlag == true
);
then
$ruleData.setResult(''SHUTTERS_OPEN'');
end
```

HumBAS provides several features and therefore specifies various communication flows between involved entities. The most relevant communication flow is between the feedback tool and the HumBAS server for users to give feedback. As depicted by Figure 7, after the feedback tool publishes the feedback on the corresponding topic, the MQTT broker will propagate this message to the subscribed HumBAS server. The built-in rule execution engine evaluates this feedback by using the most current environmental sensor values. In case the amount of feedback tools does not exceed the threshold for an open-plan office, i.e., less than two feedback tools are within the same zone, the rule execution engine may create a command for the BAS. Otherwise, the rule execution engine needs feedback from the other feedback tools within the zone as well.

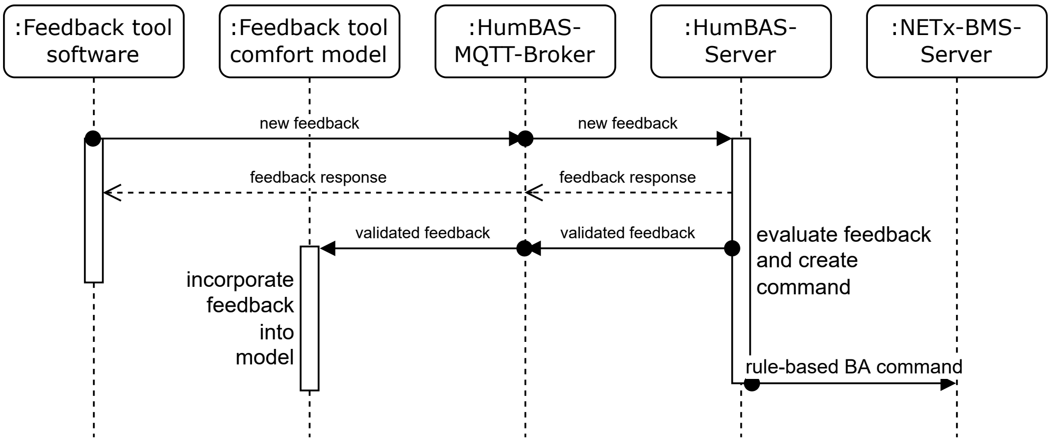

Figure 7. Sequence diagram of new feedback.

Illustrated by Figure 8, the HumBAS server will create a feedback request for every other available feedback tool. For this, it propagates the necessary information, i.e., the feedback *tooHot*, to an internal module called *feedback manager*. This threaded process handles the publishing of above-mentioned feedback requests and awaits the responses. After all requested feedback tools have responded or timeouts have been reached, the *feedback manager* returns the responses to the rule execution engine. The engine evaluates all responses, and if enough do match with the initiator feedback tool, this feedback tool receives a *valid* feedback response and a BAS command is sent to the NETx MP Server. Otherwise, if not enough feedback responses do match, no command is sent to the BAS and the initiator feedback tool receives an *invalid* feedback response.

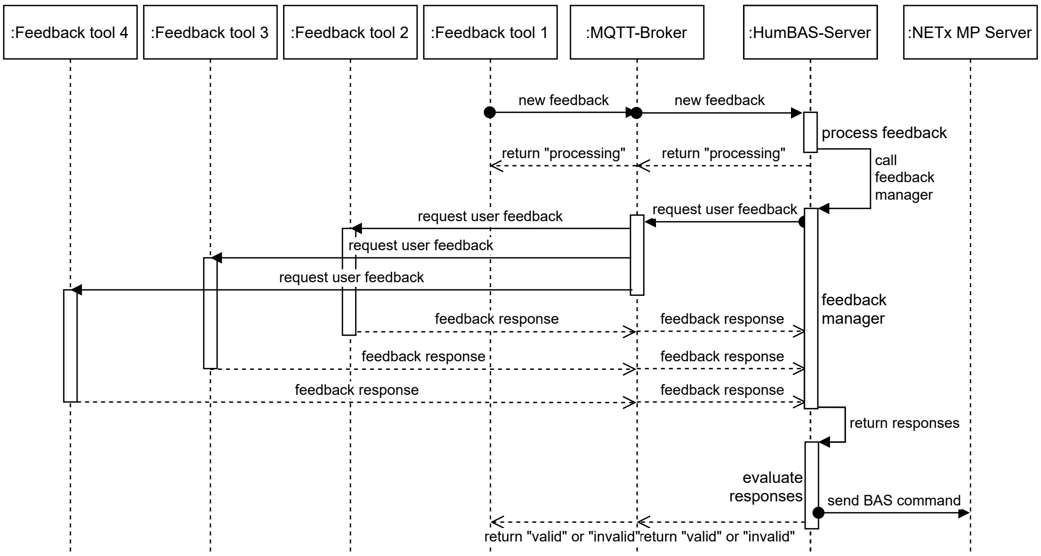

Figure 8. Sequence diagram of feedback for further processing.

Apart from these two use cases, other communication flows include requesting the comfort model of a feedback tool, publishing the latest WSN sensor data, requesting the current feedback tool statistics and updating its local stored location-based environmental values (temperature, brightness). For all changes issued by the rule engine, it is assumed that the BAS takes care of energy-efficient control strategies. In addition, HumBAS supports defining rules with priority, e.g., in a dark environment, the shutters will open first instead of turning on the lights, resulting in lower energy consumption. Besides, generated commands (e.g., changing setpoints) are only allowed in a well-defined range (determined by the facility management strategy which also has to follow legal regulations).

6. Deployment and Evaluation

In order to prove the feasibility of the project, a one-week evaluation process in an office building took place. This evaluation required access to the local BAS components, i.e., HVAC, and connectivity to the central HumBAS server for feedback evaluation. With this, users were able to give feedback via the feedback tool, and resulting in this, the rule engine can trigger events such as closing the shutters or dimming the light. Naturally, connectivity to the HumBAS Server was necessary, and as the server may not be located on site, Internet access has to be available.

However, as integrating all features of HumBAS in situ was not possible due to several constraints (i.e., integration into the existing environment for the duration of the evaluation, no Internet connection allowed), the deployment setup was converted to a mobile approach.

6.1. Mobile Deployment

The mobile deployment included two feedback tools, two WSN sensor nodes, one NVIDIA Jetson Nano (https://developer.nvidia.com/embedded/jetson-nano-developer-kit (accessed on 1 January 2022)) and one wireless router. Figure 9 depicts the components and their interconnection.

The Jetson Nano hosts the HumBAS server, an MQTT broker and the edge router for connecting the sensor nodes. Connected via the wireless router, the feedback tools publish and receive messages by establishing a communication link to the MQTT broker.

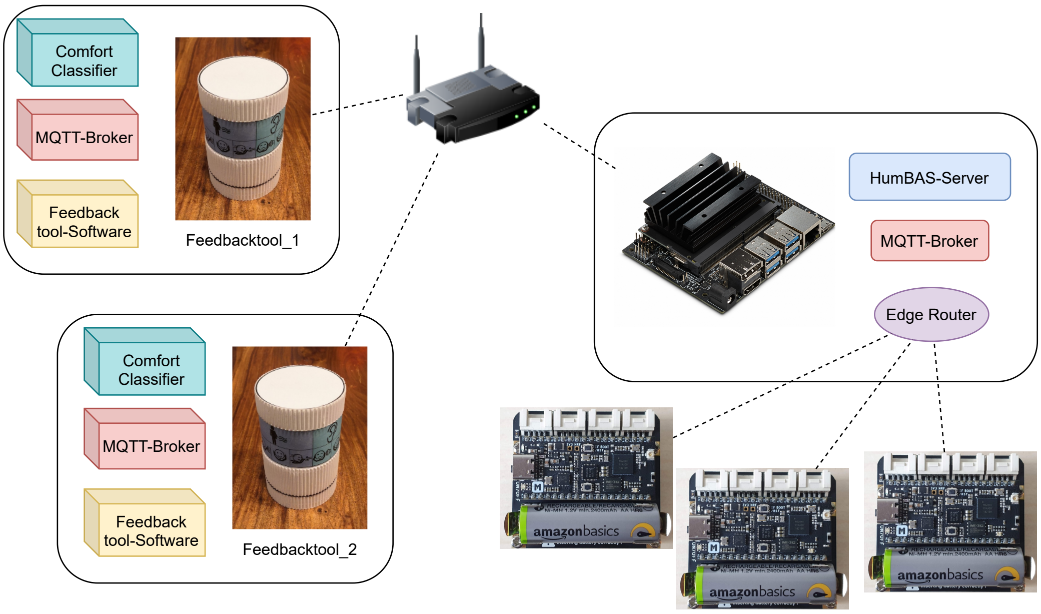

Figure 9. Components and their interconnection of the mobile deployment approach.

With this mobile approach, the evaluation process could be carried out regardless of most limiting factors.

6.2. Field Study

One central goal of the field study was to integrate the feedback tool, including the back-end system, into the test persons' working environment. The aim was to create settings as close to reality as possible, with the most significant trade-off being the effects of the feedback on the working environment. However, as a connection to the building's BAS was not possible, feedback provided by the users did not affect indoor conditions. Nevertheless, important information about the design of the interaction flows, the systems' responses, and the general configuration could be obtained through these evaluations. The central questions of this evaluation covered if the designed user-interface sufficiently communicated the intended qualities and states of the system and if the proposed system met the user's requirements. Additionally, one question deals with how the overall system should be configured so that it provides the intended functionality but at the same time does not interfere too much with the user's work rhythm.

In order to analyze this one-week field study, the feedback tools were configured to log any input and the test users received various tasks during the week. The users received manuals to illustrate and guide the evaluation process and a super user had the role to initiate an evaluation process on a daily basis. Exemplary, on the second day, the user had to trigger a feedback request for all feedback tools twice a day and log the corresponding timestamp. On the last day, the user was asked to trigger such a feedback request six times a day.

With regards to central user tasks for each day, the users should take the feedback everywhere they go within the office and should provide feedback whenever they perceive the current environmental condition as pleasant, unpleasant or they would like to change them somehow. On the first day of the study, the users did not receive any instructions on how to use the feedback tool and received the task to log any unexpected things that happened and what they tried to accomplish. At the end of day one, the users received a detailed manual of the system, including descriptions of the feedback tool's symbols. For day two and three, the users should use the system as they want and log any notable thing in the course of interaction. On the last two days, the users should not log the interaction. At the end of each day, every user had to additionally complete a survey which summarized their interaction experience, their emotional connotation, any remarkable experience as well as additional notes, such as unclear behavior or general feedback.

6.3. Results

The functionality and interaction design of the HumBAS Feedback Tool were satisfactory for the end user. The use was easy from the beginning of the evaluation process and clear in most cases. Sometimes, there were ambiguous reactions of the tool, at least, it was not clear to the user why the reaction of the system was in a certain way. Learnability and adaptability were well-provided by the tool. The main positive effect is that the tool helps increase people's awareness of the room and air situation in an office environment and provides new and innovative ways of dealing with it, especially from the users' point of view. We know that the evaluation was limited due to the COVID-19 situation and restrictions in this period, but it can be seen as a qualitative one to consider for the further improvement and development of the HumBAS Feedback Tool.

Some feedback included why the feedback tool behaved in a certain way, or why the color and duration of the LEDs was sometimes different or the missing interpretation of the feedback request as the feedback tool lights up completely. At the end of the field study, additional feedback stated that they felt confident using the feedback tool and would intend to use it in the future. Overall, feedback was provided 111 times; most notably, the feedbacks *OkLight* and *OkAcoustic* were provided (see Table 3).

Table 3. Amount of feedback given during evaluation period.

OkTemp	OkLight	OkAir	OkAcoustic	TooHot	TooCold	TooBright	TooDark	TooLoud	StuffyAir
15	20	16	19	2	15	6	3	10	6

Based on the feedback given and the environmental data received from the WSN, retrieved log files of the HumBAS server showed that the rule engine was triggered 23 times and generated control commands. Mostly, it processed feedback of the type *TooCold* and *StuffyAir*, whereas the later triggered an increase in ventilation. Roughly half of the time, *TooCold* triggered an increase in room temperature and the other half it started the process to inform the facility manager.

7. Discussion and Conclusions

This work proposes HumBAS, a system to use user feedback to enable rule-based control of the BAS for increasing the tenant's comfort. A feedback tool alongside a WSN allows for privacy-conscious tracking of the tenant's comfort requirements and their real-time location inside the building. The underlying information model defines relevant building and software components as well as their configuration. This enables the FM to keep an overview of the building's current state and eases problem finding.

To answer the first research question of this paper, "*Is there a uniform and straightforward method to incorporate occupant feedback into a building automation system automatically?*", the feedback tool provides a simple, yet expressive and clear way for users to state their current comfort perception. By utilizing a rule-based approach, the structured feedback is translated into commands for the BAS, enabling control of the building's equipment. Similarly to applications such as "Yousense" [36], the feedback tool allows users to state their current environmental perception. However, our feedback tool strongly focuses on occupant comfort in relation to the BAS, and therefore, limits the feedback possibilities, e.g., "too hot", while still providing enough expressiveness for users to state their comfort perception. This ensures that given feedback is intended to adapt the building's environmental conditions and therefore aims at increasing the users' comfort. Additionally, in cases which cannot be covered by HumBAS (for instance, when user feedback concerns the brightness even though the light cannot be controlled due to missing automation or broken equipment), the FM receives a notification, circumventing the need for the user to create a ticket. In situations where feedback is given in the open-plan office context, changes are made only by general consensus, i.e., most available feedback tools accept new room conditions. Future work could analyze whether and how much influence such an approach has on energy consumption—especially compared to the possibility that users can change the office environment settings manually, although not everyone agrees and may undo them again.

Proposed frameworks such as [38,39], which aim to strongly integrate the users into the building's operation processes, require a well-equipped building, which in their case demonstrates an advanced concept of future buildings. They focus on the energy efficiency of the building, by optimizing the daily HVAC schedules based on the feedback given by university users, meaning the user base consists of well-informed (technical) users, which is not the case for a typical office building. Additionally, the feedback given is limited by temperature, humidity and CO_2 pressure for calculating the indoor environmental quality. Similar to work conducted by [37], their focus also depends on thermal parameters, neglecting other comfort parameters. Moreover, it is not clear how, where and if the user data are stored, which may cause privacy issues when used outside of academia, which is supported by the study of [7]. By storing the comfort model of its user locally on the feedback tool, our approach does not store any personal comfort data on a central server.

These considerations are also covered by the second research question, "*What requirements for a user feedback tool have to be defined to enable gathering multi-dimensional feedback from occupants?*" Apart from above-mentioned privacy concerns, the feedback tool has to be able to allow the user to express their comfort perception based on four different parameters [11]. By using a user-centered approach, the requirements and use cases for deploying a feedback tool in an office building were defined, allowing us to cover several use cases which can occur in such a setup. In contrast to smartphone applications such as [25–31], our approach allows users to give feedback on a non-intrusive hardware-based feedback tool. This enables users to quickly react without the need to consult their smartphone and no personal user data are required to use the feedback tool. Further, by enabling users to give feedback in an open-plan office, but still allowing them to ignore generated feedback requests, this work contributes to the small field of open-plan office user comfort research [7]. Other hardware-based devices, such as [32,33] either only display the current comfort perception of their users, or focus on thermal parameters and do not have automatic integration to the BAS. Work conducted by [34,35], which uses wearable devices to track users' temperature and heart-rate data, focuses purely on thermal comfort and additionally requires an application on their smartphone for retrieving the data from the wearable sensors. Privacy concerns are either not covered or rely on the stored data on a local server, which again is supported by the findings of [7]. Both utilize smartphone applications to retrieve the user's comfort perception based on the PMV scale [13].

The presented implementation of the HumBAS environment allows us to discuss the third research question, "*What is a suitable architecture for allowing users to give context-aware feedback and enable automatic integration into a BAS?*" The proposed architecture, consisting of the feedback tool, the WSN, the HumBAS server, the BAS and the MQTT middleware, enable gathering and processing of the user's comfort preferences. Due to the automatic generation of configuration and mappings, the high configurability allows for a wide range of application scenarios. By combining localization techniques with objective sensor measurements sent from the HumBAS server, a user is able to give context-aware feedback, which is further integrated into the local comfort model. This architecture allows for combining this structured feedback automatically, by utilizing the pre-defined rule sets and mappings to send control commands to the BAS. In contrast to [37], HumBAS does not predict data based on historical occupancy data or use data-driven thermodynamic building models to regulate thermal conditions, but creates a location-based, personalized comfort model for increasing overall comfort quality. Similarly, energy consumption is currently not considered in HumBAS architecture, as it is assumed that BAS settings keep energy usage between acceptable levels. A few limitations regarding the current implementation concern the scaling for a broader use case. Further research could concentrate on how to integrate, implement and maintain a distributed approach to exchange the central HumBAS server, which our architecture supports as it uses the flexible MQTT middleware. This distributed approach would allow for management of multiple buildings, e.g., a campus, where the feedback tool can be carried between those. Another research direction could be to analyze other forms of feedback tools for different use cases and scenarios, e.g., for flexible work spaces where thermal regulation can be neglected but lighting conditions could be flexible and adapted based on the user′s preferences. Future hardware for the feedback tool, such as the recently released Raspberry Pi Zero 2 W, would allow for the implementation of additional features, such as more complex prediction models or potentially could enable local machine-learning algorithms. The proof of concept implementation of the HumBAS architecture was further supported by the one-week evaluation study, which also showed the potential of HumBAS. Even with the limited amount of time and the restricted test scenario, valuable feedback could be gathered and used for improvements on the prototype system. Future work will be built upon that feedback for enhancing the user experience. Naturally, a long-term field study with connection to the BAS would allow us to gain new and additional insights for further improvements. Similarly, future work could look at integrating HumBAS with a Building Energy Management System (BMES) to derive the energy consumption caused by the changes made by the rule-based engine. Since no visual representation of data is currently available in the prototype implementation, apart from displaying debugging information, another focus of future work could use the developed data model and the MQTT transport mechanism to provide appropriate features for displaying trends or for reasoning on statistical information.

With regards to the last research question, "*Which rules can be derived from occupants and facility managers in order to make suggestions to re-configure BAS and increase user comfort?*", the one-week evaluation indicated that even with the limited amount of data available, the rule-engine triggered events accordingly to the expectations. In half of the events, the FM was informed, which was to be expected. However, in this event HumBAS does support the FM, as it propagates potential problems without the need for the users to inform the FM through other channels. The rules derived based on the expert knowledge of facility managers further created control commands for the BAS for increasing comfort, whose efficacy has to be proven in another field study with access to a BAS.

In conclusion, HumBAS demonstrates how to integrate and merge subjective human perception and objective data from BAS to increase comfort for building occupants and derive suggestions for facility managers.

Author Contributions: Conceptualization, D.R. and M.D.; Investigation, D.R. and M.D.; Methodology, H.T. and W.K.; Project administration, W.K.; Supervision, H.T. and W.K.; Validation, H.T. and W.K.; Writing—original draft, D.R. and M.D.; Writing—review and editing, H.T. and W.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Austrian Research Promotion Agency (FFG) grant number 867681.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: Not applicable.

Acknowledgments: Open Access Funding by TU Wien.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

BAS	Building Automation System
BIM	Building Information Model

BMS	Building Management System
POE	Post Occupancy Evaluation
HVAC	Heating, Ventilation, and Air Conditioning
UFO	Unified Foundational Ontology
RDF	Resource Description Framework
W3C	World Wide Web Consortium
XML	Extensible Markup Language
OWL	Web Ontology Language
IoT	Internet of Things
WSN	Wireless Sensor Network
RTD	resistance temperature detectors
BLE	Bluetooth Low Energy
MQTT	Message Queuing Telemetry Transport
MQTT-SN	Message Queuing Telemetry Transport for Sensor Networks
M2M	machine-to-machine
QoS	Quality of Service
UTF-8	Unicode Transformation Format - 8-bit
OT	Open Thread
6LoWPAN	IPv6 over low-power wireless personal area network
PAN	personal area network
RLOC	Routing Locator
ALOC	Anycast Locator
OSI model	Open Systems Interconnect model
IPv6	Internet Protocol version 6
UDP	User Datagram Protocol
TCP	Transmission Control Protocol
FTD	Full Thread Device
MTD	Minimal Thread Device
REED	Router Eligible End Device
FED	Full End Device
MED	Minimal End Device
SED	Sleepy End Device
SOC	System on Chip
MDK	Micro Development Kit
TLS	Transport Layer Security
VOC	volatile organic compounds
TWI	Tow-Wire Interface
ADC	Analog to Digital-Converter
OCD	On-Chip Debugger
CLI	Command Line Interface
API	Application Programming Interface
UUID	Universally Unique Identifier
NCP	Network Co-Processors
IEEE 802	IEEE 802.15.4
SDK	Software Development Kit
USART	universal synchronous and asynchronous receiver-transmitter
FM	Facility Management
IFMA	International Facility Management Association
ICT	Information and Communication Technology
CAFM	Computer-Aided Facility Management
IFC	Industry Foundation Classes
RSSI	Received Signal Strength Indication
BRMS	Business Rules Management System
MPC	Model Predictive Control
PMV	Predicted Mean Vote
PPD	Predicted Percentage Dissatisfied
BMES	Building Energy Management System
SQL	Structured Query Language

JSON	JavaScript Object Notation
JDBC	Java Database Connectivity

References

1. Lah, N.; Mohammed, M.a.h.; Abdullah Mohd Asmoni, M. Office space study: A review from facilities management context. *J. Teknol.* **2015**, *75*. [CrossRef]
2. Shin, H.; Lee, H.S.; Park, M.; Lee, J. Facility Management Process of an Office Building. *J. Infrastruct. Syst.* **2018**, *24*, 04018017 . [CrossRef]
3. Houston Chapter of IFMA—What is Facility Management. Available online: https://ifmahouston.org/content.php?page=What_is_Facility_Management (accessed on 3 January 2022).
4. Elmualim, A.; Pelumi-Johnson, A. Application of computer-aided facilities management (CAFM) for intelligent buildings operation. *Facilities* **2009**, *27*, 421–428. [CrossRef]
5. Kastner, W.; Neugschwandtner, G.; Soucek, S.; Newman, H.M. Communication systems for building automation and control. *Proc. IEEE* **2005**, *93*, 1178–1203. [CrossRef]
6. Domingues, P.; Carreira, P.; Vieira, R.; Kastner, W. Building automation systems: Concepts and technology review. *Comput. Stand. Interfaces* **2016**, *45*, 1–12. [CrossRef]
7. Park, J.Y.; Ouf, M.; Gunay, B.; Peng, Y.; Kjærgaard, M.; Nagy, Z. A critical review of field implementations of occupant-centric building controls. *Build. Environ.* **2019**, *165*, 106351. [CrossRef]
8. Qiu, H.; Schneider, G.; Kauppinen, T.; Rudolph, S.; Steiger, S. Reasoning on Human Experiences of Indoor Environments Using Semantic Web Technologies. In Proceedings of the 35th International Symposium on Automation and Robotics in Construction, Berlin, Germany, 20–25 July 2018. [CrossRef]
9. Kastner, W.; Gaida, S.; Tellioğlu, H. Knowledge-based building management combining human perception and building automation systems. In Proceedings of the 2019 First International Conference on Societal Automation (SA), Krakow, Poland, 4–6 September 2019; pp. 1–6. [CrossRef]
10. Sakellaris, I.; Saraga, D.; Mandin, C.; Roda, C.; Fossati, S.; de Kluizenaar, Y.; Carrer, P.; Dimitroulopoulou, C.; Mihucz, V.; Szigeti, T.; Hänninen, O.; et al. Perceived Indoor Environment and Occupants' Comfort in European "Modern" Office Buildings: The OFFICAIR Study. *Int. J. Environ. Res. Public Health* **2016**, *13*, 444. [CrossRef] [PubMed]
11. Frontczak, M.; Wargocki, P. Literature survey on how different factors influence human comfort in indoor environments. *Build. Environ.* **2011**, *46*, 922–937. [CrossRef]
12. *Standard 55*; Thermal Environmental Conditions for Human Occupancy. American Society of Heating, Refrigerating and Air-Conditioning Engineers (ASHRAE): Peachtree Corners, GA, USA, 2010.
13. *ISO 7730:2005*; Ergonomics of the thermal environment—Analytical determination and interpretation of thermal comfort using calculation of the PMV and PPD indices and local thermal comfort criteria. International Organization for Standardization: Geneva, Switzerland, 2005.
14. *EN 12665*; Light and Lighting—Basic Terms and Criteria for Specifying Lighting Requirements. European Standards: Brussels, Belgium, 2018.
15. *EN 12464-1*; Light and Lighting. Lighting of Work Places Indoor Work Places. European Standards: Brussels, Belgium, 2018.
16. Navai, M.; Veitch, J. *Acoustic Satisfaction in Open-Plan Offices: Review and Recommendations*; Technical Report; National Research Council Canada: Ottawa, ON, Canada, 2003. [CrossRef]
17. Cowan, J.P. *Handbook of Environmental Acoustics*; John Wiley & Sons: New York, NY, USA, 1994.
18. *Standard 62.1*; Ventilation for Acceptable Indoor Air Quality. American Society of Heating, Refrigerating and Air-Conditioning Engineers (ASHRAE): Peachtree Corners, GA, USA, 2019.
19. Koch, C.; König, M.; Beetz, J.; Borrmann, A. *Building Information Modeling: Technologische Grundlagen und Industrielle Praxis*; VDI-Buch; Springer: Berlin/Heidelberg, Germany, 2015.
20. *ISO 16739:2013*; Industry Foundation Classes (IFC) for Data Sharing in the Construction and Facility Management Industries. International Organization for Standardization: Geneva, Switzerland, 2013.
21. Open Green Building XML Schema: A Building Information Modeling Solution for Our Green World. Available online: https://www.gbxml.org/ (accessed on 3 January 2022).
22. Pauwels, P.; Terkaj, W. EXPRESS to OWL for construction industry: Towards a recommendable and usable ifcOWL ontology. *Autom. Constr.* **2016**, *63*, 100–133. [CrossRef]
23. Brick, a Uniform Metadata Schema for Buildings. Available online: https://brickschema.org/ (accessed on 3 January 2022).
24. Hassenzahl, M.; Tractinsky, N. User experience—A research agenda. *Behav. Inf. Technol.* **2006**, *25*, 91–97. [CrossRef]
25. Comfy. Available online: https://www.comfyapp.com/ (accessed on 3 January 2022).
26. TherMOOstat. Available online: https://facilities.ucdavis.edu/energy-engineering/thermoostat (accessed on 3 January 2022).
27. Pritoni, M.; Salmon, K.; Sanguinetti, A.; Morejohn, J.; Modera, M. Occupant Thermal Feedback for Improved Efficiency in University Buildings. *Energy Build.* **2017**, *144*, 241–250. [CrossRef]

28. Gupta, S.K.; Atkinson, S.; O'Boyle, I.; Drogo, J.; Kar, K.; Mishra, S.; Wen, J.T. BEES: Real-time occupant feedback and environmental learning framework for collaborative thermal management in multi-zone, multi-occupant buildings. *Energy Build.* **2016**, *125*, 142–152. [CrossRef]
29. Winkler, D.A.; Beltran, A.; Esfahani, N.P.; Maglio, P.P.; Cerpa, A.E. FORCES: Feedback and control for occupants to refine comfort and energy savings. In Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp'16), Heidelberg, Germany, 12–16 September 2016; Association for Computing Machinery: New York, NY, USA, 2016; pp. 1188–1199. [CrossRef]
30. Zhao, Q.; Zhao, Y.; Wang, F.; Jiang, Y.; Jiang, Y.; Zhang, F. Preliminary study of learning individual thermal complaint behavior using one-class classifier for indoor environment control. *Build. Environ.* **2014**, *72*, 201–211. [CrossRef]
31. Lam, A.H.y.; Yuan, Y.; Wang, D. An Occupant-Participatory Approach for Thermal Comfort Enhancement and Energy Conservation in Buildings. In Proceedings of the 5th International Conference on Future Energy Systems (e-Energy'14), Xi'an, China, 3–6 November 2014; Association for Computing Machinery: New York, NY, USA, 2014; pp. 133–143. [CrossRef]
32. Rittenbruch, M.; Donovan, J.; Santo, Y. *Mini-Orb: A Personal Indoor Climate Preference Feedback Interface. Human-Computer Interaction—INTERACT 2015*; Abascal, J.; Barbosa, S.; Fetter, M.; Gross, T.; Palanque, P.; Winckler, M., Eds.; Lecture Notes in Computer Science; Springer International Publishing: Cham, Switzerland, 2015; pp. 134–149. [CrossRef]
33. Clear, A.; Mitchell Finnigan, S.; Olivier, P.; Comber, R. *ThermoKiosk: Investigating Roles for Digital Surveys of Thermal Experience in Workplace Comfort Management*; CHI: Montreal, QC, Canada, 2018; pp. 1–12. [CrossRef]
34. Barrios, L.; Kleiminger, W. The Comfstat—Automatically sensing thermal comfort for smart thermostats. In Proceedings of the 2017 IEEE International Conference on Pervasive Computing and Communications (PerCom), Kona, HI, USA, 13–17 March 2017, pp. 257–266. [CrossRef]
35. Abdallah, M.; Clevenger, C.; Vu, T.; Nguyen, A. Sensing Occupant Comfort Using Wearable Technologies. In Proceedings of the 2016 Construction Research Congress, San Juan, Puerto Rico, 31 May–2 June 2016; pp. 940–950. [CrossRef]
36. Kauppinen, T.; Litvinova, E.; Kallenbach, J. Capturing and Linking Human Sensor Observations with YouSense. In Proceedings of the International Semantic Web Conference, Riva del Garda, Italy, 19–23 October 2014; Volume 1272.
37. Winkler, D.A.; Yadav, A.; Chitu, C.; Cerpa, A.E. OFFICE: Optimization Framework For Improved Comfort amp; Efficiency. In Proceedings of the 2020 19th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN), Sydney, Australia, 21–24 April 2020; pp. 265–276. [CrossRef]
38. Rinaldi, S.; Bittenbinder, F.; Liu, C.; Bellagente, P.; Tagliabue, L.C.; Ciribini, A.L.C. Bi-directional interactions between users and cognitive buildings by means of smartphone app. In Proceedings of the 2016 IEEE International Smart Cities Conference (ISC2), Trento, Italy, 12–15 September 2016; pp. 1–6.
39. Tagliabue, L.C.; Cecconi, F.R.; Maltese, S.; Rinaldi, S.; Ciribini, A.L.C.; Flammini, A. Leveraging Digital Twin for Sustainability Assessment of an Educational Building. *Sustainability* **2021**, *13*, 480. [CrossRef]
40. Benyon, D.; Turner, P.; Turner, S. *Designing Interactive Systems: A Comprehensive Guide to HCI, UX and Interaction Design*; Harlow: Emeryville, CA, USA, 2005.
41. Mark, G.; Gudith, D.; Klocke, U. *The Cost of Interrupted Work: More Speed and Stress*; CHI: Montreal, QC, Canada, 2008. [CrossRef]
42. Pruitt, J.S.; Adlin, T. *The Persona Lifecycle*; Elsevier: Amsterdam, The Netherlands, 2006. [CrossRef]
43. OASIS. MQTT—Official Homepage. Available online: http://mqtt.org. (accessed on 3 January 2022).
44. Noy, N.; Mcguinness, D. Ontology Development 101: A Guide to Creating Your First Ontology. *Knowl. Syst. Lab.* **2001**, *32*, 1–28.
45. Busse, J.; Humm, B.; Lubbert, C.; Moelter, F.; Reibold, A.; Rewald, M.; Schlüter, V.; Seiler, B.; Tegtmeier, E.; Zeh, T. Was bedeutet eigentlich Ontologie? *Inform.-Spektrum* **2014**, *37*, 286–297. [CrossRef]
46. Stuckenschmidt, H. *Ontologien; Konzepte, Technologien und Anwendungen*; Informatik im Fokus; Springer: Berlin/Heidelberg, Germany, 2009.
47. Cimiano, P.; Maedche, A.; Staab, S.; Völker, J. *Ontology Learning*; Springer: Berlin/Heidelberg, Germany, 2009; pp. 245–267. [CrossRef]
48. Spring Framework Homepage. Available online: https://spring.io/projects/spring-framework (accessed on 3 January 2022).
49. Maven Homepage. Available online: https://maven.apache.org/ (accessed on 3 January 2022).
50. Docker Homepage. Available online: https://www.docker.com (accessed on 3 January 2022].
51. Google. OpenThread—Official Homepage. Available online: https://openthread.io/guides/thread-primer (accessed on 3 January 2022).
52. Drools—Business Rules Management System (Java™, Open Source). Available online: https://www.drools.org/ (accessed on 3 January 2022).