Design Guidelines for IoT Systems in Smart Companies

Grégoire, Jeroen Els

VRIJE
UNIVERSITEIT
BRUSSEL

Graduation thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Applied Sciences and Engineering: Toegepaste Informatica

DESIGN GUIDELINES FOR IOT SYSTEMS IN SMART COMPANIES

JEROEN GRÉGOIRE

Academic year 2020–2021

Promoter: Prof. Dr. Beat Signer
Advisor: Maxim Van de Wynckel
Faculty of Sciences and Bio-Engineering Sciences

VRIJE
UNIVERSITEIT
BRUSSEL

Afstudeereindwerk ingediend in gedeeltelijke vervulling van de eisen voor het behalen van de graad Master of Science in de Ingenieurswetenschappen: Toegepaste Computerwetenschappen

DESIGN GUIDELINES FOR IOT SYSTEMS IN SMART COMPANIES

JEROEN GRÉGOIRE

Academiejaar 2020–2021

Promoter: Prof. Dr. Beat Signer
Advisor: Maxim Van de Wynckel

Faculteit Wetenschappen en Bio-ingenieurswetenschappen

Declaration of Originality

I hereby declare that this thesis was entirely my own work and that any additional sources of information have been duly cited. I certify that, to the best of my knowledge, my thesis does not infringe upon anyone's copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. Furthermore, to the extent that I have included copyrighted material, I certify that I have obtained a written permission from the copyright owner(s) to include such material(s) in my thesis and have included copies of such copyright clearances to my appendix. I declare that this thesis has not been submitted for a higher degree to any other University or Institution.

Abstract (EN)

Smart offices hold many definitions and can take many forms. The easiest form is an automated task, performed by one or more computing entities, to facilitate the reduction of costs. The advancements in the field of smart office generally follow the development of technology and the adaption of manufacturers. Connecting Internet of Things (IoT) devices in an algorithm, or service does not require the same expert knowledge or programming effort then a couple of years ago. The focus has shifted from developing to adopting and integrating. Creating a smart office no longer requires a team of experts to set up a fixed set of routines with devices uniquely build for their solution.

The main goal of this thesis is to design guidelines for defining requirements to create a smart office. Starting from analysing existing research and related work in the broader field of smart offices, we describe different considerations to help choosing the right solution for the defined requirements. The pros and cons of different market available systems are used as a base for these considerations. A smart company is implemented following these guidelines using industry available systems.

Abstract (NL)

Smart office kan op verschillende manieren gedefiniëerd worden en kan verschillende vormen aannemen. In zijn meest eenvoudige vorm gaat het om een geautomatiseerde taak, uitgevoerd door één of meer computationele entiteiten, met het oog op een kostenbesparing. De vooruitgang op het gebied van smart office volgt over het algemeen de technologische ontwikkeling en de aanpassing hieraan door fabrikanten. Het verbinden van IoT toestellen, via een algoritme of service, vereist niet hetzelfde technische kennisniveau of programmeerwerk als tien jaar geleden. De focus is verschoven van ontwikkelen naar aanpassen en integreren. Het opzetten van een smart office vereist niet langer een team van experten om procedures te bepalen voor toestellen die uitsluitend voor één specifieke oplossing gebouwd zijn.

Het doel van deze thesis is het bepalen van design guidelines om de reuirements van een smart office te bepalen. Vanuit een analyse van eerder onderzoek en gelijkaardige implementaties in het ruimer vakgebied van smart offices, beschrijven we verschillende overwegingen die helpen om een juiste oplossing te kiezen voor de bepaalde requirements. De voor- en nadelen van verschillende beschikbare toepassingen worden gebruikt als basis voor deze overwegingen.We implementeren een smart office op basis van deze richtlijnen met systemen die in de markt beschikbaar zijn.

Acknowledgements

A thesis is the result of combining knowledge, gathered over a course of time. In my professional life I had the privilege to work with numerous talented people. Too many to mention, each adding a piece to experience, approach and reasoning. The VUB has offered me a variety of courses, given by inspirational people. The structured build throughout the bachelor and master program, has enriched me and offered me tools to continue learning. As a strong promoter of the VUB, I try to inspire potential students to subscribe, based on my own experience. In the light of this thesis, I want to thank Prof. Dr. Beat Signer and Maxim Van de Wynckel for accepting my thesis subject and helping me fine tune the ideas and structure.

I want to thank Sterre Janssens for supporting the process and structure of the thesis. Without her the project would not have been feasible.

For the content of the thesis, I want to thank the following people for the time invested in sharing their views and experience:

- Bart De Bruycker - DIRAC Industries NV
- Rastislav Mokris- DIRAC Industries SRO
- Katrien Geens - Flemish Radio and Television Broadcasting (VRT) NV
- Steven De Ruyter - VRT NV

Acronyms and Abbreviations

IoT Internet of Things

VRT Flemish Radio and Television Broadcasting

CoBra Context Broker Architecture

CMT Context Modelling Toolkit

openHAB open Home Automation Bus

NHC Niko Home Control

IFTTT If This Then That

TAPI Telephony Application Programming Interface

OWL Web Ontology Language

PPM Prediction by Partial Matching

ISSP integrated semantic service platform

WiFi Wireless Fidelity

SLAC Simultaneous Localisation and Configuration

RSSI received signal strength indicator

AI Artificial Intelligence

QoL Quality of Life

BLE Bluetooth Low Energy

HVAC heating, ventilation and air conditioning

API Application Programmable Interface

MQTT Message Queuing Telemetry Transport

REST Representational State Transfer

VUB Vrije Universiteit Brussel

SME Small and medium-sized enterprises

PaaS Platform as a service

UI User Interface

ROI Return on Investment

GDPR General Data Protection Regulation

PbD Privacy by Design

FTE full time equivalents

ERP Enterprise Resource Planning

VoIP Voice over Internet Protocol

CRM Customer relationship management

MAC address media access control address

RFID Radio-frequency identification

EMS energy management system

UPS Uninterruptible Power Supply

Contents

List of Figures

1

Introduction

1.1 Introduction

According to Senion[1], a smart office or smart company is a workplace where modern technology is leveraged to help employees work smarter, better, and faster. This is achieved by removing menial tasks for employees and unnecessary obstacles and activities that drain time and energy from the workforce. The smart company solutions let employees focus on what really matters. Typically, a smart company solution consists of a suite of

[1] https://senion.com

technologies that connect with the employees, the building and the existing IT infrastructure to achieve these goals.

A smart office automates routines, ultimately to save money for a company, whether by eliminating mistakes, improving productivity, or speeding up processes. The subject of smart offices has been researched a lot over the last decade ([2, 6, 8–15]). With the technological advancements and the rise of IoT, the possibilities for automating grow as well. Companies wanting to smarten or automate their procedures of infrastructure can choose between a range of solutions to achieve their goals. In fact, there are so many solutions, that finding the most suitable solution is not an easy task.

Any size of company can benefit from implementing a smart environment. Cost reduction and improving productivity are often part of investments in automation. Defining the requirements for a smart office starts with a specific problem. This is generally an error prone procedure, a time consuming activity or an effort to reduce costs. Although the initial requirements can be defined relatively easy because the problem is known, they also trigger additional requirements, like scalability, budget, redundancy, etc.

Therefore, companies could benefit from guidelines highlighting relevant considerations for defining requirements of a smart office. This resulted into the main objective of this thesis: to list considerations to help companies selecting the right smart solution for their requirements.

During our research we came across many approaches and specific implementations. Even though the topic of smart office is actively researched, both academically and by industry leading companies, we could not find a guideline to aid decision making on creating a smart company. we tried to find the best solution, but had to conclude that there is no single best solution. Solutions can be compared on parameters or benchmarked, but that does not lead to an absolute best solution. There are however better solutions for a defined set of requirements.

After studying the related work and gathering industry examples, the main objective of this thesis was refined to create design guidelines for smart companies. It resulted in a set of considerations that helps selecting the best solution for creating a smart company.

In Chapter 2 we start by providing an overview of research done in the field of Smart Offices. Next in Chapter 3, we go over related work by researching market available systems in the field of smart office spaces. The pros and cons of the researched work are discussed to facilitate the choice for a smart company implementation. After investigating market available implementations, the information of Chapter 2 and Chapter 3 is used as a base for design guidelines and considerations for a smart office. In Chapter 4 we offer considerations in the form of guidelines to help defining the full requirements for a smart company. Each guideline discusses a specific aspect that needs top be taken into account when selecting scope and mapping an implementation. In Chapter 5 we discuss a possible solution for each use case. In Chapter 6 we have taken an industry example from Chapter 5 and implemented a smart company by using the guidelines from Chapter 4. The smart company meets the defined requirements and is implemented using market available solutions. Finally, Chapter 7 finishes this thesis with a conclusion, limitations and future work.

2

Background

2.1 A Smart Office

The concept of a smart office is first described by A. K. Townsend in 1998 [16]. The book describes a smart office as several examples on how companies can save money. A smart office is an office that is managed efficiently, and where the word smart refers to methods on how a company can save money and increase its profits and productivity by streamlining its resource use and creating healthier workplaces. The smart office is filled with success stories of companies that have saved money, boosted employee morale, and increased productivity drastically by overcoming inefficient and unhealthy work practices. It offers tips, strategies, and methods for designing, renovating, and furnishing your workplace, conducting a smart office audit, establishing smarter company policies and goals, eradicating waste, and investing successfully. It also lists hundreds of companies, organizations, and government programs that offer additional information and smarter products. These strategies can be categorised in the following themes:

- Streamline resources: by optimising the planning or use of resources.
- Increase productivity: by increasing the output generated with the same amount of resources, the same amount of output generated with the same amount of resource, or the combination of both.
- Reduce costs: by reducing operating costs directly.

Later work on smart offices ([2, 14, 15, 17]) are additions to strategies or methods, in the respective timelines. As technology evolves, so do the possibilities. The same categories are expanded. Different techniques are researched to tackle specific problems, resulting in smarter offices.

2.2 Streamline Resources

In 1999, Akyol et al. [18] imagine a world in which every device has an embedded processor and a high-speed wireless link. Any two devices can talk to each other and you link devices together as needed to get your work done. Overcoming the problems, rather than cause frustration. Today we do not have to imagine communication between multiple devices any more. Finding a device without embedded processors is becoming a challenge. Since then, also with the rise of IoT, the technological advancement has taken a step forward. A set of protocols and means of communicating has been created and widely accepted.

As a practical example the vision of smart doorplates within an office building is introduced [1]. The doorplates are able to display current situational information about the office owner, to act on behalf of the office owner in case of absence, and to direct visitors to the current location of the office owner based on a location-tracking system.

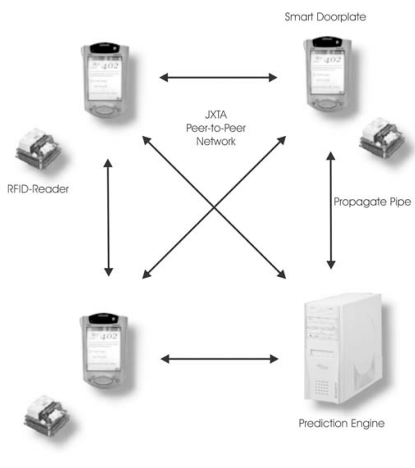


Figure 2.1: Sample application with a smart doorplate [1]

To combine resources for a smart office, the support of a computing system architecture is required to make all resources available for context reasoners.

An example is the Context Broker Architecture (CoBra) [19], a broker-centric agent architecture for pervasive context-aware systems. CoBra exploits the Web Ontology Language (OWL)[1] for supporting knowledge sharing and data fusion, using logic inferences for resolving and detecting inconsistent context knowledge, and provides users with a policy language to control their private information. Central to CoBra is an intelligent broker agent that maintains a share model of context for all agents, services, and devices in the space, and protects users' privacy by enforcing the policy rules that they have defined. The context reasoning mechanisms, and privacy protection will be discussed further in the Section 3.2.

Smart offices aim to integrate physical devices, human beings and computing entities in buildings. They have to support characteristics coming from mobile computing, ubiquitous computing and sensor networking in order to provide intelligent, context-aware and pro-active applications. In this vision, *context* plays a central role, but the raw context data, obtained by various data sources, are represented in heterogeneous formats and thus it is not possible to understand their meanings without a prior knowledge of the particular context information model. Therefore, data sources are not able to produce high-level context information, but provide only raw information. Such low-level information should be combined with a Semantic Context Service [20], that exploits Semantic Web technologies to support smart offices. In particular, this service relies on ontologies and rules to classify several typologies of entities, which can be involved in smart offices, and to infer higher-level context information from low-level information coming from positioning systems and sensors in the physical environments.

[1] `https://www.w3.org/OWL/`

According to Ramos et al. [21], the development of Smart Rooms projects with their appliances is very diversified. The appliances include projects in the context of workplace or everyday living, entertainment, play and education. These appliances envisage to acquire and apply knowledge about the environment state in order to reason about it so as to define a desired state for its inhabitants and perform adaptation to these desires and therefore improving their involvement and satisfaction with that environment.

Interdisciplinary research from the domains of pervasive computing or ubiquitous computing, human-computer-interaction and computer science has led to the development of many intelligent environments. While several middleware [22] has been developed in this field, no standard middleware for intelligent environments or ubiquitous computing has evolved yet.

The lack of a standard middleware for distributed sensor-actuator environments is one of the key issues limiting research on intelligent environments. In addition, the advent of personal robotics for healthcare and ambient assisted living scenarios in the context of ubiquitous computing was expected to emerge in the years following 2010 [22]. A robotic middleware can be used as glue between sensors, actuators and services and its application in a deployed example scenario. Thereby, we verify by examples the applicability of robotic middleware for complex ubiquitous computing environments.

Ubiquitous systems use context information to adapt appliance behaviour to human needs. Even more convenience is reached if the appliance foresees the user's desires and acts proactively. With context prediction techniques based on previous behaviour patterns, as described by Petzold et al. [10], a person's next movement can be anticipated. The predicted users next move can be used as a resource. They focus on two-level predictors with global first-level histories and two-state predictors respectively frequency analysis in the second level and compare these predictors with the prediction by the Prediction by Partial Matching (PPM) method [23]. The predictors reach an accuracy up to 96% with pre-training. In the authors' later work [24] they rely on different prediction techniques like Neural networks, Bayesian networks, State and Markov predictors. Exactly the same evaluation setup and benchmarks are used to compare the different methods.

According to Furdík et al. [2], a smart office system based on the LinkSmart[2] [25] semantic middleware, enables a construction of distributed enterprise IoT applications. LinkSmart was enhanced on features of seamless networking and interoperability of connected devices, deploying the approach of user experience monitoring and continuous evaluation for adjusting the application to desired knowledge, social, and business parameters.

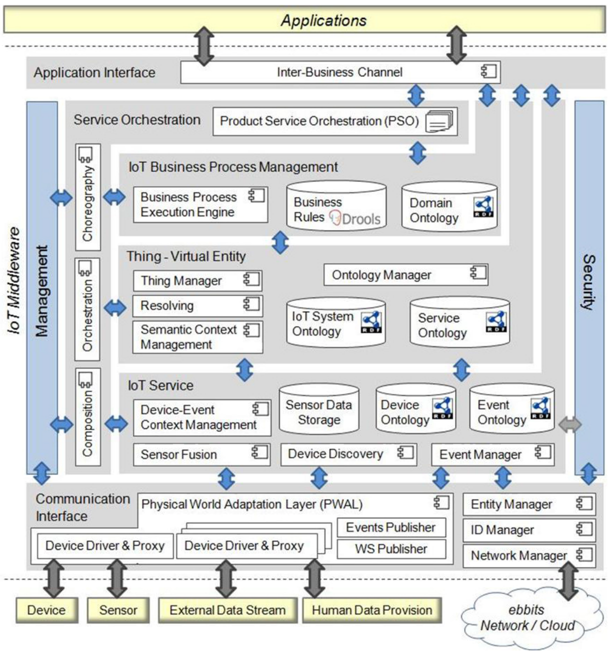


Figure 2.2: Modular architecture of the LinkSmart IoT middleware [2]

[2]https://linksmart.eu

Combining the context of IoT with semantic technologies, we can build integrated semantic systems to support semantic interoperability like integrated semantic service platform (ISSP), to support ontological models in various IoT-based service domains. Ryu et al. [26] address three main problems for providing integrated semantic services together with IoT systems: semantic discovery, dynamic semantic representation and semantic data repository for IoT resources. To show the feasibility of the integrated semantic service platform (ISSP), they developed a prototype service for a smart office using the ISSP, which can provide a preset, personalised office environment by interpreting user text input via a smartphone.

Indoor localisation is of great importance for a range of pervasive applications, attracting many research efforts in the past decades ([27–29]). Most radio-based solutions require a process of site survey, in which radio signatures of an interested area are annotated with their real recorded locations. Site survey involves intensive costs on manpower and time, limiting the applicable buildings of wireless localisation worldwide. Yang et al. [27] investigate novel sensors integrated in modern mobile phones and leverage user motions to construct the radio map of a floor plan, which is previously obtained only by site survey. On this basis, they designed LiFS, an indoor localisation system based on off-the-shelf Wireless Fidelity (WiFi) infrastructure and mobile phones.

The indoor localisation problem in smart offices is more than just finding the whereabouts of users. Finding positions of users relative to the devices of a smart space is just as important. Configuring such systems manually requires expert knowledge and is not resilient to changes in the environment. Rossum et al. [29] propose a new system, called Simultaneous Localisation and Configuration (SLAC), to address these two challenges, by combining the user and the devices a single estimation problem. The SLAC algorithm is able to locate devices using the received signal strength indicator (RSSI) of devices and motion data from users. Simulations have been used to show two main effects on the localisation performance: the amount of received signal strength indicator (RSSI) updates and the location of devices in the space. Live tests, in non-trivial environments, showed that room-level accuracy can be achieved and that the localisation can be performed in real time.

By reversing the sender and receiver, Van de Winckel [28] defined indoor navigation by centralised tracking. Using only the smartphone of a person, the system should be able to track and guide the user to their destination. Unlike other possible solutions that focus on the smartphone application to determine the position, the proposed solution uses a centralised server with Bluetooth scanners. These scanners are distributed throughout the hospital and report their scan results back to the centralised server for position calculation and navigation instruction generation. Finally, these instructions are sent to the smartphone application to guide the user to their destination.

2.3 Increase Productivity

Besides streamlining resources, smart office strategies can by used with the purpose of increasing productivity. As a new Artificial Intelligence (AI) application to our everyday life, Mizoguchi [30] designed and implemented a smart office environment in which various information appliances work collaboratively to support our office activities, such as deliver documents and printed out papers to a worker's room. The delivery task is a typical example of an important class of tasks supporting humans in the smart office. The collaboration between the agents is realised by software components including device controller, multiagent programming, a robot access server on the Web and a remote control browser. These components integrate various office devices and sensors, and a user can control the robots through a cellular phone and a browser.

The goal of a smart office is to enable the user to work as in a normal office. The office's intelligence observes the user in order to anticipate their intentions and augments their environment to communicate useful information. In the year 2000, this implies computers are involved in user activities in order to help in everyday tasks [31]. The system interacts with users using voice, gesture or movement. The Intelligent Environment provides a test-bed for collaboration and combination of independent modules integrated into a single coherent application [**?**]. Integration requires a flexible working environment in which module developers should not worry about low-level communication between modules. Two major modules are considered: the first module is the MagicBoard, an ordinary whiteboard augmented by a camera and a video projector; the second is a user localisation module.

Data in smart offices will be read from single or multi-sensor devices and will be used for context extraction. New location-based services will be adapted to user preferences. Knowing user profiles, user preferences and habits play an important role in the context extraction. In 2006 an idea was described by Bagci et al. [12] where the user profile is stored with the user, in the form of wearable or portable computers (or the smart environment takes responsibility for transporting them). The amount of new devices and services makes an efficient use by centralised systems very difficult. The idea is that a virtual reflection of the user is represented by a mobile agent accompanying in the smart environment. Mobile agents offer a possibility to encapsulate information of a person and the person's preferences and perform location-based services of the ubiquitous system in the name of the user. Security and privacy are major concerns of such an agent system. This paper describes a ubiquitous mobile agent system named UbiMAS which has security extensions to provide high protection of agents and significant personal data. UbiMAS is applied in the smart doorplate project as part of a smart office environment.

Security and privacy are major concerns of smart offices. A cross-cultural study analysing the willingness of users to share context information in work environments, aims to bring insights. The focus of the study by Röcker [32, 33] is on three aspects: the general willingness to provide different types of context information, the acceptance of manual and automated data capturing mechanisms and the identification of personal and cultural differences among users. The results of the study show that potential users are rather reluctant to provide context information, especially if the data is automatically captured by the system, and that the willingness to provide context information differs significantly between user groups with different cultural backgrounds and different degrees of computer knowledge.

A smart office system that recognises office workers' mental and physiological states to improve their Quality of Life (QoL) at the office [34], is an example of user data captured by a smart office system. Wearable, sensor-equipped devices - often referred to as wearables - became increasingly public accessible during the last years. Health and fitness wearables offer ubiquitous and continuous sensing of various aspects of our life. The rising social acceptance of body-worn technology is also a driver for the increasing adoption of wearables on the consumer level. It allows monitoring the state of health of a worker in a smart office.

IoT contains a high potential to create value for users in a smart office. The existing applications comprise a large quantity of systems and services, but there are still application scenarios, which cannot be covered by existing solutions [35]. Examples for uncovered applications are shared office cleaning services or a flexible in-room coffee service. The acceptance of technology for the integration, the management and the control of service devices, like a network calendar such as MS Exchange and a voice service like Amazon Alexa, increases with the availability in the users day-to-day life. The more value for the user, the higher the level of acceptance.

Luis [34] presented a smart office chair in 2012. Because most people spend the majority of the day in an office environment, he aims to improve their Quality of Life (QoL). By monitoring the worker's mental and physiological states, such as sleepiness and concentration, the chairs alters the working environment by multi-modal displays like a motion chair, a variable colour-temperature LED light and a hypersonic directional speaker. The changes have an effect on the workers' performance.

Abstract occupational stress is increasingly present in our society. Usually, it is detected too late, resulting in physical and mental health problems for the worker, as well as economic losses for the companies due to the consequent absenteeism, presentism, reduced motivation or staff turnover. Therefore, the development of early stress detection systems that allow individuals to take timely action and prevent irreversible damage is required. To address this need, Alberdi et al. [3] investigated a method to analyse changes in physiological and behavioural patterns using unobtrusively and ubiquitously gathered smart office data. Results confirm the predictability of behavioural changes for stress and mental workload levels, as well as for change in workload conditions. Results also suggest that computer-use patterns together with body posture and movements are the best predictors for this purpose. Moreover, the importance of self-reported scores' standardisation and the suitability of the NASA Task Load Index test for workload assessment is noticed. This work contributes significantly towards the development of an unobtrusive and ubiquitous early stress detection system in smart office environments, whose implementation in the industrial environment would make a great beneficial impact on workers' health status and on the economy of companies.

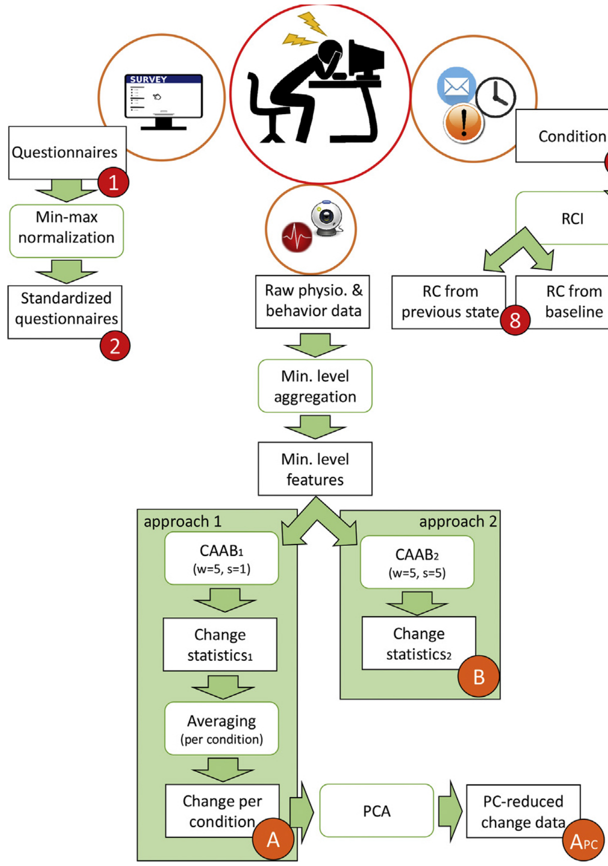


Figure 2.3: Flow-chart of the whole preprocessing method [3]

2.4 Reduce Costs

Streamlining resources and increasing productivity ultimately lead to reducing costs. A smart office energy management system impacts energy costs. A system can be implemented using Bluetooth Low Energy (BLE) beacons, smart plugs, and a mobile app, to provide the location-aware energy-saving service without user intervention [8]. The reasonable real-time energy-saving can produce meaningful energy savings for PCs, monitors, and lights. They can be used to determine whether a user enters or exits the office to change the power saving mode of the user's PCs, monitors, and lights [15, 36]. These sensor readings can be conveyed to the occupant of an office as effective feedback for motivation or instruction. By installing sensors to gather data of water and electricity consumption, behaviour patterns of workers can be determined [13]. A feedback system would encourage the workers to decrease their resource consumption, which could lead to an important energy saving. A use case of intelligent processing of events coming from the conventional inexpensive sensors in order to support better energy consumption in commercial buildings was setup by Stojanovic in 2011 [37].

Occupancy detection plays an important role in many smart building applications such as controlling heating, ventilation and air conditioning (HVAC) systems, monitoring systems and managing lighting systems. Many techniques for detecting occupancy require multiple sensors fusion for attaining acceptable performance. These techniques come with an increased cost and incur extra expenses of installation and maintenance as well. All of these methods are intended to deal with only two states: when a user is present or absent and control the system accordingly. Akbar et al. (2015) [14] propose a non-intrusive approach to detect an occupancy state in a smart office using electricity consumption data. By introducing the concept of a third state as standby, they suggest a solution for situations when the user leaves their seat for small breaks.

The evolution of the electricity grid towards the smart grid paradigm is fostering the integration of distributed renewable energy sources in Smart Buildings: a combination of local power generation, battery storage and controllable loads can greatly increase the energetic self-sufficiency of a *smart building*, enabling it to operate in island mode or to participate in an Automatic Demand Response framework, thus taking advantage of time-variable tariffs to achieve economical savings [38, 39]. An energy management system specifically tailored for a smart office building, which relies on actual data and on forecasting algorithms to predict the future patterns of both local energy generation and power loads.

Nowadays, despite the use of efficient LED lighting, lighting consumes a considerable amount of energy. To reduce the energy consumption, many office lighting systems are equipped with occupancy sensors. Since these sensors have a limited reliability in detecting presence, usually very conservative strategies [40] are used such as keeping the lights on for fifteen minutes after the last detected presence.

3
Related Work

3.1 Context of a Smart Office

A smart office can range from a small office building to a vast production facility. Each situation has different challenges, different users and thus different actions to be automated. What every smart office has in common on an abstract level, are the different building blocks. Various sensors, hardware and software, generate context for a smart office. The context is meaningful information about a location, its environmental attributes (e.g., noise level, light intensity, temperature and motion) and the people, devices, objects and software agents it contains. Context-aware systems are computer systems that can provide relevant services and information to users by exploiting context. Context may also include system capabilities, services offered and sought, the activities and tasks in which people and computing entities are engaged, and their situational roles, beliefs, and intentions [19]. All of these data generation feeds reasoners are used to calculate actionable events, or generate more complex context.

3.2 Context Broker Architecture (CoBra)

Context Broker Architecture (CoBra) is an agent-based architecture for supporting context-aware systems in smart spaces (e.g., intelligent meeting rooms, smart homes, and smart vehicles) [41]. Central to this architecture is an intelligent agent called context broker that maintains a shared model of context on behalf of a community of agents, services, and devices in the space and provides privacy protections for the users in the space by enforcing the policy rules that they define. Context-aware systems exploit the use of situational information, or context, to provide relevant information and services to users.

COBRA-ONT is an ontology, developed as a part of CoBra. COBRA-ONT is a collection of ontologies, expressed in the Web Ontology Language OWL, for describing places, agents and events and their associated properties in an intelligent smart office domain. It can be described as an inference engine for context reasoning with information expressed using the COBRA-ONT ontology [42].

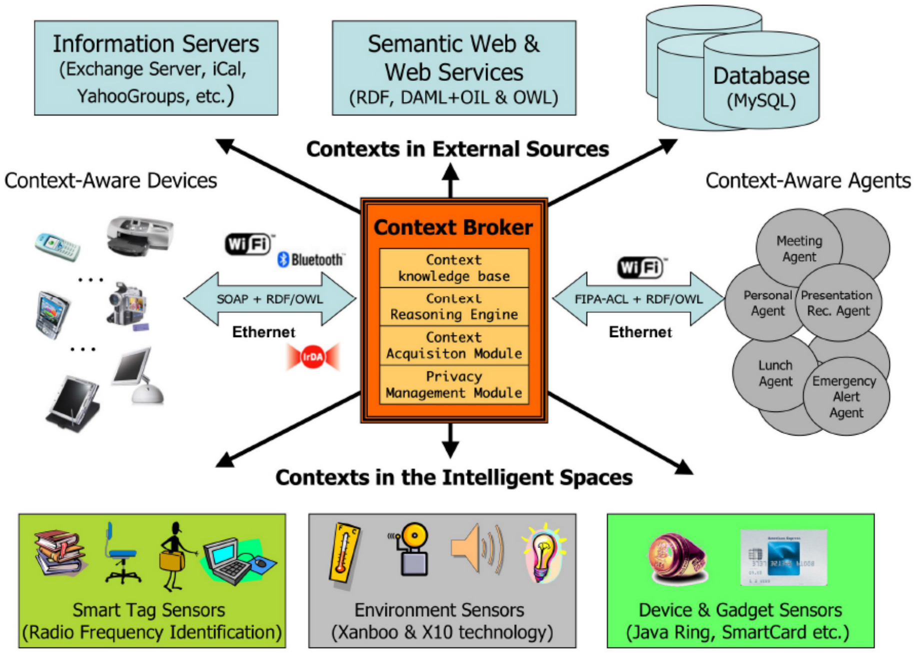


Figure 3.1: An intelligent context broker acquires context information from devices, agents and sensors in its environment and fuses it into a coherent model, which is then shared with the devices and their agents [4]

3.2.1 Strong Points

CoBra offers a strong support for knowledge sharing and context reasoning. There is no need for low-level implementation agreements between programs that wish to share information. A Context broker allows easy knowledge sharing in an open and dynamic environment. Because the context-related vocabularies are used with explicit shared semantics, there is no room for ambiguous interpretations. A centralised oracle, holding all the knowledge in a system, allows the creation of complex rules. The implementation in a semantically standardised language makes the access available for a large number of users.

The different components can be hosted on several servers, allowing the smart office of parts thereof to continue in case a server cannot be reached. By using ontologies, semantic vocabularies can be agreed and shared by the different components. Querying the broker can be done on an efficient manner. The developer can assume all the different components use the same standard and semantics.This makes any addition to the smart office more straight forward. A developer who is familiar with ontologies, only needs to understand these semantics.

3.2.2 Limitations

Implementing a smart office using CoBra requires a developer with expert knowledge to define an ontology including possible situations and their description in the OWL ontology language. All the different context elements require an ontology implementation. This can lead to huge overhead when expanding the scope, or updating to new industry standards.

Changing the ontology at runtime introduces possible issues with respect to conflicts and ontology integrity, and thus need to be compiled. It is not ideal for dynamic context modelling.

All computing entities in a smart office are presumed to have prior knowledge about the presence of a context broker, and the high-level agents are presumed to communicate with the broker.

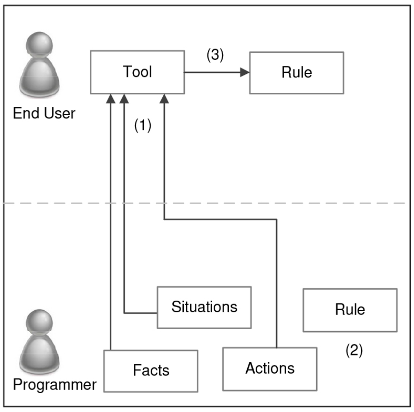
A centralised server is required to run and link the broker and context computing entities. The centralised design of a broker could create a "bottleneck" situation in a large-scale intelligent space such as a building or a university campus, hindering overall system performance.

It will only work if there is a standard, with agreed semantics, used by all software developers. There is a learning curve and requirement to studying the semantics. Defining the semantics will become hazardous when they need to be agreed by many developers, or when a new addition requires altered semantics.

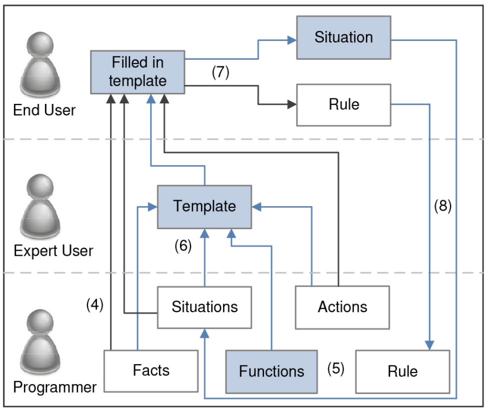
3.3 Context Modelling Toolkit (CMT)

The Context Modelling Toolkit (CMT) is a context modelling framework [5] which enables a seamless transition between the different levels of expertise of programmers who are experienced with low-level programming, expert users with little or no programming experience as well as end users without any programming knowledge.

Programmers are responsible for providing interfaces with devices and applications since this step requires programming skills. Expert users can create templates for rules which only require some basic knowledge about data types and declarative concepts. Finally, end users can fill in rule templates with data instances such as a specific device but do not have to be familiar with the internal logic of a rule. Users can switch between the three user types if they have the necessary expertise for a particular level. The CMT framework further enables the creation of new routines at runtime and fosters the reuse of the custom situations at all three user levels. CMT is implemented as a rule-based client-server architecture. While the server side takes care of the context reasoning, clients feed the server with context data such as facts (i.e. persistent data such as a person), situations (i.e. temporal data such as the current temperature) and actions which can be invoked when situations are detected.



(a) Existing context modelling approaches



(b) Multi-layered CMT approach

Figure 3.2: Difference between existing context modelling approaches and CMT [5]

3.3.1 Strong Points

Although CMT is not actively maintained, it offers a variety of benefits that hold their truth. The CMT is in essence a context broker, focusing more on the modelling of the context. Different components are linked to this middleware using modern Application Programmable Interface (API)s. The modelling is done using a visual design tool. A visual approach lowers the obstacles for smart office architects, allowing for an intuitive and human readable setup of rules with a low learning curve.

The interfacing with the different components needs to be created by a software developer, but the creation of the context is done by expert users and domain experts. Complex rules can be easily created by using user defined logic and actions. User rights and access control guards abuse of automated routines. The semantics can be decided by the domain experts, and shared with the users with administration rights.

The debugging of the smart office can be done in an own managed system with full control and transparency on linked components for the users. The expert users have good control over the linked elements. There is no need for (re)compiling as context can be added in real time.

3.3.2 Limitations

Just like other middleware solutions, a server is needed to run the software. With heavy usage, traffic issues can occur, which may effect the response time of the server. Worst case, the server goes down, making it a single point of failure. When the server becomes unavailable, there is no redundancy for the smart office.

Each addition to the system needs to be interfaced with the CMT by a developer. The interfacing is limited to the accepted protocols of the CMT. Alternatively, the accepted ways of communicating can be extended. This requires all the linked computing entities to have the option of interfacing. The maintenance cost can potentially be high. Because there is no sharing or predefined semantics, each new installation needs to be setup from scratch.

3.4 Linksmart

LinkSmart®[1] is an open source IoT platform offering services for device abstraction, data storage and machine learning. The linksmart platform aims to eliminate the burden for developers trying to develop IoT applications. It consists out of three main components. A localConnect (LC) module for local device and service discovery and linking. The globalConnect (GC) extends the localConnect to global networks and services. The last module is the Linksmart Services module. This module acts as a middleware service for IoT Systems and platforms. LinkSmart® uses wide spread standards to ease the integration with your services and systems: OGC SensorThings, Message Queuing Telemetry Transport (MQTT) and Representational State Transfer (REST).

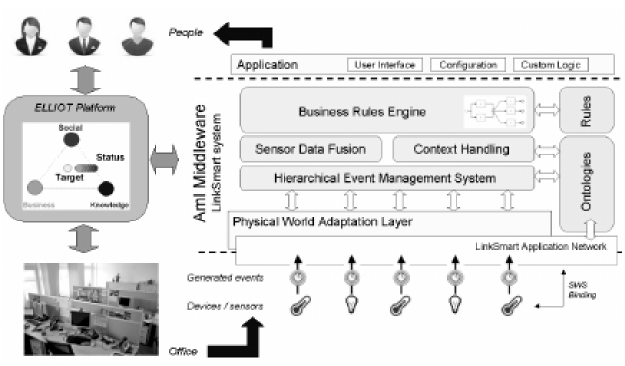


Figure 3.3: The LinkSmart middleware included in the smart office application [6]

3.4.1 Strong Points

As an open source platform, LinkSmart® is extended and maintained by a community. Community members can add modules and functionality to the platform. Once final, they make them publicly available via GitHub[2].

[1] https://linksmart.eu
[2] https://github.com

Many companies have implemented the LinkSmart® framework successfully, connecting numerous systems. Because it originates from an academic project there is ongoing research keeping the framework up to date with the latest challenges. Furthermore, the platform relies on standards, which adds to uniformity.

A big advantage of the LinkSmart® platform is the modular structure. There is no need to install the whole system; implementers can make a selection, based on the specific needs of the customers or users. As a centralised context middleware, all the components have access to the complete context.

3.4.2 Limitations

The community is poorly managed and without involvement in the core software. The biggest contributors are employed by companies that have a LinkSmart® installation on-premise. They are creating small modules to extend their running installation. Expert knowledge is required to use LinkSmart®. The guidelines for creating new add-ons are poorly documented, relying mostly on user created videos and forums.

The software needs to be installed on a server, which can lead to a bottleneck with high-traffic, or long pings. When using a contributed module, there is no control of quality on the development. These modules are not generic but created as a very specific solution. The risk lies in updating the core software. All the modules needs to be updated after framework changes. When a module is abandoned, it may stop working after an upgrade. Debugging is difficult with all the levels of abstraction, and the various modules altering the context.

3.5 openHAB

The openHAB[3] is an open source home automation software written in Java. It is deployed on-premise and connects to devices and services from different vendors. As of 2019, close to 300 bindings are available. Actions, such as switching on lights, are triggered by rules, voice commands, or controls on the openHAB user interface. In 2013, the core functionality became an official project of the Eclipse Foundation under the name Eclipse SmartHome. openHAB is based on Eclipse SmartHome and remains the project for the development of bindings. According to Black Duck[4], It is a powerful and flexible engine to design rules, with time and event-based triggers, scripts, actions, notifications and voice control.

3.5.1 Strong Points

openHAB platform is open source, allowing a community to extend and maintain the system. This active community is considered as one of the largest open-source teams in the world. The openHAB development is road mapped with continuous improvements. The installation procedure, use and development of bindings is well documented with endless tutorials and solutions for known issues. New issues can be escalated in a ticketing system, making bugfixes organised.

[3] `https://www.openhab.org`
[4] `https://www.blackducksoftware.com`

There is a proven industry track record with numerous running systems and a large number of existing systems linked. With over 300 bindings with modern devices and protocols, setting up smart office is shown to be easy if a connection already exists. The community is large enough to follow new computing entity providers, limiting the waiting time of integration options. The bindings use the same logic and generic template for all modules, making the code transparent for trained users. Being a modular system, you only need to install what you use, maximising the processing power for requirements only.

3.5.2 Limitations

The security and data sharing is not controllable because of the many layers of abstraction. You can only rely on the community reporting vulnerabilities when they are discovered. Companies sharing sensitive data might want to consider a more closed environment. Debugging modules can be a hand full with the distributed code and the hides the quality of development.

Although well documented, expert knowledge is required to learn how to use the system. Because of the large number of tutorials, a solid amount of time needs to be invested learning how to use openHAB.

The community maintains the connectors. Maintenance of connectors follows the need for update of the creator. Finding the right person to report issues too can lead to delays in setting up a smart office. In case of an update of the core, the large number of modules need to be updated as well, slowing down the roll out of a new version.

There is a need for a server, which can lead to a bottleneck with high-traffic, or long pings. If the openHAB installation automates business critical processes, any down time can lead to high costs.

3.6 Home Assistant

Home Assistant[5] is an open source home automation that puts local control and privacy first. Powered by a worldwide community of tinkerers and DIY enthusiasts. Perfect to run on a Raspberry Pi or a local server. It can be accessed via a web-based user interface, via companion apps for Android and iOS, or using voice commands via a supported virtual assistant like Google Assistant or Amazon Alexa.

Home Assistant acts as a central smart home controller hub by combining different devices and services in a single place and integrating them as entities. The provided rule-based system for automations allows creating custom routines based on a trigger event, conditions and actions, including scripts. These enable building automation, alarm management of security alarms and video surveillance for home security system as well as monitoring of energy measuring devices.

IoT technologies, devices, software, applications and services are supported by modular integration components, which not only include native integrations for local connectivity protocols like Bluetooth, MQTT, Zigbee and Z-Wave, but also offer support for controlling proprietary ecosystems if they provide access via an public API for third-party interface.

[5] https://www.home-assistant.io

3.6.1 Strong Points

Home Assistant is focussed on *do it yourself* and non-expert users, making the software accessible for the majority of users. It comes with a strongly developed smartphone app, and allows easy linking. Many suppliers of home appliances have products that interact with Home Assistant. Even though the focus lies on domestic use, the available devices are used in office spaces as well. The solution integrates with larger services like IFTTT, making it perfect to bridge the control of smaller components.

The use is well documented with easy to follow tutorials, without the need of development. The development is done by the hardware suppliers. Every linked product is hipped with instructions how to link and use the device with Home Assistant. Linking a product can be executed at the same time of installing the hardware.

3.6.2 Limitations

Home assistant only works with a set of products made available by the supplier. Linking other devices requires a license and is costly for unique integrations. There is no control on security and requires a server with cloud connectivity. Even though the code is open source, the business model behind relies on adding value for manufacturers. Using Home assistant has benefits if the devices installed in a smart office have been selected from the list of compatible devices, limiting the choice of the manufacturers.

Being open sourced offers limited benefits as a complete solution. Using Home assistant as a bridge between a device and other services, like IFTTT, is more likely

3.7 Problem Statement

Smart offices or office automation is a field that is researched continuously. The work done on an academic level offers solutions for specific problems, often for sharing information between different systems. The systems available in the market follow the general research. The most commonly used automation solutions today are open source and offer various functionality out of he box. Some are more focussed on expandability, whilst others focus on the ease installation and configuration. A particular solution is not better than the next solution, only better for specific requirements. Choosing the best solution starts with defining the requirements. Companies looking to automate an unique problem might be better off implementing a solution from scratch. Large companies, like Vrije Universiteit Brussel (VUB), can benefit from standardised solution like ontologies. Small and medium-sized enterprises (SME) can find an out of the box packet that provides the required automation.

To help defining the requirements, and thus finding a suitable solution, different aspects need to to be considered. In the next chapter we will list additional aspects to be considered in the form of guidelines. These guidelines will help defining requirements and ultimately help selecting a smart solution.

4

Design guidelines

4.1 Introduction

Designing a smart company starts with identifying the automation needs and defining the requirements. There exists a wide range of frameworks for creating a smart company. The requirements will help narrowing down the existing frameworks to a manageable list. A gap analysis brings clarity on the requirements that are not met by an existing solution. Hard requirements, also seen as deal breakers, are the mandatory requirements. In many cases gaps can be filled by custom developing the missing functionality or re-evaluating the requirements.

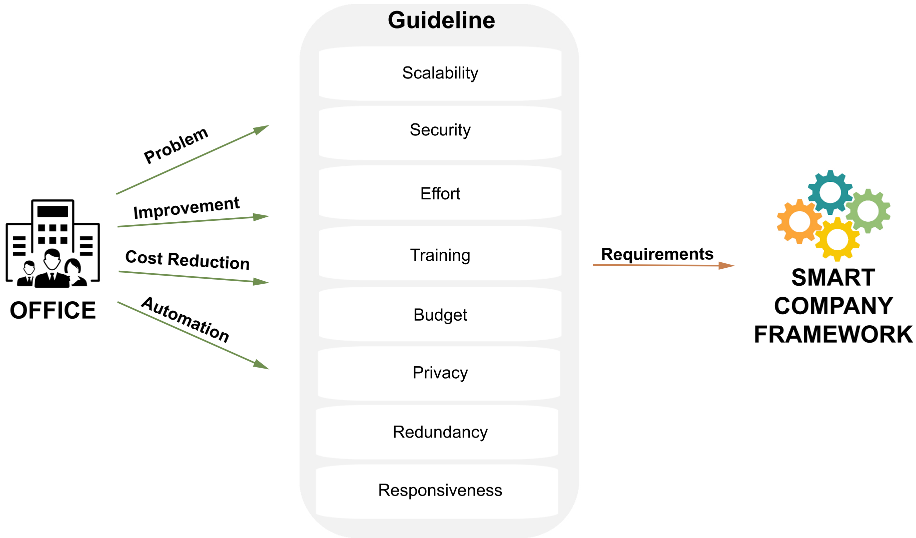


Figure 4.1: Using design guidelines to build requirements

The success of a smart company implementation starts by setting the boundaries of the automation. Defining requirements for a set of procedures or routines tends to be straightforward. The complexity is determined by the number of requirements. In this chapter we highlight different aspects to be considered when designing a smart company. These guidelines offer insight in selecting the best approach for an automation.

4.2 Scalability

The scalability has a big, if not the biggest, impact on all the aspects of a smart company implementation. Allowing high flexibility in scale will influence the required effort and budget needs. Some level of flexibility is often required to keep later expansions possible. The number of users, triggers and actions or routines are mostly not a concern when using middleware systems. For open systems from hardware suppliers this can be a limitation. A benefit of middleware systems is the choice of installation medium. Linksmart of openHab can be installed on server starting from Raspberry pi[1] to a full on-premise server or in the cloud as Platform as a service (PaaS). The computing resources like bandwidth, memory, storage and processing can be upscaled if desired, or the installation can be migrated to another type of server. Cloud based solutions generally come with a monthly fee that corresponds with a service level and available resources. There is a direct link on budget and the cloud parameters.

Scaling linked IoT components is also easier using middleware. The purpose of middleware is, amongst others, to add a middle layer that acts as a context broker between various components. This implies the middleware needs to accommodate communication using different protocols. An IoT component, which uses a modern communication protocol, is very likely linkable to middleware. In open source systems the popular components have connectors available. Closed systems like home assistant is limited to a list of compatible IoT devices. If all the components of a smart company can be selected from that list, the implementation has benefits in effort and redundancy. Home assistant is such an example. Hardware providers can purchase a license to link their products, but creating a unique connection is not as straightforward as with a community driven open-source framework.

Scale can be part of the release cycle. Creating a proof of concept in larger projects is common practice. As the investment size follows the size of the company, there is little room for failed project. Pilot projects can serve as initial release for a select number of users, or a select number of automations. After a defined test period, the implementation will be rolled out to the entire users base, or be expanded to automate additional processes. Accommodating this type of scalability is recommended for a successful implementation in larger projects. The scalability is very

[1]`https://www.raspberrypi.org`

important for these implementations, as it is certain they will scale up. The scope of the full project should be taken into account when selecting the right solution. The effort and budget makes sense for companies with access to multiple teams and needed technical skills. In practice, this kind of scale is uncommon.

Smart companies where the required scale can not be fixed at the start of project, are better of choosing a middleware solution. The extra effort for maintaining, learning and configuring a middleware solution is negligible in comparison with the level of flexibility. When the scale is known and the implementation can be realised using a closed framework, like home assistant or open software from hardware supplier, the effort and budget can be limited. If visual design tools and codeless design interfaces allow domain experts to define routine without the need of a developer, a closed framework can be recommended over middleware systems.

4.3 Security

A second aspect defining the implementation of a smart office strategy is security. A smart company connects per definition with computing entities and components. This implies various security risks. Protection of sensitive data or misuse by external attackers becomes relevant when connecting IoT components. The scalability requirement results in a list of possible solutions. A cloud solution has different security risks than a closed local network system. State-of-the-art security is required when making the automation publicly accessible. The first step is to choose a framework with the smallest security risks. A smart company where outside access can be blocked has different security needs. Making a risk assessment and determining how the risks are covered is advisable. A bank with cloud based access control to the vault requires a stricter policy then a light in the warehouse of a maintenance company. User rights and authentication, encrypted communication or network isolation are recommended minimal security measures.

Working with frameworks offers clear benefits. Bespoke development requires implementing security as well. Installing middleware on on-premise servers requires server and network security to be implemented. Cloud solutions offer security as part of the service level. IFTTT can be expected to follow latest security, where as openHab might require all modules to be updated to close a security hole. The difficulty lies in custom modules in open source frameworks. The used libraries and the quality of the code is often hidden behind layers of abstraction, complicating risk assessments. Choosing modules that are actively maintained in open source projects with large communities helps identifying security risks in those modules. An actively maintained module reduces long waiting times for update. Same applies for release cycles of the framework. A middleware system that is compromised and with access to all components threatens the security of the entire smart company. Therefore it is recommended to regularly check for security updates in modules and frameworks. Large communities like for the openHab project have a roadmapped release cycle. They use the communities strength in numbers to continuously validate the framework's security and apply the latest techniques in security. A closed system, like Home Assistant, is generally less prone to security issues. The security is the responsibility of the supplier in these cases.

4.4 Effort

The required effort can be a decisive factor when implementing a smart company. Effort can be seen as the total time needed to design, develop and operate the smart company. The impact can be a determining factor, thus requires a clear definition of what to expect. Effort can be grouped in three types: development, maintenance and usage.

4.4.1 Development

Middleware solutions bring many benefits to any smart company project. The overhead which comes with a software implementation does not have to be coded. Data persistence, user access rights, User Interface (UI) and API are part of the core of commonly used middleware. Popular IoT components can often be connected via modules. In case of an open source community, the modules are also updated and maintained by a community of contributor. The effort shifts from developing to configuring and administrating. If all the components and routines can be set up in a middleware system, the effort of installing is minimal. IoT components can be selected with compatibility to a specific middleware, if the choice of hardware is part of a project. Open source frameworks are built with generic expandability in mind. Online tutorials and user fora lower obstacles to create missing connections or integrations. As long as the IoT components come with modern means of communication, adding components can be done with acceptable development effort. Defining the routines is an important, if not the most important, part of the development. Domain experts hold the knowledge for the automation. Visual design tools, like CMT, have shown that codeless building of automations lowers the required skills, and thus the obstacles, for setting up a routine. Allowing domain experts to implement themselves speeds up a smart company project. Communicating mistakes between developers and domain experts ban be minimised or even eliminated if creating and testing is done by a domain expert.

4.4.2 Maintenance

The maintenance requires an effort that is often underestimated and has a big impact on the long term. Custom code will require bugfixes, middleware requires updates and hardware needs to be regularly checked on signs of failure. No matter which solution is implemented, a maintenance schedule with a quantified effort in time, skill and budget is recommended for realistic continuation. The maintenance effort ranges from low, for cloud solutions and standard middleware frameworks, to high for bespoke coding and self maintained servers.

4.4.3 Usage

The best automations require limited to no effort from the end user. An automation solves a problem to reduce costs in the end. An end user will not adopt an automation if the effort surpasses the benefit. The more intuitive the required effort for an end user, the lower the obstacle. Providing feedback to the user improves the acceptance, especially when privacy is a concern. A smart company project is successful when accepted by the end users. For example, a train conductor is required to give an input with his feet on a timed interval. This automation is designed to prevent a conductor to fall a sleep and limits the chance of accidents due to human errors. This trigger is accepted by the conductors as the required effort is minimal for the benefit the automation offers.

4.5 Training

Training has different aspects to consider. Open source middleware generally has online tutorials and instructions for getting started. Hardware suppliers often offer online or class room trainings. Frameworks like openHab facilitate installation and configuration instructions to encourage people to use their product. But not all open source platforms have good training per definition. Linksmart has a track record of successful installations in various industries. Learning the ins and outs of setting up and creating custom modules lacks documentation. Prior knowledge of the programming language is required, and the level of technical skills needs to be higher. A benefit of middleware solution is that it tends to make connections generic. If you know how to add one component, you know how to add all components. There is no need to know how to configure the individual components. This is provided that the individual components have been installed prior to linking to a middleware, or if they are able to be completely configured by the middleware connection. Integrating and interlinking different systems, often the case with software from hardware suppliers, requires training for all components. The required training for developers and administrators is the first aspect.

The second aspect is training the developer by the domain expert. Domain experts know the semantics about triggers and routines. They need to communicate these semantics to the developers, which can be error prone. A good approach is allowing the domain experts to set up automations themselves. When a domain expert has advanced developing skills, they can have both roles. For the other cases codeless programming is an asset. As CMT has shown, visual modelling tools offer an intuitive low obstacle alternative to writing codes. Testing and maintaining automations by the actors that hold the knowledge or even use the system impacts the setup and acceptance positively.

The end user requires training on how to use and interpret automations. Smart companies with a large employment list or with high rate of staff turnover, need a structured training approach for end users. Keeping user input minimal and providing feedback to the user helps successful implementations. The best automations are those that require no user input to work. Adopting an automation is easier when there is limited effort from the end user required. When user input is required, lowering the obstacles by making the effort smaller than the benefits helps the acceptance.

The last important consideration is how to ensure the knowledge of the smart company configuration stays within the company. Documentation and handover procedures on the setup and maintaining is required when using bespoke implementations. Finding the balance between training needs for developers and end users should be taken into account when selecting the solution. Visual design tools involve domain experts early in the setup. Limit the required user input to a minimum, or make input obsolete. Identify where training has the biggest impact and focus on simplifying the learning curve.

4.6 Budget

The required budget determines the financial sense of a smart company and is for most companies the decisive factor. The benefits of automation should in the end generate a return that is higher than the investment cost. Quantifying the return can be challenging. If a company decides to invest in a smart automation, for example to maximize production and usage of renewable energy, the return can be hidden in branding or marketing. Measuring this return is more difficult when the effect is hidden or noticeable over time. When talking to company owners, there is no question about interest or potential for a smart company implementation. In most cases the Return on Investment (ROI) allows an implementation. When the return is recurring, limiting the recurring costs in favour of the initial costs. It makes a project worth investing in. Outsourcing can help fixing the ongoing costs and facilitate calculating the total budget.

The required budget includes the purchase of the hardware, subscription fees for leased material and software, staffing of required skills, training of the end users and maintaining the implementation. The required budget is the sum of costs for each linked component and the requirement mentioned in this chapter. Every choice has an effect on the budget and the choice should make sense in regards to the budget. A company that invests in better lighting and HVAC management has no benefit in setting up a high redundancy system, flexibility for upscaling or fail safe systems if the required budget is higher the potential benefits.

4.7 Privacy

Companies in Europe need to comply with the General Data Protection Regulation (GDPR)[2]. Storing personal data requires explicit permission. At any given time, the employee may request an overview of the stored data and can request to delete their personal data. Smart companies that do not store personal data are not subjected to the GDPR. A recommended approach is Privacy by Design (PbD). PbD means nothing more than "data protection through technology design." Behind this is the thought that data protection in data processing procedures is best adhered to when it is already integrated in the technology when created. [43] A smart company that requires to store personal data, or expects to the need to store personal data in later phase, are better of choosing a framework that use a PbD approach. Also the geological location where data is stored is part of GDPR. For cloud solutions the location of the data centre is best verified.

4.8 Redundancy

Redundancy is the inclusion of extra components which are not strictly necessary to functioning, in case of failure in other components. Hardware failure or communication failure are bound to happen. Cloud hosting generally comes with service level agreements and uptime guarantees, but they still have downtime because of maintenance. An on-premise server is more prone to hardware failures at some point in time. Any interruption needs to be considered. A cloud implementation with load balancers and mirrored servers is useless if the internet connection is lost. In the example of the access control at the VUB, people could get stuck if the locks are no longer controllable. When an emergency is the root cause, the need for

[2]`https://eur-lex.europa.eu/eli/reg/2016/679/oj`

redundancy increases. Fail safe systems and mechanical solutions can be added to cover the risks. The more business critical the automation is, the better a backup system needs to be. An option is keep state via hardwired connections. When parts of the smart company fail, the remaining components could still work. Avoid cloud based solutions for business critical routines.

4.9 Responsiveness

The responsiveness can be a determining factor for final stages. A wired connection on a frequently used access door can be faster than an cloud AI system processing various inputs to calculate access or not. When an automation provides user feedback, delays can be frustrating or confusing. If the task is monitoring a business critical process, delays can be very costly. The bandwidth and processing power should follow the need for reaction time. An automation for reporting or aggregating data might take longer or can be executed in idle time or queued as a background task. Responsiveness has an influence on the acceptance by the end users. Smart company implementations can be technologically sound, if the end users have obstacles in using them, they risk being cancelled nevertheless. A well balanced choice on all aspects is key to success.

5

Industry Examples

5.1 In the Field

Smart offices are designed to automate and facilitate tasks. In the end, they lead to cost reduction for companies. With technology evolving, the possible implementations for a smart office does too. Over the years, research has shown many applications can be developed. However, there is no such thing as a generic smart office. Companies will adopt an implementation if it suits their specific needs. Developing an individual solution requires resources and strategy. Implementing a smart office is not a goal for the major part of companies. Eliminating mistakes, facilitating growth, minimising energy costs, maximising communication or even using technological advancement as a marketing tool are reasons to invest in making the office smart.

Although a lot of research has been done on the subject, a smart office is not commonly implemented. Interviews with business owners and process stakeholders show that the interest in automating is very much alive. There is however no one-size-fits-all solution. Smart office providers generally offer an implementation as a product. After a successful implementation, the product is adjusted to serve in a wider range. Meeting rooms are commonly offered with automations like smartboards, moods for lighting and temperature, and easy presentation solutions. As most companies have a meeting room of some kind, an automated solution can be sold in a generic way.

The question *what will it bring me?* or a benefit for implementation (often in the form of Return on Investment (ROI)) is the ultimate motivation for implementing a smart office. In this section we present four use cases where a smart office offers a solution. For those companies, a need is determined, and a possible solution proposed.

5.2 DIRAC Industries

5.2.1 Company

DIRAC Industries[1] is a multinational with offices in four different countries. It specialises in industrial electric heating and metalworks with high-quality products and a reliable service. DIRAC Industries has 8 business units. Worldwide DIRAC Industries employs 188 full time equivalents (FTE), with the majority working in the production facility located in the Czech Republic. The Belgium headquarters has 36 people on the payroll [44]. In 2018 the company implemented Oracle NetSuite[2] as the worldwide Enterprise Resource Planning (ERP) system.

5.2.2 Situation in the Belgian Offices

The office building has 10 flexible workspaces, each equipped with an IP-telephone. The sales people come in for project calculations and sales follow up several times a week. Given the fact that there are more people than desks, there is no fixed seating arrangement. The presence of a sales person creates a high increase of incoming phone calls from customers returning calls. When the sales executives choose a flexible office space, it is not clear where they are seated.

The phone calls arrive on a general number, answered by an office clerk. The clerk has the task of forwarding the calls to the right person, however, finding the correct phone on the correct workspace proves to be challenging. This results in long waiting times for customers on in bound calls and lost time in locating the correct person and forwarding. The more people are in, the more complicated. Currently there is a list at the desk of the office clerk. When

[1] `https://diracindustries.com`
[2] `https://www.netsuite.com`

a sales person has chosen a desk, he is supposed to inform the office clerk of the phone systems internal extension, for easy forwarding. More often than not, the list is not up to date. Least to say it is error prone and demanding a lot of effort and manual follow up. Every outbound call has the companies main number as caller identification is a company policy.

5.2.3 Solution for the Belgian Offices

The telephone system is a Voice over Internet Protocol (VoIP) system. All employees have their own internal extension. The VoIP server works with the TAPI-protocol, provides a specific software driver that interfaces directly with the hardware. The ERP system documents the Customer relationship management (CRM) actions. All the steps in the sales process are logged in the client histories, including all communication with customers. When a sales person logs in to NetSuite, the option to indicate current flexible desk number is presented. This number is displayed on each desk. When the option is filled in, their personal phone extension is logged out from other desks, if any, and linked to the VoIP phone at the specified desk. The office clerk has a digital overview of all the sales people in-house, with their personal extension. This solution eliminates delays in call forwarding, simplifies the procedure presence sharing and allows the flexible desks to be kept flexible. As an added bonus, the inbound calls are automatically logged in the CRM module of the ERP system, if the inbound caller id is known in the system.

The linking of an extension to a phone is programmed by a software developer. The link between a desk number and a media access control address (MAC address) is managed by a domain expert. An end user, a sales person in this case, can chose to use the system, just by entering a number. In practice the automation is used 90% of the time because it is low obstacle .

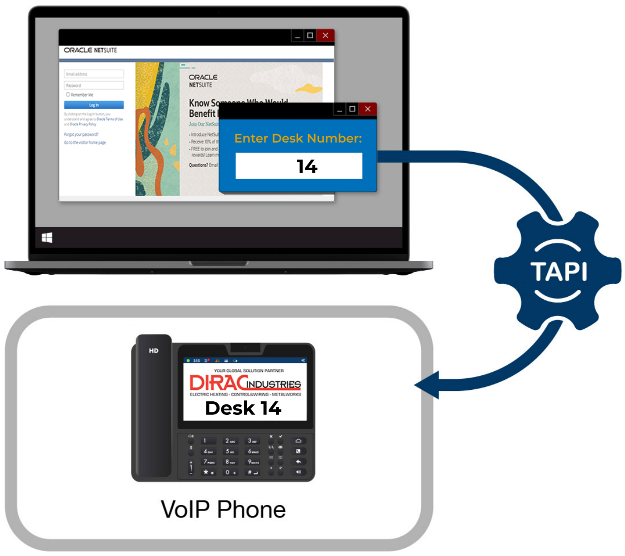


Figure 5.1: Linking NetSuite to a phone using the TAPI protocol

5.2.4 Situation in the Czech Manufacturing Facility

The production facility in the Czech Republic staffs around 100 blue-collar FTE. 90% of them are women, working on flexible half-time schedules. In total around 170 workers fill the FTE position on the assembly lines. During a shift, workers are moved around between different departments and assembly lines, depending on the products being produced and the staffing needs for those products.

Each employee has an access key card. This key card holds a Radio-frequency identification (RFID) chip and operates door locks in various areas, depending on access rights. A key card generally opens the entrance door, the door to the dressing room, the production hall and break area. The employees are required to hold the key card on them at all times.

With the large number of employees and flexible working schedules, keeping track of presence and time sheets, is a major task for the various level of management. On the one hand, the presence and time sheets are used for the payroll calculations. As there are many entries, a mistake is difficult to spot and delicate as it involves the wages of staff. On the other hand, to identify training opportunities and to manage the quality of produced goods, DIRAC Industries wants to log for each individual product who was part of the assembly line performing which task.

In the NetSuite implementation, the assembly lines and the working spaces on the assembly lines are uniquely identified. Each product has a unique identifier as well, tracking the progress of production orders in real time.

5.2.5 Solution for the Czech Manufacturing Facility

The Management of DIRAC Industries has investigated different approaches to automate time sheets for employees and linking a person to a produced product. There are two major software solutions; NetSuite, their ERP system, and the access control with RFID. Linking both systems to a separate middleware has been an option. This implies syncing two databases via APIs with the middleware. The middleware would run on the internal server, written and maintained by the company's in-house developer.

The alternative solution, sharing the needed information between the two systems, keeping them running independently, was the chosen implementation. By adding a key card reader to all the workstations, mapping the digital floor plan in NetSuite and linking the key card to an employee record, a smart office implementation was executed. The NetSuite program was extended with a service querying the access control system for relevant information.

The working times of an employee were calculated from passing specific checkpoints, like entrance to production hall and break areas. For every finished product, the list of employees and their corresponding workstations was logged on the work order. By passing a time range to the access control system, a list of people present on routing's workstations is generated.

5.3 VRT

5.3.1 Company

VRT[3] is the national public service broadcaster for the Flemish Community of Belgium. [45] VRT owns four televisionchannels and five radiostations. They employed 2.046 FTE in 2019 of which 15.3% are statutory and 84,7% contractors. [46] The statutory employees have a mobility budget as part of their wages. This budget can be claimed in the form of a monthly bonus, a public transport voucher, the lease of an electric bike or a company car. In 2020, 230 employees chose a company car, 12% hybrid and 15% full electric.

[3] `https://www.vrt.be`

5.3.2 Situation

With a rising number of vehicles that need to be charged, the available on-site parking with charging possibilities brings extra challenges. The cars need to be charged regularly, within office hours. When a car is finished charging, the space needs to be vacated in order for a new car to be charged. The rotation of cars on dedicated parking spaces with charging capabilities needs to be managed carefully to limit time of searching car owner and asking them to move.

Depending on whether a car belongs to a statutory or a contractor, the charge is free of charge or invoiced. The charging stations come with a RFID badge. Only allowed badges can charge. After a full charge, locating the driver of a vehicle can be a quest. In theory all the cars can be charged on site, but in practise the chaos of locating drivers and rotating cars proves to be a challenge.

5.3.3 Solution

As the entrance to the parking lot requires the registration of the license numbers, the identity of drivers is known in the personnel files. An employee has their employment status, phone number and license number stored. The charging stations can be extended with a camera, registering if a parking space is occupied and which license plate is parked. When a car is connected to the charging station, the charging management system can check with the personnel files if a car is allowed to be charged, and if it should be invoiced or not. There is no need for badging or manual identification.

When a car is charged (no more current is flowing) the owner can be informed and asked to move the vehicle to make way for the next car. Once the car is physically moved, the driver of a car that was not charged during the day can be notified. There are several possibilities of knowing which cars are queueing for a charge. A simple web form on the internal software system would allow an employee to queue with his personal login details.

5.4 Off Grid bv

5.4.1 Company

Off Grid[4] specialises in renewable energy projects, focused around battery storage, for both residential and industrial clients. [47]

The main activities of Off Grid include, but are not limited to:

- installing and maintaining solar panels on domestic and commercial buildings
- installing and maintaining wind turbines as an additional source of energy
- installing and maintaining charging stations for electrical vehicles
- designing, installing and maintaining cooling and HVAC systems
- designing, installing and maintaining *off grid* solutions for complete disconnection from the electricity grid

[4] `https://www.offgrid.expert`

5.4.2 Situation

For the majority of renewable energy projects, the aim is to be as close as possible to full self consumption of the generated energy. Producing the required energy on the moment of consumption, and consuming the generated energy is the challenge of sustainable energy. A storage system like batteries buffers the difference between generation and consumption. The more energy is directly consumed, the fewer energy needs to be stored or buffered. The renewable energy system has a lot of functionality out of the box to charge and discharge the battery. To limit the purchase of external power, the total consumed energy should not exceed the total generation at any given time. This could lead to injecting a surplus at a later moment.

5.4.3 Solution

A smart office can steer the direct consumption by activating a consumer when there is more supply than demand. This can be achieved by switching outlets, or linking appliances to an automation system, like NHC. A charging station can be switched on when there is energy to spare. Charging stations typically charge with a maximum power of 7,4kW or 22kW, depending on the number of phases connected. If there is an excess of less than the maximum power, the charging station should be modulated to avoid the purchase of external energy.

NHC can be connected to the energy management system (EMS) of SMA[5]. The EMS measures in which direction the electrical current flow. A negative flow indicates the cuurent is flowing to the grid, meaning there is an injecting or in other words, there is more production than consumption. If current is injected into the grid, the EMS sets the state of a virtual output in NHC to *true*. Depending on the total consumption, the charging station can be switched on with the value of the charging power being altered every 5 seconds, if there is a surplus and there is a car connected. This can be achieved by linking the charging station to NHC.

[5] https://www.sma-benelux.com

6
Implementation

6.1 Smart Company

As a proof of concept, we implement the industry example described in Section 5.4.3, together with some common scenarios in a smart company. The automation described in this chapter is installed in an office building with a warehouse attached. The office is 150m^2 and the warehouse 200m^2. On the roof of the building there are 48 solar panels installed, producing up to 100 kWh a day. A battery storage system with the capacity of 24 kWh buffers the generated energy to maximise the self consumption of own energy. The average daily electricity consumption, except charging of electrical vehicles is 33 kWh.

Four people work in the office. Three of them drive to work using a speed pedelec bike. The last employee has a full electric car. As the 4 vehicles are company owned, the charging is done on-site. The bikes use indoor chargers connected to dedicated switched power outlets. The car uses an outdoor charging station, installed near the parking spot. Part of the office setup is a showroom for customers. During opening hours, customers have free access to the showroom, located near the front entrance door. The smart office has been installed and has been running for eight months to date.

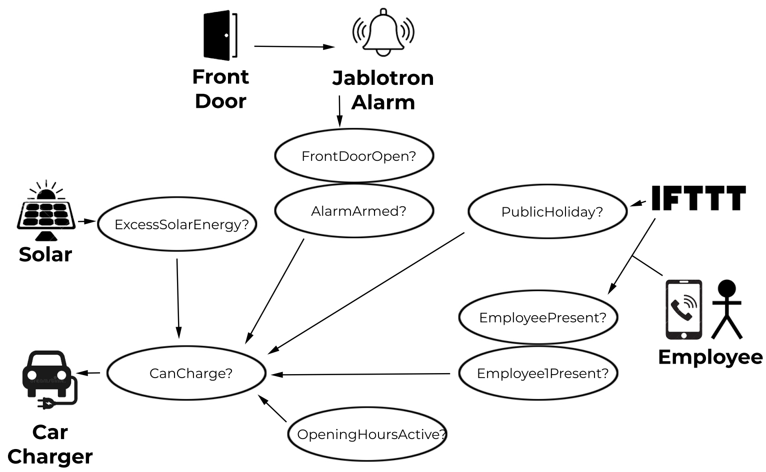


Figure 6.1: Virtual outputs set by different IoT components

In this chapter we describe the different systems used in the smart office and we present scenarios that are implemented in at least two independent systems. The smart office has numerous additional automations, only actions that are relevant to the subject of this thesis have been included. For every scenario we will indicate which system is used and how the information is shared. All the independent systems have their own smartphone app and can be controlled from outside the office building.

6.1.1 Applying the Guidelines

Scalability

At the time of the analysis, the office building just started renovations. The hardware components have been selected with automation in mind. The advantage of starting in an early phase of construction, is that compatibility between components can be mandatory. When considering suppliers for lights, doors, HVAC, alarm and renewable energy, external communication possibility and protocols are taken into account. The office and warehouse are physically limited by their dimensions. For this implementation, the choice was made to limit the smart company to the initial hardware triggers. The required flexibility was the option to add IoT components that use the same communication as base components. The different components for this smart company use modern techniques for cloud communication, allowing a high level of flexibility.

Security

Access to the building can be a major security risk. To ensure the alarm system and door access are protected from external attacks, the individual systems run independently. The different components are configured to share context on need to know basis. For this implementation, this was the decisive requirement to choose an alternative to middleware. Hard wired connections are used in combination with software triggers to allow multiple checks for similar routines. Cloud connections are always verified by the state of electrical contactors and relays. This implementation could have been done using a context broker middleware framework. The redundancy, effort and budget supported the choice. The access of employees can be revoked at any given time by the administrator by blocking the access to the main component (see Section 6.2).

Effort

Creating, expanding of debugging actions are to be done by domain experts. The coding is to be kept to a minimum.

Maintenance

Every 3 months the administrator needs to validate new releases for each provider. There was an explicit desire to not use servers that required maintenance. Also outsourcing would have a big effect on budget.

Usage

The goal is the create a smart company where no user action, apart from setup, is required for day to day work. Removing small repetitive tasks, the employees can focus on the content on the their assignments.

Training

Each hardware component requires a training from the hardware supplier. The administrator, a domain expert in this case, needed a total of five days class room training, organized by the different suppliers. The required skills for the training sessions are technical profiles, but no software engineers. The benefit is the possibility to train new administrators by the hardware supplier, even without hand over. The company owner valued the continuity and access to installation knowledge as top priority. Cloud based and IoT components required a few days of online tutorials.

There is low staff turnover. The training on how to use smart company has been integrated in the onboarding procedure and is estimated to take 4 hours to set up. The termination procedure also includes a smart company procedure s part of the privacy requirements.

Budget

The owner wanted low recurring costs, especially in the form of subscriptions and maintenance. The initial costs include mainly training, implementation and hardware costs. As the company was doing renovations, the hardware did not have to be purchased seperatley, but as base of the building.

Privacy

The employees have the choice to take part in the smart company or not. To comply with the GDPR regulations, the stored personal data has to be listed during onboarding, together with the procedure of deleting all personal data. A very important requirement was the transparency of stored data and the possibility to delete them by the employees themselves. The option was linking data that could be seen as privacy concerns to the mobile phones of the employees. An individual user profile holds the only data, and can be removed easily.

Redundancy

The alarm system and light control require redundancy in case of network or grid failures. The entire building runs with backup power, and each individual system has their Uninterruptible Power Supply (UPS) When network fails, the hardwired connections control states in the system. The hardware has manual overrides in case of emergency. Especially the front door needs to be opened manually in case of fire.

Responsiveness

Lights, alarm and door access required feedback and fast reaction. The other automations are seen as background tasks and only need to provide feedback.

6.1.2 Users

Electrical Engineer

In our smart company implementation, an electrical engineer is needed for the installation of the physical components. The inputs and outputs of the NHC installation are connected and configured with unique addresses for every component. These addresses allow each component to be used in automated actions. The Jablotron hardware is also installed by the electrical engineer. This engineer needs to be accredited by the Ministry of Internal Affairs, as alarm system installer is a protected profession. In our case, we used an electrician with the required documents.

Between Jablotron and NHC there is a physical wired connection. In both systems there are mirrored connectors sharing the status of virtual outputs. The charging station also has a wired connection to the NHC system, indicating if charging is allowed, as explained in the Section 6.3.

Developer and Domain Expert

The digital configuration of each linked system requires basic development skills. In our implementation, we used a dedicated domain expert for the role of developer. Only configuring the software with the addresses of the physical inputs and outputs, defining virtual outputs and creating the rules for the automation is needed as part of the development. The visual design software allows programming with the need of writing code. The maintenance of each independent system will also be handled by this person.

There is a learning curve getting to know each system. Both suppliers of NHC and Jablotron offer a two day training for connecting, configuring, administrating and maintaining their systems. All the systems in our smart office implementation offer online tutorials and a large knowledge base for implementation and problem solving. The tasks of the developer include configuring the system, creating and deleting a user profile for every employee on the company, and maintaining the different systems.

End Users

The end users in this case are the employees of the company. They each have a personal profile in NHC and can adjust their own parameters. Theoretically they can decide themselves how many personal information they wish to share. When an employee decides to leave the company, they can erase all personal data from the smart office, making the implementation compliant with GDPR[1]. For the purpose of this thesis, we were given the agreement of each employee to use their personal parameters in light of the smart office implementation. The data has not been shared and is made anonymous.

[1] https://ec.europa.eu/info/law/law-topic/data-protection_en

6.2 Niko Home Control (NHC)

NHC is a state-of-the-art automation solution [48]. The system offers central control, cloud control, monitoring of energy consumption and automatic adjustment of lighting, HVAC and switched sockets. NHC is shipped with visual programming software[2] where installers can program actions based on events. The software is user-friendly and has a visual design interface to make programming intuitive and structured. More advanced configurations can be programmed by using a wizard. Before any change is uploaded to the main controller, the software will automatically scan the installation to check whether an upgrade is required. The administrator is informed when an update is available.

In the Niko Home Control programming software, multiple user profiles can be defined. Each profile has unique access rights defining the level of control over the system. End users can use the same software as the developer to personalise their installation. Certain parameters, like the desired room temperature, can be changed within their profile.

Niko nv, the manufacurer of NHC, has an partnership as service provider of IFTTT. End users can link their profile from NHC to IFTTT, creating personal automation if desired. The user, employees in this case, can setup feedback from automated routines. In our implementation we use the IFTTT to inform NHC when the employee arrives at the office.

SMA (see Section 6.3) has an integration with the main controller of the NHC via the modbus protocol. Knowledge about the renewable energy production and consumption is shared by the SMA smart energy meter.

[2] `https://www.niko.eu/en/products/niko-home-control/software-and-apps`

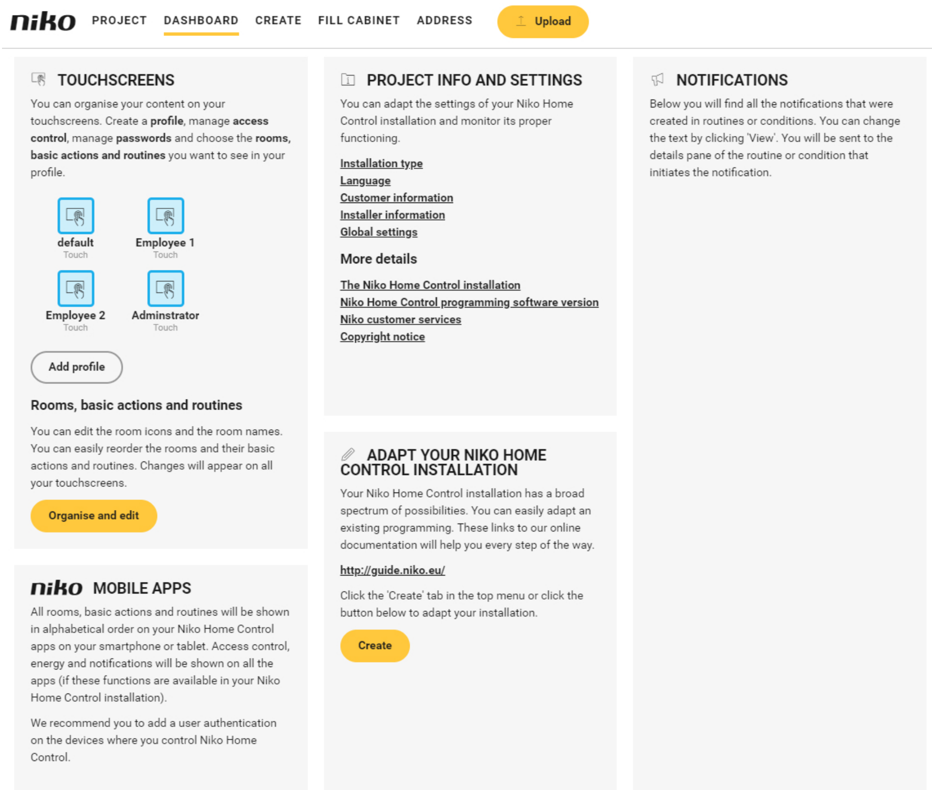


Figure 6.2: The main screen of the NHC visual design software

NHC allows communication with external systems via the MQTT protocol. An IoT device can be integrated on local network connection, opening the door to many external products. In our implementation we use only market available products.

6.2.1 Input and Output

Every light and power outlet in the smart office is connected to NHC. They are considered outputs and the can be set, unset, locked or unlocked via *actions*. Every output has an unique address in the configuration. The state of an output can be changed by linked services like IFTTT[3]. In this implementation we opted to implement virtual outputs as an abstraction

[3] `https://ifttt.com/nikohomecontrol`

layer between NHC and external services. This provides more control over complicated actions, and allows continuing of the automation if the connection to the external service is lost.

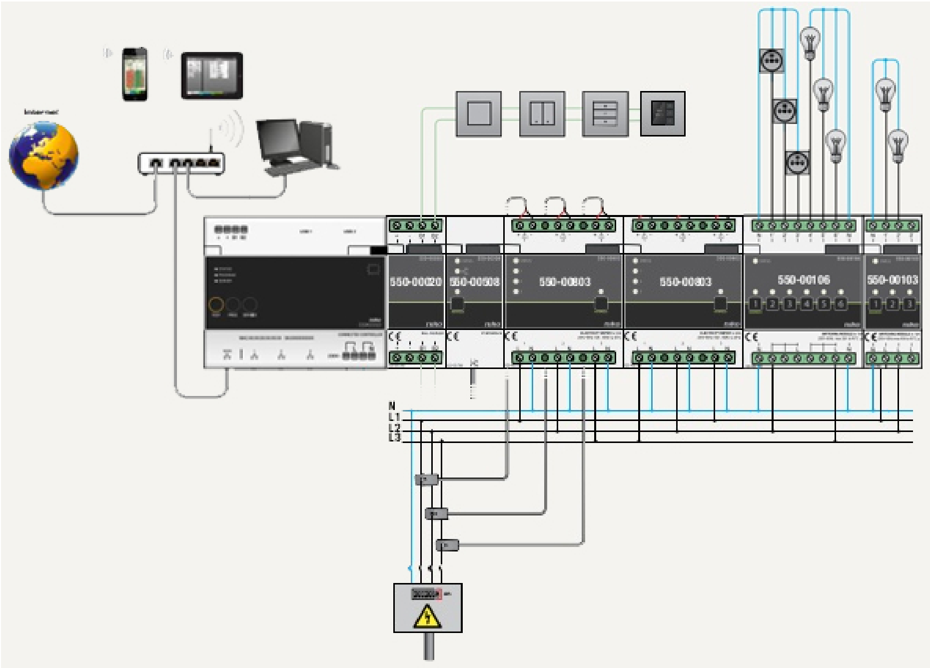


Figure 6.3: Wiring example of a NHC installation

An electrical lock of type FUHR multitronic 881[4] controls the front door. This lock is connected as an output on NHC. Opening and closing takes place electronically. As soon as the door closes, the lock engages and locks automatically. When the output is activated, is keeps the lock opened, allowing free access. The door does not have an outside handle. Customers can use the free access to enter the showroom when unlocked. This setting is used in one of the scenarios of the smart office implementation.

Lights are used in groups in predefined modi. The advertisement above the entrance on the outside door, the front window lights and some products are lit in what is called *night modus*. When the office is closed, the night modus is activated. The counterpart is called *day modus* and is activated during

[4] `https://www.fuhr.de/fuhr/en/produkte/prod_tuer_881.html`

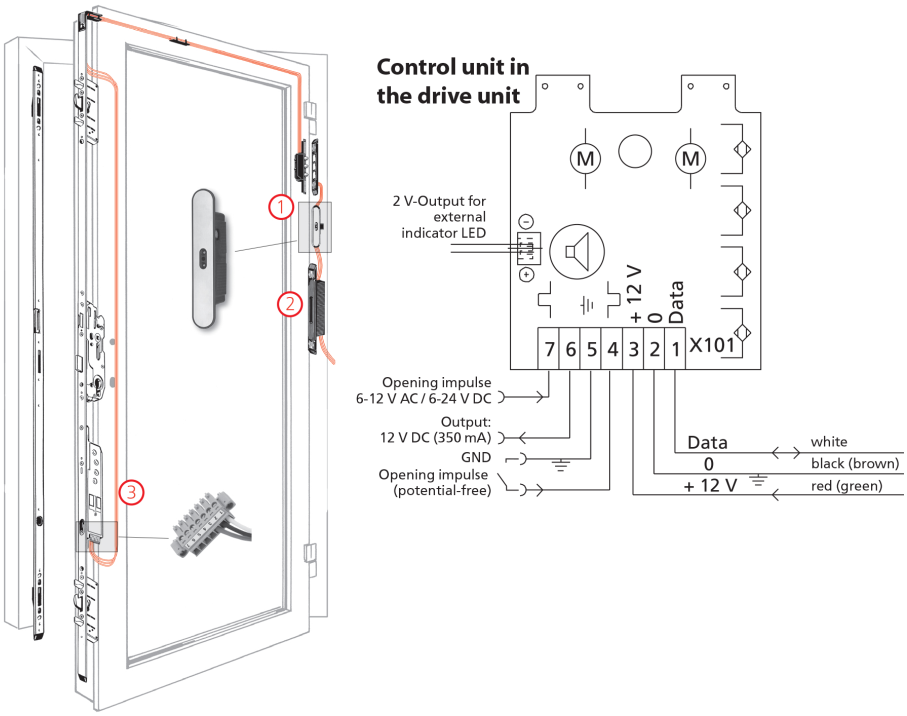


Figure 6.4: Wiring diagram of the FUHR multitronic 881

office hours. At any given time, there is only the day or night modus active. Near the entrance door a red light is installed to act as a feedback for status of the alarm.

Warehouse lights, hallways and toilets are not part of these two modi. They are controlled by motion sensors, and follow logic that aims to control energy consumption. The day modus automatically adjust the lumen of the lights depending on the brightness outside. Each employee can manually overwrite the lumen using their personal login in the NHC app.

The pedelec bike chargers use dedicated power outlets. Together with the wired connection to the SMA charging station, they are setup as outputs as part of a scenario in the smart office implementation.

6.2.2 Virtual Outputs

Virtual outputs can be defined in NHC. As a layer of abstraction, they are used in this implementation to create complex actions and to link other systems in the smart office. The output can be set directly from IFTTT and they can be programmed the follow the status of a contactor, like the Jablotron and SMA links.

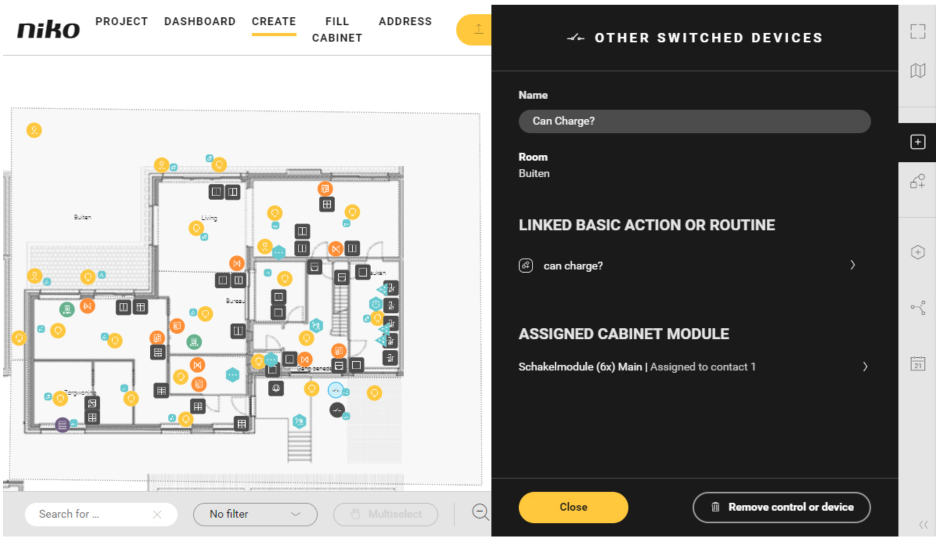


Figure 6.5: Defining a virtual input in NHC

In this implementation the following virtual outputs have been defined:

- *OpeningHoursActive?*: a time schedule activates the output on weekdays from 9:00 to 12:00 and from 13:00 to 17:00. The office hours do not take public holidays into account. The time schedule is defined in NHC.
- *PublicHoliday?*: IFTTT is linked to a google calendar with public holidays. When the calendar has an entry, this output is set to true. Together with the *Opening hours active?* output it is used in the routine to put the front door in free access mode.

- *FrontDoorOpen?*: The front door has a magnetic contact as part of the Jablotron alarm system. If the front door is opened, the state is mirrored on this output.
- *GateOpen?*: Similar to the previous one, this output mirrors the state of the magnetic contact of the warehouse gate in the Jablotron system.
- *ExcessSolarEnergy?*: When there is electricity being injected into the grid, this output is set to true. This indicates additional consumers can be switched.
- *CanCharge?*: The charging station can only be used during office hours, when an employee is present. The wired connection with the SMA charging station is steered by this output.
- *Employee 1 ... 4Present?*: Every employee has their own virtual output, indicating their presence. If their smartphone is connected to the office network, IFTTT sets the output for the employee.
- *EmployeePresent?*: If any employee is present this output is set.
- *AlarmArmed?*: A wired connection to the Jablotron alarm system is used to monitor whether the alarm system is armed. If the contactor is closed, the virtual output is set to true.

6.3 SMA

SMA Solar Technology AG is a German solar energy equipment supplier founded in 1981 and headquartered in Niestetal, Northern Hesse, Germany. SMA is a producer and manufacturer of solar inverters for photovoltaic systems with grid connection, off-grid power supply and backup operations.

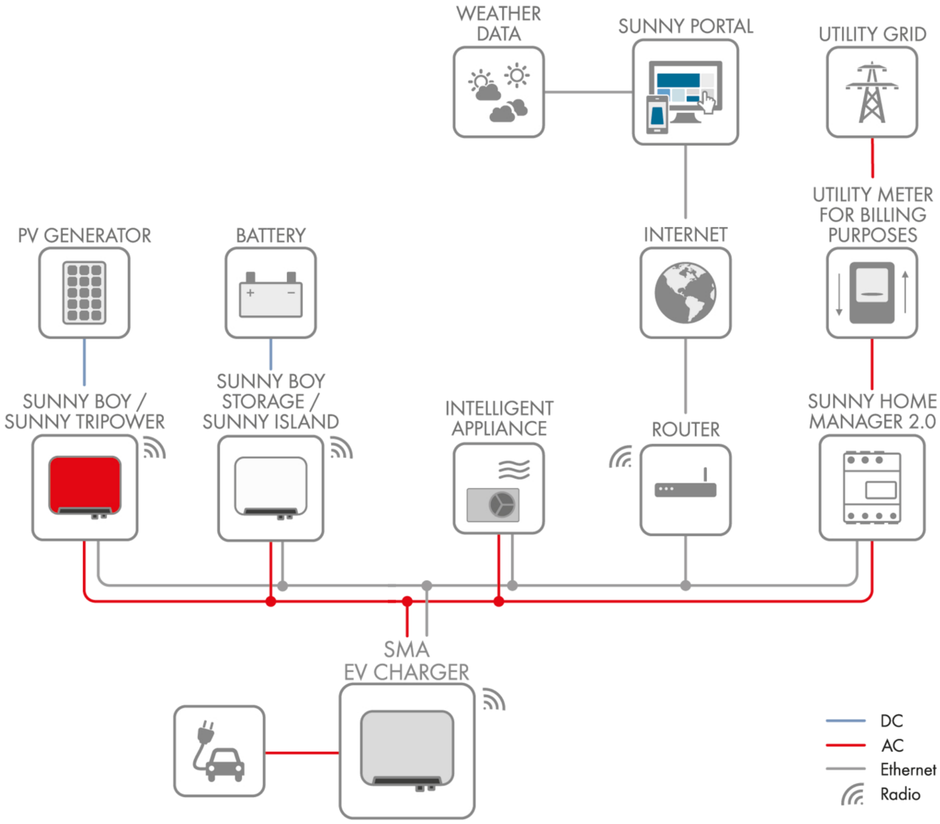


Figure 6.6: Renewable energy system configuration with SMA [7]

SMA is the supplier of the inverters for both the solar panels and the batteries, the charging station and the EMS. The software has an Artificial Intelligence (AI) element, used to forecast energy consumption and production. Historic consumption data from the EMS is processed to forecast the energy consumption. The production of energy is forecasted for the three upcoming days using weather forecasts, installed hardware characteristics and nearby installations. The EMS is managed by a smart controller called *Sunny Home Manager 2.0*. This controller holds the information about solar production, energy consumption and state of the battery storage. The software dictates the direction of the flow of energy. The communication with NHC is done via the modbus protocol.

There is also a wired connection between NHC and the charging station. When there is no employee present in the building, or the current time is outside opening hours, the charging station is not allowed to be used. The digital input of the charging station follows the state of a virtual output in NHC called *CanCharge?*, created in Section 6.2.2.

6.4 IFTTT

IFTTT is a service that allows a user to program a response to events in the world. IFTTT has partnerships with different service providers that supply event notifications to IFTTT and execute commands that implement the responses. Some event and command interfaces are simply public APIs. Users can define simple rules via so-called recipes. A recipe consists of a trigger and an action. Recipes, triggers and actions can be shared among users [49].

IFTTT is used in our implementation to automate user specific action and to act as a bridge between NHC and iRobot. IFTTT is installed as an app on Android smartphones. Every employee has the option to link their phone and take part in the automation. By linking the smart phone, meaningful personal information is made available. When an employee is close to the office, and connects to the WiFi network, the personal virtual output in NHC is activated, allowing execution of presence based routines. The same information is shared when an employee disarms the alarm system. The Jablotron smart hub forwards the alarm messages to IFTTT to be processed. An employee can allow feedback on activated routines in the form of push notification, text messages, emails or sounds on the Android smart phone.

The communication between NHC and iRobot is bridged via IFTTT. After the last employee has left the building and the opening hours have passed, the vacuum cleaner is started.

6.5 Jablotron

Jablotron Alarms[5] is part of the Jablotron Group and a leading European player in the field of alarm systems. The company was founded in the Czech Republic in 1990 and has grown dynamically since. Jablotron produces alarm systems with bus communication. The functioning of the system is completely digital and can be monitored and controlled via the Jablotron smart phone app.

Because of legal requirements, the alarm can not be controlled via a public API. There is however a hardware module called *Jablotron smart hub*, that sends messages to IFTTT. These messages are custom and can be linked to changes in the alarm system. Arming and disarming the alarm will send a message, including who executed the change to IFTTT. Upon receipt, they will be interpreted as triggers in IFTTT recipes.

A wired connection to a hardware contactor in the Jablotron central, as opposed to API, can be used for changing the state of the alarm system. In our implementation, NHC is connected via wires to such a Jablotron contactor. The benefit of a physical connection is the ability to share the state of the alarm in case of network or electrical failure. The virtual output in NHC can be initialised with the state of the contactor. When the contactor is closed, the virtual output is set to *true*.

Jablotron is a closed system. Any integration in a smart office should respect the integrity of the security features. In our solution alarm can be set and unset only via a wired connection, when NHC is online. Additionally a wired contactor is used to indicate if the front door and warehouse gate are open. This information could also be monitored by NHC, but as the door and gate are part of the alarm perimeter, we implemented passing of information.

[5] `https://www.jablotron.com/en`

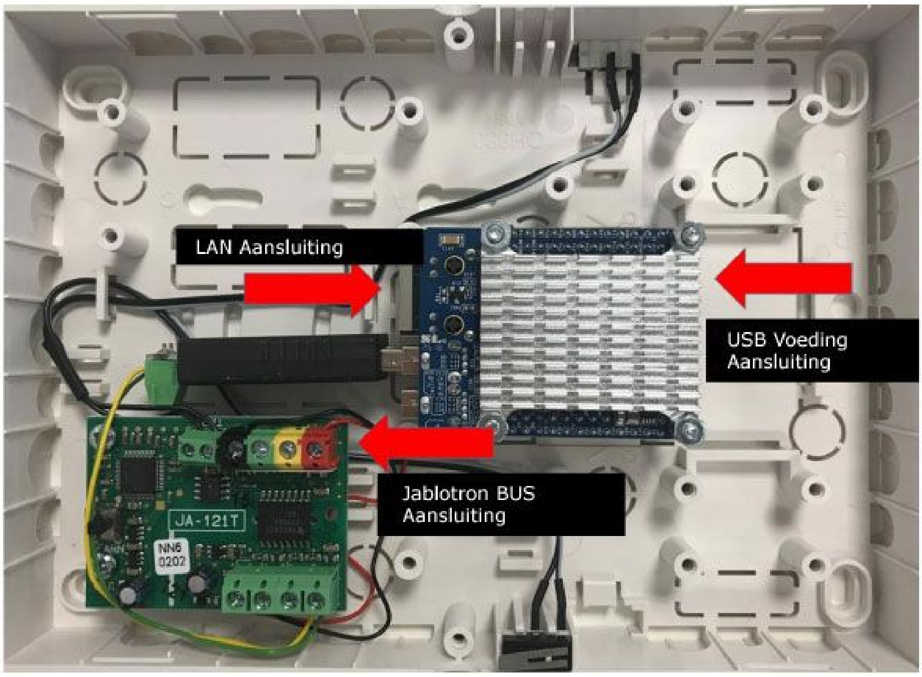


Figure 6.7: The Jablotron smart-hub component

Every employee has its own alarm code. Any activity made by an employee is uniquely identified and shared with IFTTT as a trigger. This is an alternative way how employee specific automation can be created.

6.6 iRobot

iRobot® is a supplier of vacuum robots. The Roomba® i7+ robot vacuum comes with automatic dirt disposal. During cleaning, it goes back to its base for automatically emptying the dirt container into an enclosed bag that can hold up to 30 robot bins. The vacuum robot learns, maps, and adapts to the office using smart mapping technology. The robot comes with a app, controlling which rooms are cleaned and when. iRobot can be started, stopped and paused via IFTTT[6] over WiFi.

[6] `https://ifttt.com/irobot`

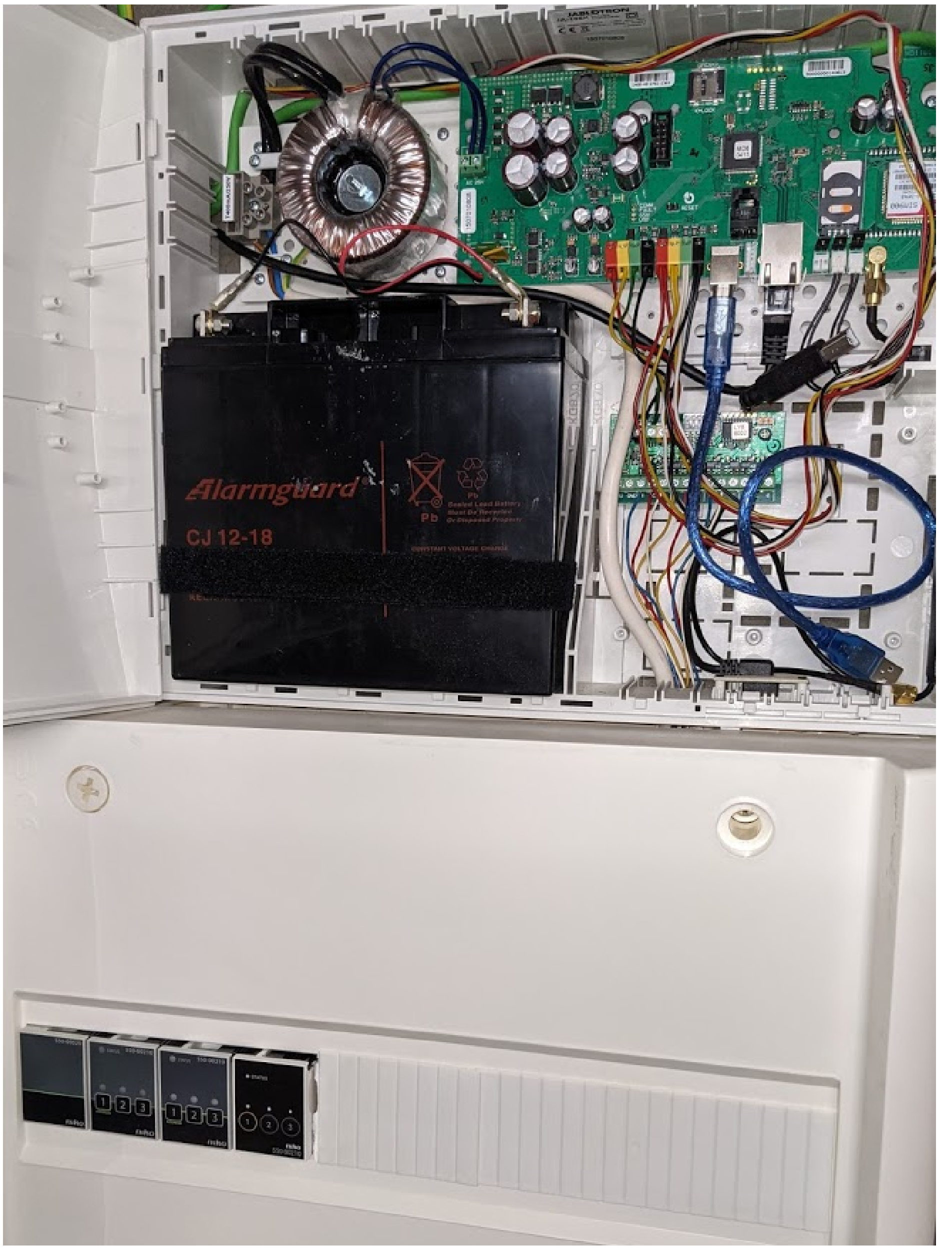


Figure 6.8: The wired connection between NHC and Jablotron

6.7 The Smart Office Scenarios

6.7.1 Scenario 1

The employees drive to work by speed pedelec bikes or electrical car. As part of the mobility policy, the batteries of the vehicles can only be charged from renewable energy. The electrical car has an outdoor charging station, without access control. The charging station is part of the SMA Solar Technology's EMS and knows when and how much energy of the solar production is available. The charging station should only be able to charge during office hours and modulate to the available current. The modulation is a feature in the charging station software. A wired connection between the digital input on the charging station and NHC allows charging when the contactor is closed. The charging station has a green LED indicating when active.

The pedelec bikes each have their charger in the warehouse. Those chargers are indoors, and plugged in to dedicated switched power outlets. The power outlets, toghether with a feedback LED above the outlets, are switched on if there is excess renewable energy.

Require: SMA
  if Energy injected in grid **then**
    Change $ModbusInjectionRegister$
  end if
Require: $IFTTT$
  if Entry In Public Holiday Calendar **then**
    Set $NHC\_PublicHoliday? \rightarrow true$
  end if
Require: NHC
  if SMA_Modbus Injection Register > 0 **then**
    Set $ExcessSolaEnergy? \rightarrow true$

end if
if WeekdayBetween9And17 **then**
  Set $OpeningHoursActive?$ $\rightarrow$ $true$
end if
if $OpeningHoursActive?$ and not($PublicHoliday?$) **then**
  Set $CanCharge?$ $\rightarrow$ $true$
end if
if $EmployeePresent?$ and $ExcessSolaEnergy?$ **then**
  Set $PedelecSocket?$ $\rightarrow$ $true$
  Set $FeedBackPedelecSocket?$ $\rightarrow$ $true$
else
  Set $PedelecSocket?$ $\rightarrow$ $false$
  Set $FeedBackPedelecSocket?$ $\rightarrow$ $false$
end if
if $CanCharge?$ **then**
  Close $ChargingStationContactor$
  Set $FeedBackChargingStation$ $\rightarrow$ $true$
else
  Open $ChargingStationContactor$
  Set $FeedBackChargingStation$ $\rightarrow$ $false$
end if

6.7.2 Scenario 2

The office building has a main entrance door, which is locked outside working hours and on public holidays. The door does not have a handle on the outside. During office hours, when there is an employee present, the door should be unlocked, and be able to be pushed open by visiting customers. At that start of a working day, the alarm has to be disarmed. The lights modus needs to be changed from night mode to day mode. After the last employee leaves the office, these actions need to be inverted.

Require: $NHC, IFTTT, Jablotron$

if $OpeningHoursActive?$ and not($PublicHoliday?$) and $EmployeePresent?$ and not($AlarmArmed?$) **then**

Set $OpenFrontDoor \rightarrow true$

Set $LightMood \rightarrow DayMode$

else

Set $OpenFrontDoor \rightarrow false$

Set $LightMood \rightarrow NightMode$

end if

6.7.3 Scenario 3

The office is the public meeting point for customers. As a representation of the project upon handover, the office is vacuumed every day. A vacuum robot is purchased to limit costs for external cleaning service and time used from employees. The office needs to be vacuumed every day, but the vacuum cleaner cannot operate when there are people in the office. The noise of the robot can be distracting during phone calls of customer presence. Therefore the cleaning can start after the last employee leaves the building.

Require: $NHC, IFTTT, iRobot$

if not($employeepresent?$) and $AlarmArmed?$ **then**

Send $StartCleaningInstruction \rightarrow iRobot(viaIFTTT)$

end if

6.7.4 Scenario 4

As the employees can be anywhere in the building, they require a notification when a customer enters through the entrance door. A classic ringer is insufficient if the sound can not be heard by the employee. As an office rule, every person that enters the office should be addressed within two minutes. To maximize the attention on someone entering through the front door, there are multiple means of informing the employees. There is a soft ring in the office and the lights in the warehouse flicker twice to indicate a door opening. If desired, employees can receive a push notification on their smartphone and is achieved by activating a trigger in IFTTT that they can use in an action.

Require: $NHC, IFTTT$
  if *employeepresent*? and not($AlarmArmed$?) and $FrontDoorOpen$? **then**
    Run $WarehouseLightFlicker$
    Run $NotificationSound$
    Send $FrontDoorOpened \rightarrow IFTTT$
  end if

6.8 Evaluation

For the implementation the choice was made to work with software from a hardware supplier. A middleware solution was an option as well. The final decision was based on redundancy, scalability and effort. The redundancy for network failure influenced the chosen solution. As the different components can work independently and have hardwired connections, the functionality that is not cloud based remains operational when a network failure occurs. The implementation did not require to be scaled beyond the limits of the NHC computing components. If higher flexibility was required, the selected setup would not have sufficed. The required effort in maintenance is minimal and can be planned. Most of the programming is done by domain experts because of the available visual design tools.

Even though similar smart company implementations can be made using middleware solutions, following the design guidelines allowed the selection of a suitable solution. On the present date, the automation runs in an office and warehouse environment and is a proof of concept of the design guidelines proposed in this document.

Following the guidelines, more than one potential solution could be selected. The final choice was arbitrary and cannot be compared to another implementation of the same smart office. Thus we cannot conclude that chosen implementation is the best for the given requirements. The only conclusion we can make is that the requirements have been fulfilled based on the considerations of the design guidelines presented in Chapter 4. Choosing a solution can be made less arbitrary by creating a list with possible solutions, sorted by *best match* for the requirements. This can be achieved by eliminating or scoring different solutions per requirements. A first approach can be to build a decision tree where every guideline is a node leading to one best solution. This requires elimination of potential solutions after every decision. To make the result more decisive and allow multiple options, each solution can be scored. A framework receives a score for every aspect. If an aspect can not be accommodated by a framework or solution, it is removed from the sorted option list. The guidelines can provide a weight to requirements, making some frameworks a better match than others.

7

Conclusion and Future Work

The main objective of this thesis was to create design guidelines to help companies defining requirements for a smart company implementation. Research done in the field of smart office focusses on improving approaches and specific implementations. There are numerous existing frameworks available for smart company automations. Selecting the best approach or framework for an automation proofs to be a challenge for company owners, as there is no best solution, only a better solution for defined requirements. To help creating these requirements, we have created design guidelines on different aspects of a smart company implementation.

After talking to company owners about scenarios where a smart company implementation can bring added value, we applied the design guidelines to one of the discussed solutions. A best approach for the requirements is implemented and tested in the field. The design guideline helped prioritising and weighting the requirements and led to a successful smart company.

When the requirements are defined, they should help to implement the smart company. In practice we learned that different solutions could be used for a set of requirements. To make the result more decisive, a recommender system would be a proposed future work. A recommender system is an information filtering system that seeks to predict the *rating* or *preference* a user would give to an item [50]. For smart companies they could generate a sorted list of best matching solutions for requirements. The guidelines presented in this paper can be used as criteria for which requirements need to be defined.

Bibliography

[1] W. Trumler, F. Bagci, J. Petzold, and T. Ungerer, "Smart doorplate," *Personal and Ubiquitous Computing*, vol. 7, no. 3-4, pp. 221–226, 2003.

[2] K. Furdik, G. Lukac, T. Sabol, and P. Kostelnik, "The network architecture designed for an adaptable iot-based smart office solution," *International Journal of Computer Networks and Communications Security*, vol. 1, no. 6, pp. 216–224, 2013.

[3] A. Alberdi, A. Aztiria, A. Basarab, and D. J. Cook, "Using smart offices to predict occupational stress," *International Journal of Industrial Ergonomics*, vol. 67, pp. 13–26, 2018.

[4] H. Chen, T. Finin, and A. Joshi, "An Intelligent Broker for Context-Aware Systems," *Adjunct Proceedings of UbiComp*, pp. 183–184, October 2003.

[5] S. Trullemans, L. Van Holsbeeke, and B. Signer, "The context modelling toolkit: A unified multi-layered context modelling approach," *Proceedings of the ACM on Human-Computer Interaction*, vol. 1, no. EICS, pp. 1–16, 2017.

[6] K. Furdík and G. Lukác, "Events processing and device interoperability in a smart office iot application," in *Central European Conference on*

Information and Intelligent Systems, p. 387, Faculty of Organization and Informatics Varazdin, 2012.

[7] . S. S. T. AG, “System configuration with SMA EV Charger.” `https://www.sma.de/en/products/e-mobility-charging-solutions/sma-ev-charger-74-22.html`, 2021. [Online retrieved on 25-May-2021].

[8] D. Minder, P. J. Marrón, A. Lachenmann, and K. Rothermel, “Experimental construction of a meeting model for smart office environments,” in *Proc. of the Workshop on Real-World Wireless Sensor Networks, SICS Technical Report*, p. 09, Citeseer, 2005.

[9] C. Le Gal, J. Martin, A. Lux, and J. L. Crowley, “Smart office: Design of an intelligent environment,” *IEEE Intelligent Systems*, no. 4, pp. 60–66, 2001.

[10] J. Petzold, F. Bagci, W. Trumler, T. Ungerer, and L. Vintan, “Global state context prediction techniques applied to a smart office building,” in *The communication networks and distributed systems modeling and simulation conference*, Citeseer, 2004.

[11] A. R. Hagale, J. E. Kelley, and R. Rozich, “RFID Smart Office Chair,” November 2005. US Patent 6,964,370.

[12] F. Bagci, H. Schick, J. Petzold, W. Trumler, and T. Ungerer, “The reflective mobile agent paradigm implemented in a smart office environment,” *Personal and Ubiquitous Computing*, vol. 11, no. 1, pp. 11–19, 2006.

[13] O. Vidal, “Smart office: wireless sensor network for energy monitoring and user profiling,” Kungliga Tekniska Hőgskolan, Stockholm - Master Thesis, 2009.

[14] A. Akbar, M. Nati, F. Carrez, and K. Moessner, “Contextual occupancy detection for smart office by pattern recognition of

electricity consumption data," in *2015 IEEE international conference on communications (ICC)*, pp. 561–566, IEEE, 2015.

[15] M. Choi, W.-K. Park, and I. Lee, "Smart office energy management system using bluetooth low energy based beacons and a mobile app," in *2015 IEEE International Conference on Consumer Electronics (ICCE)*, pp. 501–502, IEEE, 2015.

[16] A. K. Townsend, *The Smart Office: Turning Your Company on Its Head.* Gila Press, January 1998. ISBN: 978-0965708104.

[17] K. Hänsel, "Wearable and ambient sensing for well-being and emotional awareness in the smart workplace," in *UbiComp Adjunct*, 2016.

[18] B. A. Akyol, A. W. Jackson, R. Krishnan, D. Mankins, C. Partridge, N. Shectman, and G. D. Troxel, "Smart Office Spaces," in *USENIX Workshop on Embedded Systems*, pp. 29–31, 1999.

[19] H. Chen, T. Finin, A. Joshi, *et al.*, "A context broker for building smart meeting rooms," in *Proceedings of the AAAI Symposium on Knowledge Representation and Ontology for Autonomous Systems Symposium, 2004 AAAI Spring Symposium*, pp. 53–60, 2004.

[20] A. Coronato, G. De Pietro, and M. Esposito, "A semantic context service for smart offices," in *2006 International Conference on Hybrid Information Technology*, vol. 2, pp. 391–399, IEEE, 2006.

[21] C. Ramos, G. Marreiros, R. Santos, and C. F. Freitas, "Smart offices and intelligent decision rooms," in *Handbook of Ambient Intelligence and Smart Environments*, pp. 851–880, Springer, 2010. ISBN: 978-0-387-93808-0.

[22] L. Roalter, M. Kranz, and A. Möller, "A middleware for intelligent environments and the internet of things," in *International Conference on Ubiquitous Intelligence and Computing*, pp. 267–281, Springer, 2010.

[23] I. Burbey and T. Martin, "Predicting future locations using prediction-by-partial-match," pp. 1–6, 01 2008. DOI: 10.1145/1410012.1410014.

[24] J. Petzold, F. Bagci, W. Trumler, and T. Ungerer, "Next location prediction within a smart office building," *Cognitive Science Research Paper-University of Sussex CSRP*, vol. 577, p. 69, 2005.

[25] Linksmart, "Free, open source IoT platform." `https://linksmart.eu/`, 2020. [Online retrieved on 30-October-2020].

[26] M. Ryu, J. Kim, and J. Yun, "Integrated Semantics Service Platform for the Internet of Things: A Case Study of a Smart Office," *Sensors*, vol. 15, no. 1, pp. 2137–2160, 2015. DOI: 10.3390/s150102137, ISSN: 1424-8220.

[27] Z. Yang, C. Wu, and Y. Liu, "Locating in fingerprint space: wireless indoor localization with little human intervention," in *Proceedings of the 18th annual international conference on Mobile computing and networking*, pp. 269–280, 2012.

[28] M. V. de Wynckel and O. D. Troyer, "Indoor navigation by centralized tracking," VUB - Master Thesis, 2018-2019.

[29] W. Bulten, A. C. Van Rossum, and W. F. Haselager, "Human slam, indoor localisation of devices and users," in *2016 IEEE First International Conference on Internet-of-Things Design and Implementation (IoTDI)*, pp. 211–222, IEEE, 2016.

[30] F. Mizoguchi, H. Nishiyama, H. Ohwada, and H. Hiraishi, "Next location prediction within a smart office building," *Artificial Intelligence*, vol. 114, no. 1-2, pp. 57–94, 1999. DOI: 10.1.1/jpb001.

[31] C. Le Gal, J. Martin, and G. Durand, "Smart office: an intelligent and interactive environment," in *Managing Interactions in Smart Environments*, pp. 104–113, Springer, 2000.

[32] C. Röcker, "Information Privacy in Smart Office Environments: A Cross-Cultural Study Analyzing the Willingness of Users to Share Context Information," in *ICCSA*, 2010.

[33] C. Röcker, "Services and Applications for Smart Office Environments - A Survey of State-of-the-Art Usage Scenarios," *World Academy of Science, Engineering and Technology*, vol. 61, January 2010.

[34] H. Shigeta, J. Nakase, Y. Tsunematsu, K. Kiyokawa, M. Hatanaka, K. Hosoda, M. Okada, Y. Ishihara, F. Ooshita, H. Kakugawa, *et al.*, "Implementation of a smart office system in an ambient environment," in *2012 IEEE Virtual Reality Workshops (VRW)*, pp. 1–2, IEEE, 2012.

[35] A. Horch, M. Kubach, H. Rossnagel, and U. Laufs, "Why Should Only Your Home Be Smart? - A Vision for the Office of Tomorrow," *2017 IEEE International Conference on Smart Cloud (SmartCloud)*, pp. 52–59, 2017.

[36] M. Choi, W.-K. Park, and I. Lee, "Smart office energy-saving service using bluetooth low energy beacons and smart plugs," in *2015 IEEE International Conference on Data Science and Data Intensive Systems*, pp. 247–251, IEEE, 2015.

[37] N. Stojanovic, D. Milenovic, Y. Xu, L. Stojanovic, D. Anicic, and R. Studer, "An intelligent event-driven approach for efficient energy consumption in commercial buildings: Smart office use case," in *Proceedings of the 5th ACM international conference on Distributed event-based system*, pp. 303–312, ACM, 2011.

[38] C. Rottondi, M. Duchon, D. Koss, A. Palamarciuc, A. Pití, G. Verticale, and B. Schätz, "An energy management service for the smart office," *Energies*, vol. 8, no. 10, pp. 11667–11684, 2015.

[39] C. Rottondi, M. Duchon, D. Koss, G. Verticale, and B. Schätz, "An energy management system for a smart office environment," in

2015 International Conference and Workshops on Networked Systems (NetSys), pp. 1–6, IEEE, 2015.

[40] X. Wang, T. Tjalkens, and J.-P. Linnartz, "Smart office lighting control using occupancy sensors," in *2017 IEEE 14th International Conference on Networking, Sensing and Control (ICNSC)*, pp. 453–458, IEEE, 2017.

[41] H. Chen, T. Finin, and A. Joshi, "Semantic Web in the context broker architecture," in *Second IEEE Annual Conference on Pervasive Computing and Communications, 2004. Proceedings of the*, pp. 277–286, 2004. DOI: 10.1109/PERCOM.2004.1276865.

[42] H. Chen, T. Finin, and A. Joshi, "An ontology for context-aware pervasive computing environments," *The Knowledge Engineering Review*, no. 18:3, p. 197–207, 2004.

[43] intersoft consulting, "GDPR Privacy by Design." `https://gdpr-info.eu/issues/privacy-by-design`, 2021. [Online retrieved on 31-May-2021].

[44] D. Industries, "Partner in industrial heating solutions." `https://diracindustries.com/`, 2021. [Online retrieved on 21-May-2021].

[45] VRT, "VRT (broadcaster)." `https://en.wikipedia.org/wiki/VRT_(broadcaster)`, 2021. [Online retrieved on 21-May-2021].

[46] VRT, "Vlaamse Radio- en Televisieomroeporganisatie." `https://www.vrt.be/`, 2021. [Online retrieved on 21-May-2021].

[47] O. Grid, "Off Grid." `https://www.offgrid.Expert/`, 2021. [Online retrieved on 21-May-2021].

[48] N. NV, "Niko Home Control is state-of-the-art." `https://www.niko.eu/en/products/niko-home-control`, 2021. [Online retrieved on 25-May-2021].

[49] L. Tibbets and J. Tane, "IFTTT." `https://en.wikipedia.org/wiki/IFTTT`, 2021. [Online retrieved on 2-June-2021].

[50] Wikipedia, "Recommender system." `https://en.wikipedia.org/wiki/Recommender_system`, 2021. [Online retrieved on 8-August-2021].