University of Ljubljana
Faculty of Computer and Information Science

Jan Meznarič

Extending BPMN for integration of internet of things devices with process-driven applications

MASTER'S THESIS

THE 2nd CYCLE MASTERS STUDY PROGRAMME
COMPUTER AND INFORMATION SCIENCE

Mentor: Prof. Dr. Matjaž Branko Jurič
Co-mentor: Prof. Dr. Axel Küpper

Ljubljana, 2016

Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Jan Meznarič

Razširitev BPMN za potrebe integracije naprav interneta stvari s procesnimi aplikacijami

magistrsko delo

študijski program druge stopnje
računalništvo in informatika

Mentor: prof. dr. Matjaž Branko Jurič
Somentor: prof. dr. Axel Küpper

Ljubljana, 2016

Declaration of Authorship of Final Work

I, the undersigned student Jan Meznarič, registration number 63110268, the author of the final work of studies:

Extending BPMN for integration of internet of things devices with process-driven applications (slov. *Razširitev BPMN za potrebe integracije naprav interneta stvari s procesnimi aplikacijami*)

DECLARE

1. The written final work of studies is a result of my independent work mentored by prof. dr. Matjaž Branko Jurič and co-mentored by prof. dr. Axel Küpper.
2. The printed form of the written final work of studies is identical to the electronic form of the written final work of studies.
3. I have acquired all the necessary permissions for the use of data and copyrighted works in the written final work of studies and have clearly marked them in the written final work of studies.
4. I have acted in accordance with ethical principles during the preparation of the written final work of studies and have, where necessary, obtained agreement of the ethics commission.
5. I give my consent to use of the electronic form of the written final work of studies for the detection of content similarity with other works, using similarity detection software that is connected with the study information system of the university member.
6. I transfer to the UL — free of charge, non-exclusively, geographically and time-wise unlimited — the right of saving the work in the electronic form, the right of reproduction, as well as the right of making the written final work of studies available to the public on the world wide web via the Repository of UL.
7. I give my consent to publication of my personal data that are included in the written final work of studies and in this declaration, together with the publication of the written final work of studies.

In Ljubljana, September 9th, 2016 Student's signature:

I would like to thank my mentor, Prof. Dr. Matjaž Branko Jurič, for his support and guidance. I would also like to thank my co-mentor, Prof. Dr. Axel Küpper, and my supervisor, Kai Grunert, for their help and advice during my stay at TU Berlin.

Special thanks go to my family and friends, for their support during my time at the university.

Contents

List of abbreviations

Abbreviation	Meaning
BPEL	Business Process Execution Language
BPM	Business Process Management
BPMN	Business Process Model and Notation
EPC	Event-driven Process Chain
eEPC	Extended Event-driven Process Chain
HVAC	Heating, Ventilation, and Air Conditioning
IoE	Internet of Everything
IoT	Internet of Things
M2M	Machine to Machine
PaaS	Platform as a Service
REST	REpresentational State Transfer
RFID	Radio-Frequency IDentification
SOA	Service-Oriented Architecture
SOAP	Simple Object Access Protocol
UML	Unified Modelling Language
WSN	Wireless Sensor Network
WSAN	Wireless Sensor and Actuator Network

Povzetek

Naslov: Razširitev BPMN za potrebe integracije naprav interneta stvari s procesnimi aplikacijami

Pojem internet stvari *(Internet of Things - IoT)* opisuje svet, v katerem so vsakodnevni objekti opremljeni z računskimi in komunikacijskimi zmogljivostmi ter povezani z obstoječimi informacijskimi sistemi. Takšni objekti lahko zaznavajo in spreminjajo lastnosti entitet iz stvarnega sveta in tako omogočajo kreiranje informacijskih sistemov, ki se odzivajo na spremembe iz stvarnega sveta in ga tudi spreminjajo. Internet stvari postaja vedno bolj razširjena tehnologija, pomembna na mnogih področjih. Omogoča gradnjo pametnih hiš in mest, izboljšanje infrastrukture in transporta, pomemben pa je tudi v poslovnih okoljih. Preden pa lahko IoT postane del poslovnih okolij, morajo naprave IoT postati aktivni udeleženci v poslovnih procesih.

Poslovni procesi so sestavljeni iz množice koordiniranih aktivnosti, z izvajanjem katerih dosegamo poslovne cilje v podjetju. Posamezno aktivnost poslovnega procesa izvaja bodisi informacijski sistem bodisi zaposleni v podjetju. Z metodami upravljanja poslovnih procesov vseskozi prilagajamo in izboljšujemo poslovne procese ter s tem povečujemo njihovo učinkovitost in izboljšujemo poslovne rezultate podjetja. Poslovne procese modeliramo z različnimi notacijami za modeliranje, med katerimi je najbolj priljubljena in uporabljana notacija BPMN. Modelirane poslovne procese implementiramo in s tem omogočimo njihovo izvajanje. Aplikacije, katerih izvajanje temelji na modeliranih poslovnih procesih, imenujemo procesne aplikacije.

Integracija naprav IoT s procesnimi aplikacijami omogoča modeliranje in

izvajanje poslovnih procesov, ki lahko zaznavajo in spreminjajo lastnosti entitet iz stvarnega sveta. Takšnim procesom pravimo stvarni procesi. Naprave IoT tako postanejo aktivni udeleženci v poslovnih procesih in prevzemajo odgovornost za izvedbo posameznih aktivnosti. Uporaba storitveno usmerjene arhitekture *(Service-Oriented Architecture - SOA)* omogoča učinkovito integracijo naprav IoT z informacijskimi sistemi. Da pa lahko naprave IoT postanejo del poslovnih procesov, je potrebno razširiti notacije za modeliranje procesov, tako da bodo podpirale modeliranje specifik sveta IoT in stvarnih procesov.

V tej nalogi se osredotočamo na razširitev notacije BPMN. Naša odločitev temelji na dveh raziskavah na temo razširjenosti notacije BPMN in njene primernosti za modeliranje specifik sveta IoT. Sodeč po raziskavah je, prvič, notacija BPMN trenutno najbolj razširjena notacija za modeliranje poslovnih procesov, drugič, notacija BPMN med vsemi analiziranimi notacijami podpira največ specifik sveta IoT in je najbolj primerna za razširitev za podporo trenutno nepodprtih specifik. Preden pa se lahko lotimo razširjanja notacije BPMN, je potrebno poznati domenski model interneta stvari.

Najpomembnejši elementi iz domenskega modela interneta stvari so fizična entiteta, naprava IoT in storitev IoT. Fizična entiteta je poljuben objekt iz stvarnega sveta, ki ga želimo vključiti v aplikacijo IoT. Primera fizičnih entitet sta soba in okno. Naprave IoT so posredniki med stvarnim in virtualnim svetom. Delijo se na senzorje, aktuatorje in značke. Senzorji merijo lastnosti fizičnih entitet, aktuatorji pa jih spreminjajo. Primer senzorja je senzor temperature, primer aktuatorja pa klimatska naprava. Značke predstavljajo posebno vrsto naprav IoT, ki so pričvrščene na fizične entitete in se uporabljajo za njihovo identifikacijo. Storitev IoT je spletna storitev, ki preko standardnih vmesnikov izpostavlja funkcionalnosti naprav IoT. Uporabnik lahko tako izvede klic storitve IoT, ki z uporabo funkcionalnosti naprav IoT meri in/ali spreminja lastnosti fizičnih entitet.

Naprave IoT uporabljajo dva načina komunikacije. Prvi način je način tipa zahteva/odgovor, pri uporabi katerega aplikacija pošlje zahtevo napravi

IoT, ta pa izvede zahtevano aktivnost ter aplikaciji odgovori z rezultatom. Drugi način je dogodkovna komunikacija, ki se uporablja, ko naprava IoT pošilja podatke aplikaciji le ob določenih dogodkih. Senzor lahko tako na primer zaledni aplikaciji pošlje izmerjeno temperaturo vsakih 10 minut, ali pa le v primeru, ko temperatura preseže določeno mejo.

Pri modeliranju stvarnih procesov se je potrebno zavedati in razumeti karakteristike, ki so specifične za IoT. Najpomembnejši je koncept fizične entitete, ki je ključnega pomena v stvarnih procesih. Notacije morajo zato zagotoviti ustrezne elemente za modeliranje fizičnih entitet. Stvarni procesi se lahko deloma izvajajo na robnih napravah IoT, zato je potrebno to specifiko podpreti tudi v notacijah. Podobno velja tudi za podatke, ki so prav tako lahko shranjeni na robnih napravah IoT. Zaradi velikega števila naprav IoT in njihovih komunikacijskih sposobnosti, je potrebno zagotoviti elemente za učinkovito modeliranje velikega števila interakcij. Ustvarjeni procesni diagrami morajo kljub velikemu številu naprav ostati nekompleksni. Notacije morajo zagotoviti podporo za modeliranje abstrakcij med storitvami in napravami IoT. Prav tako morajo podpirati mobilno naravo naprav IoT, ki lahko vodi tudi do nedosegljivih naprav. Podobno je potrebno zagotoviti tudi modeliranje morebitne slabe natančnosti podatkov, izmerjenih na napravi IoT, zaradi morebitne slabše ločljivosti ali napake v delovanju. Zaradi velikega števila naprav udeleženih v stvarnih procesih je potrebno zagotoviti nemoteno izvajanje procesa tudi ob morebitnem izpadu posamezne naprave. Zaradi same narave IoT je število dogodkov v stvarnih procesih mnogo večje kot v običajnih poslovnih procesih. Potrebno je zagotoviti podporo za modeliranje teh dogodkov in njihovo učinkovito obdelavo v času izvajanja procesa. Ker je izvajanje stvarnih procesov odvisno od stvarnega sveta in lahko nanj tudi vpliva, je pomembna časovna komponenta pri izvajanju procesov, predvsem zato, da se aktivnosti izvajajo ob točno določenih točkah v času. Vse omenjene karakteristike je potrebno upoštevati in nasloviti, da bo mogoče učinkovito modelirati in izvajati stvarne procese. V tej nalogi se osredotočamo na karakteristike, ki se nanašajo na modeliranje fizičnih entitet, interakcij, dogodkov in abstrakcije naprav IoT.

Trenutno je raziskovalno delo na področju integracije sveta IoT in procesnih aplikacij usmerjeno v razširjanje notacije BPMN, s predlogi razširitev za podporo modeliranju fizičnih entitet, naprav IoT in aktivnosti IoT. Znanstveni članki definirajo nov grafični element in razred za modeliranje fizične entitete. Za modeliranje naprave IoT predlagajo uporabo stez, tako da vsaka steza predstavlja eno napravo IoT. Steze vsebujejo aktivnosti BPMN, ki predstavljajo aktivnosti IoT, ki jih izvede modelirana naprava.

V nalogi analiziramo omenjene predlagane razširitve iz sorodnega raziskovalnega dela na področju modeliranja specifik sveta IoT z notacijo BPMN. Po opravljeni analizi izpostavimo odprta vprašanja, za katera menimo, da najnujneje potrebujejo rešitev. Nadaljnje raziskovanje je osredotočeno na tri raziskovalna vprašanja. Prvič, kako modelirati posamezne lastnosti fizičnih entitet. Drugič, kako izboljšati modeliranje naprav IoT. Tretjič, kako modelirati dogodkoven tip komunikacije uporabljene s strani naprav IoT.

Vsaka fizična entiteta ima več lastnosti. Pri modeliranju procesov želimo modelirati le tiste lastnosti entitete, na katere se sklicujejo aktivnosti v modeliranem procesu. Razširitev modela fizične entitete s podatki o njenih lastnostih omogoča kreiranje natančnejših procesnih diagramov in izboljša njihovo berljivost. Notacije za modeliranje procesov je zato potrebno razširiti, da bodo omogočale modeliranje posameznih lastnosti fizičnih entitet.

Sorodno raziskovalno delo predlaga modeliranje naprav IoT s stezami, tako da je vsaka naprava modelirana s svojo stezo. Pri procesih z večjim številom naprav IoT procesni diagrami zaradi velikega števila stez postanejo veliki in nepregledni. Modeliranje naprav IoT je potrebno izboljšati, tako da bodo modelirani procesi bolj kompleksni in lažje berljivi tudi pri velikem številu modeliranih naprav.

Naprave IoT lahko pri izvajanju aktivnosti IoT uporabljajo dogodkovni tip komunikacije. Trenutno raziskovalno delo predlaga pristope le za modeliranje aktivnosti IoT, ki uporabljajo tip komunikacije zahteva/odgovor. Modeliranje dogodkovnega tipa komunikacije trenutno ni podprto. Notacijo

BPMN je potrebno dopolniti in izboljšati, da bo omogočala modeliranje dogodkovnega tipa komunikacije.

Zastavljena raziskovalna vprašanja rešujemo sledeč naslednji metodologiji. Najprej definiramo zahteve, ki jih morajo izpolnjevati predstavljene rešitve. V naslednjem koraku predstavimo šest rešitev za zastavljena raziskovalna vprašanja. Vsaka predstavljena rešitev je sestavljena iz pristopa modeliranja in predlaganih razširitev notacije BPMN, če je to potrebno. Predstavljene rešitve so v naslednjem koraku evalvirane z uporabo definiranih zahtev.

Pred predstavitvijo rešitev definiramo zahteve, ki nas vodijo skozi proces iskanja rešitev in so uporabljene kot kriterij pri njihovi evalvaciji. Definirane zahteve so razdeljene v štiri skupine: zahteve za podporo modeliranju procesov, zahteve glede veljavnosti procesov, zahteve za podporo izvajanju procesov in zahteve za proces razširjanja notacije BPMN. Najpomembnejše zahteve lahko strnemo v naslednjo izjavo: Uporabnik mora imeti možnost modelirati, katera naprava IoT je odgovorna za izvedbo katere aktivnosti IoT na specificirani lastnosti fizične entitete. Pri aktivnostih senzorjev mora imeti uporabnik možnost modelirati tudi dogodkovni tip komunikacije.

Vse predlagane rešitve so sestavljene iz pristopa modeliranja in morebitnih razširitev notacije BPMN. Vsaki predlagani rešitvi je priložen vzorčen procesni diagram, ki prikazuje uporabo predlagane rešitve. Pri vsaki rešitvi skušamo definirati pristop modeliranja za oba tipa komunikacije: komunikacijo tipa zahteva/odgovor ter dogodkovnega tipa komunikacije.

Prva predlagana rešitev temelji na pristopih modeliranja, predstavljenih v aktualnem raziskovalnem delu. Za modeliranje fizičnih entitet uporabimo elemente, predstavljene v sorodnem raziskovalnem delu. Fizične entitete modeliramo z grafičnim simbolom za prazen bazen z dodanim simbolom in imenom entitete. Vsaka naprava IoT je modelirana s svojo stezo. Aktivnosti IoT, izvedene s komunikacijo tipa zahteva/odgovor, so modelirane z aktivnostmi BPMN. Za modeliranje dogodkovnega tipa komunikacije predlagamo uporabo dogodkov BPMN, postavljenih pred aktivnosti IoT. Lastnosti fizičnih entitet so modelirane zgolj implicitno, s kombinacijo fizične entitete in naprave ter aktivnosti IoT. Evalvacija predlagane rešitve vrne skromne rezultate, saj rešitev ne omogoča eksplicitnega modeliranja lastnosti fizičnih entitet in ne izboljša modeliranja naprav IoT. Rešitev izpolnjuje 6 od 16 definiranih zahtev.

Druga predlagana rešitev temelji na prvi in predlaga združitev vseh naprav IoT v eno stezo in s tem zmanjša kompleksnost dobljenih procesnih diagramov. Ker pri takem načinu modeliranja izgubimo informacijo o napravi IoT, ki je odgovorna za izvedbo aktivnosti, predlagamo modeliranje odgovorne naprave IoT z razširitvijo elementa, uporabljenega za modeliranje asociacije med aktivnostmi IoT in fizičnimi entitetami. Za modeliranje dogodkovnega tipa komunikacije predlagamo uporabo dogodkov BPMN, enako kot v prejšnji predlagani rešitvi. Lastnosti fizičnih entitet so spet modelirane zgolj implicitno. Evalvacija predlagane rešitve vrne rahlo boljše rezultate kot pri prejšnji rešitvi, saj rešitev izboljša modeliranje naprav IoT. Rešitev izpolnjuje 8 od 16 predlaganih zahtev.

Tretja predlagana rešitev je podobna drugi, razlikuje se le v načinu modeliranja naprav IoT. Namesto modeliranja vseh naprav IoT z eno stezo, predlagamo razdelitev na dve stezi, eno za modeliranje senzorjev in eno za modeliranje aktuatorjev. Evalvacija predlagane rešitve je zaradi podobnosti rešitev podobna kot pri prejšnji rešitvi. Tudi ta rešitev tako izpolnjuje 8 od 16 predlaganih zahtev.

Četrta predlaga rešitev poskuša modelirati naprave IoT z uporabo bazenov, s katerimi lahko bolje modeliramo postavitev naprav iz stvarnega sveta. Tako lahko na primer z enim bazenom modeliramo vse naprave, priključene na en prehod IoT. Dogodkovna komunikacija je modelirana z dogodki BPMN. Posamezno napravo IoT modeliramo z razširjenim elementom za modeliranje asociacije med aktivnostmi IoT in fizičnimi entitetami. Lastnosti fizičnih entitet so modelirane zgolj implicitno. Pri evalvaciji predlagane rešitve ocenimo, da so procesni diagrami, ustvarjeni s predlaganim pristopom modeliranja, okorni in slabo berljivi. Rešitev izpolnjuje 7 od 16 predlaganih zahtev.

Peta predlagana rešitev definira nov element notacije BPMN za modeliranje aktivnosti IoT, ki uporabljajo dogodkoven tip komunikacije. Definiramo element *sensing intermediate event*, ki predstavlja točko v procesu, v kateri proces čaka, da prejme podatke od naprave IoT. Element označimo z lastnostjo fizične entitete, katere meritev pričakuje od naprave IoT. Aktivnosti IoT, ki uporabljajo tip komunikacije zahteva/odgovor so še naprej modelirane z uporabo aktivnosti notacije BPMN. Posamezno napravo IoT modeliramo z razširjenim elementom za modeliranje asociacije med aktivnostmi IoT in fizičnimi entitetami. Kljub možnosti definiranja lastnosti fizičnih entitet na novo definiranem elementu, še vedno ni mogoče modelirati fizične entitete ter vseh njenih lastnosti, ki nas zanimajo. Evalvacija predlagane rešitve pove, da rešitev izpolnjuje 8 od 16 predlaganih zahtev.

Šesta predlagana rešitev definira nov element notacije BPMN za modeliranje lastnosti fizičnih entitet. Z novo definiranim elementom lahko modelu fizične entitete dodamo informacije o lastnostih, na katere se sklicujejo aktivnosti IoT. Modeliranje lastnosti fizične entitete je tako eksplicitno. Preostali raziskovalni vprašanji, modeliranje naprav IoT ter dogodkovnega tipa komunikacije, rešujemo z najboljšimi pristopi iz do sedaj predstavljenih rešitev. Vse naprave IoT tako modeliramo z eno stezo, posamezno napravo pa z razširjenim elementom za asociacijo aktivnosti IoT in fizičnih entitet. Dogodkovni tip komunikacije modeliramo z elementom *sensing intermediate event*, ki smo ga definirali v peti predlagani rešitvi. Z evalvacijo predlagane rešitve dobimo dobre rezultate, saj predlagana rešitev naslavlja ter rešuje vsa zastavljena raziskovalna vprašanja. Rešitev izpolnjuje 15 od 16 predlaganih zahtev.

Z evalvacijo predlaganih rešitev smo dobili oceno, koliko definiranih zahtev izpolnjuje posamezna predlagana rešitev. Najboljši rezultat je dosegla šesta predlagana rešitev, ki izpolnjuje 15 od 16 definiranih zahtev. Predlagana rešitev omogoča eksplicitno modeliranje lastnosti fizičnih entitet z uporabo novo definiranega elementa BPMN. Modeliranje naprav IoT je izboljšano z združitvijo vseh naprav v eno stezo in razširitvijo elementa za asociacijo aktivnosti IoT in fizične entitete. Dogodkovni tip komunikacije naprav IoT je mogoče modelirati z novo definiranim elementom *sensing intermediate event*. To rešitev predlagamo kot končno rešitev za raziskovalna vprašanja, zastavljena v tej nalogi. Ocenjujemo, da je rešitev ustrezna in da predstavlja korak v pravo smer pri razvoju notacij za modeliranje stvarnih procesov.

Ključne besede: internet stvari, procesne aplikacije, BPMN.

Abstract

Title: Extending BPMN for integration of internet of things devices with process-driven applications

Internet of Things (IoT) is a rising technology, which is becoming an important building block of information systems in many different areas, including enterprise environments. For IoT to become a part of process-driven enterprise applications, process modelling notations have to be extended. This thesis focuses on introducing a modelling notation for modelling of properties of real-world entities. It also improves the modelling of IoT devices and different communication types used by them. To solve stated problems, we propose six solutions in form of modelling approaches with corresponding extensions to BPMN. Proposed solutions are based upon and evaluated against four sets of identified requirements. Solution, which satisfies the highest number of requirements, is our final solution and is an adequate answer to our research questions.

Keywords: internet of things, process-driven applications, BPMN.

Chapter 1

Introduction

With the rise of information technology came the idea of making everyday objects identifiable and later interconnected, which shaped into the paradigm today known as Internet of Things (IoT). Although this paradigm is quite old, IoT is still a new rising technology, that needs a lot of research, before it can become widely accepted. Currently, there is a lot of buzz surrounding IoT and its impact on our everyday lives. It promises the emergence of smart homes, smart cities, smart infrastructure, smart health monitoring systems and many more. But in order to make all of the listed possible, many problems have to be solved first.

IoT will enable us to build a world, where everyday objects are interconnected and integrated into existing information systems. To achieve this goal, we have to develop and define widely accepted and standardised solutions, for integrating all the new concepts from the IoT world with existing technologies, architectures, frameworks, software solutions and information systems.

As it is happening in many different application areas, IoT is also becoming an important building block of enterprise applications. Business oriented software solutions and information systems have to address the raising importance of IoT and integrate it into existing solutions and systems. Since enterprise information systems are often implemented in the form of process-driven applications, we have to introduce new process modelling concepts and extensions to modelling notations. By defining new modelling concepts and extensions we will be able to include IoT concepts into process diagrams and enable execution of such processes in process execution engines.

In this thesis we aim to introduce important concepts from IoT domain model to business process modelling notations. Since BPMN is the most widely accepted and used business process modelling notation, we propose extensions to BPMN, that enable creation of processes, supporting characteristics from the IoT world. We analyse and evaluate IoT domain model and state of the art research, and focus on three open topics, that are currently not addressed by process modelling notations. We propose research questions with focus on modelling of properties of real-world entities, IoT devices and different types of communication used by them. Further, we propose solutions to the research questions, evaluate them, and present a final solution.

The rest of the thesis is structured as follows. Chapter 2 presents motivation for our work, introduces Business Process Model and Notation (BPMN) and describes the state of the art research on this topic. In chapter 3 we analyse the state of the art research, identify open topics, propose our research questions and describe used methodology. Chapter 4 states our requirements for solving proposed research questions. In chapter 5 we propose different solutions for solving proposed research questions, which are then evaluated against defined requirements in chapter 6. Chapter 7 proposes future work and concludes the thesis.

Chapter 2

Foundation

This chapter first describes the integration of IoT with process-driven applications. Secondly, it describes aspects of Business Process Model and Notation significant for this thesis. Lastly, it presents the state of the art research on IoT and its integration with process-driven applications.

2.1 Motivation

This section first describes IoT, its most important application areas and open challenges, that still need to be addressed, in order for IoT to become a technology worth investing in. As IoT is becoming an important building block of enterprise applications, we further describe process-driven applications and our reasoning, why process modelling notations shall be extended to support modelling of real-world processes.

2.1.1 Internet of Things

Internet of Things (IoT) is a hot research topic nowadays and an important technology for the future of the internet. This section describes different definitions of IoT, its future trends and expected growth, application areas and open challenges.

Definitions

The term Internet of Things (IoT) was first used by Kevin Ashton at the end of the last century to describe the idea of using RFID tags to identify physical objects. Based on the unique identifier stored on the tag, information related to identified objects could be easily retrieved from an underlying information system.

Since then the idea of IoT grew alongside the development of new technologies and number of new use cases and problems, which could be solved by this new technology. IoT evolved from simple RFID based applications to broad networks of interconnected devices with sensing and actuating capabilities. Definitions of IoT evolved in parallel to development of IoT technologies. Since IoT became much more than just system for identifying RFID equipped physical objects, consequently new definitions emerged.

The RFID group defines Internet of Things as: "*The worldwide network of interconnected objects uniquely addressable based on standard communication protocols.*" [8]. This definition focuses on IoT as a network of interconnected objects, but does not specify computing capabilities of mentioned objects.

As IoT developed further, a SAP IoT focused research group proposed the following definition for IoT: "*A world where physical objects are seamlessly integrated into the information network, and where the physical objects can become active participants in business processes. Services are available to interact with these 'smart objects' over the internet, query and change their state and any information associated with them, taking into account security and privacy issues.*" [28]. This definition focusses not only on IoT as a network of interconnected devices, but also includes integration of such networks into information systems and business processes. Integration with existing information systems enables development of applications, which can solve existing problems in real-word scenarios and help IoT become a widely-spread technology.

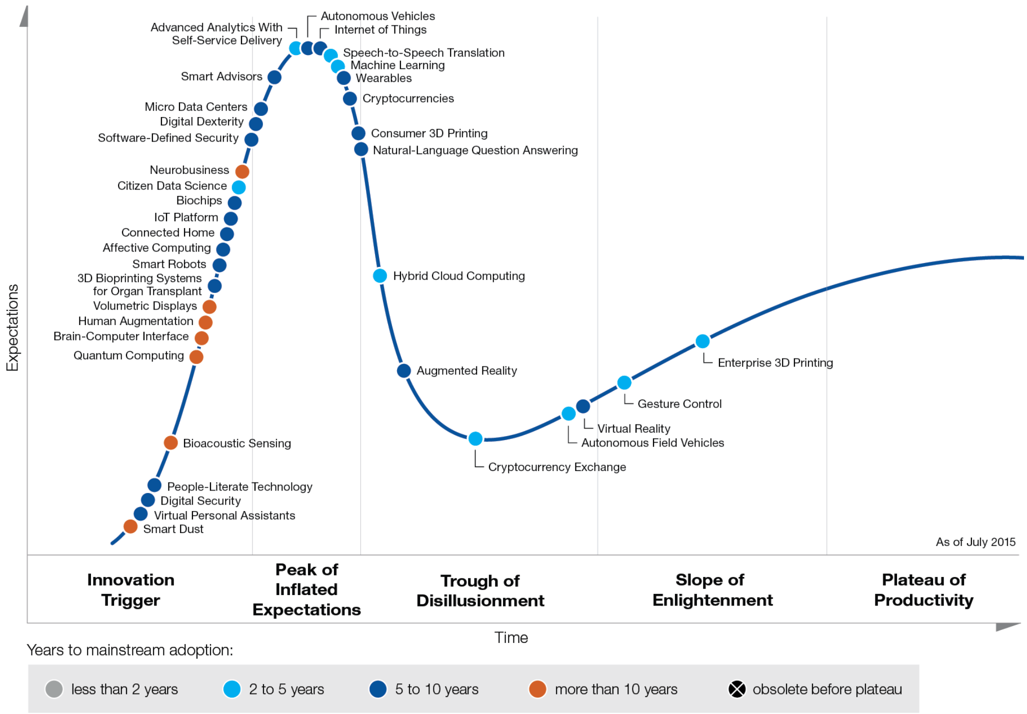

Figure 2.1: Gartner 2015 Hype Cycle of Emerging Technologies (Source: Gartner Inc. [19])

Trends and expected growth

Number of interconnected devices raised rapidly in last years. According to [8], in year 2011 the number of interconnected devices on the planet overtook the number of living people. Number of interconnected devices is expected to reach 24 billion devices by 2020.

With number of interconnected devices rising, the field of IoT is getting a hot topic in research and business communities. IoT opens the door for numerous new research projects and holds a lot of value, which can be utilised by businesses from many different fields.

For a few years now, Gartner places IoT on the hype cycle, which is used to represent emergence, adoption, maturity and impact on applications of different technologies. Hype cycle from year 2015 forecasts, that IoT is

5 to 10 years away from mainstream adoption and on the peak of inflated expectations. Hype cycle is shown in figure 2.1.

Hype cycle also includes technology named *IoT platform*, which is placed slightly behind IoT in the cycle. *IoT platform* is a term used to describe integration of IoT with Platform as a Service (PaaS) cloud infrastructure. Such integration enables businesses to control IoT endpoints and build applications utilizing potential of IoT. Another prediction made by Gartner, supporting emergence of *IoT platforms*, is that by 2020, more than 50 percent of all new applications developed on PaaS will be IoT-centric [34].

Cisco talks about Internet of Everything (IoE) as a next evolutionary step after IoT which interconnects people, processes, data and things. They estimate that IoE creates $14.4 trillion of value at stake for companies and industries [3]. Estimated value at stake is fueled by improving the areas of asset utilization, employee productivity, supply chain and logistics, customer experience and innovation in reducing time to market.

All trends and expectations described in this section illustrate the importance of IoT in future research projects and business environments. IoT is clearly the technology that will be around for many years to come and needs a lot of research and development work in order to unleash its full potential.

Application areas

As stated before, IoT started as a technology used to identify physical objects in real world by equipping them with RFID tags. Since then, it has developed and entered many different application areas. Survey made in [18] shows, that IoT is now present in a wide range of industries, including retailing, manufacturing, healthcare, insurance, home appliances, heavy equipment, airline and logistics.

Authors in [8] define four major application domains for IoT: personal and home, enterprise, utilities and mobile. The personal and home domain focuses on individuals and their devices. It enables creation of smart homes supporting home automation, energy management and smart entertainment.

According to [18], "*The smart home is known to be at the forefront of innovation regarding IoT monitoring and control systems.*" Health monitoring devices in combination with smart homes enable creation of monitoring systems for aged-care and present large market in personal and home IoT domain.

Enterprise domain focuses on enterprise applications, that can interact with real world environments. IoT enables creation of environmental monitoring applications in factories and other enterprise environments, providing support for manufacturing and supply chain integrity management [10]. With introduction of IoT into agriculture, we talk about precision agriculture with improved efficiency and reduced costs. In retail domain, retailers can now focus on providing better experience for customers and better business results for them. Example of smart retail system is presented in [20], where authors present an IoT based retail system, that provides quality sensing of stocked products, dynamic pricing and targeted advertising, improving the business efficiency.

In utilities domain, information provided by devices is used to improve services and to build applications for resource management in companies. Device networks in this domain are extensive and laid out on regional or national scale. They provide support for creating smart metering systems and environmental monitoring, which lay groundwork for upcoming smart cities.

Cheap and available mobile devices enable the emergence of a mobile IoT domain. It enables creation of smart transportation and smart logistics, which can reduce traffic noise, air pollution and improve efficiency of transportation. With traffic monitoring and smart routing, such systems can prevent traffic jams and improve commute experience. Mobile IoT supports efficient logistics management with monitoring of transported items and efficient transportation planing.

Open challenges

Before IoT can become a mainstream technology, there are still many challenges that need to be addressed. Solving those challenges will drive the industry to start investing in IoT and adopt it in future projects and applications.

According to [27], the biggest challenges result from the large scale of the IoT ecosystem, consisting of large number of devices that form vast networks, which produce high amount of events. Those large networks are based on the internet, which architecture has many limitations in terms of mobility, availability, manageability and scalability, that present major barriers to IoT [2].

To organize and maintain such a big number of devices, efficient standardized architectures are needed. Such architecture should support low-power, low-cost and yet fully networked and integrated devices, compatible with standard communication technologies [25]. While some specific-purposes architectures have already been proposed, the overall IoT architecture with user at its centre is still needed [8].

Standardization in IoT is needed to overcome barriers between different devices, networks and applications and improve interoperability. Various standards need to be designed and adopted, such as security, communication and identification standards. For implementing IoT in industrial environments, industry-specific guidelines need to be considered [5]. In enterprise environments, standards for successful integration of IoT into existing enterprise IT systems are needed.

Many of the challenges caused by the large number of non-standardised heterogeneous devices can be solved with middleware. Authors in [27] analyse existing middleware, describe their shortcomings and argue, that more research in field of IoT middleware is needed in order to provide middleware support for vast IoT ecosystem.

From the device point of view, efficient energy sensing is needed in order to deploy wireless standalone sensors in the environments with minimum mandatory maintenance. Energy consumption may be reduced with better communication protocols and smart infrastructure, which minimises the amount of communication between participants in the network.

As with all large scale networks, major concerns with IoT are security and privacy. IoT networks can be attacked or abused in many different ways, such as disabling the network availability and accessing or manipulating sensible data in the network. Because of the limited computing capabilities of IoT devices, possibilities for security measures, which are common in other IT systems, are limited in IoT.

In order to make sense of the large number of data generated by vast IoT networks, it has to be analysed with data mining approaches. There is a need for advanced data mining tools to mine streaming data from sensor networks as well as image and video data [18]. Future advantages in data mining research will enable better understanding of IoT-generated data and provide possibilities for creating new types of smart applications.

Cloud computing provides ideal infrastructure for storing, analysing and visualising the data generated by IoT. It enables the rapid creation of applications by providing domain specific programming tools and environments [8]. Future research is needed to provide such tools, that will simplify the creation of cloud-based IoT applications.

Cloud-based services are typically orchestrated with process-driven applications, which enable creation of cloud-based applications constructed mainly of carefully orchestrated services. With emergence of IoT to the cloud, process-driven applications are becoming a hot research topic in relation to IoT. To build process-driven applications that support special characteristics from IoT domain, process modelling notations shall be extended to include notations for such characteristics.

In next section we describe process-driven applications, so that we can later better understand their integration with IoT.

2.1.2 Process-driven applications

The main task of information systems in business environments is to support existing business processes by providing applications and services for performing individual tasks of said processes.

Service-oriented architecture (SOA) advocate the use of loosely coupled service based information systems for supporting business activities in enterprise environments. In SOA architecture, services are usually orchestrated in a specific order, reflecting business processes of the company. Applications, which execute modelled business processes, are process-driven applications.

To better understand process-driven applications, we describe business processes and later their relation to SOA.

Business processes

A business process consists of a set of coordinated activities, which are performed by enterprise's information system and its services or by employees. Each business process exists to accomplish particular business goal. Efficiently managed and highly optimised business processes help companies to achieve their business goals with maximum possible efficiency [12][14].

Companies react to ever-changing business environments by modifying their business processes. Business process management (BPM) is a method of aligning a business organization with its clients needs as well as continuously improving the business processes. Each change in the business process requires changes in supporting IT systems. BPM suites enable businesses to efficiently maintain their business processes with minimum changes to their existing IT systems.

Business processes are modelled with process modelling notations. The most widely used notation is Business Process Model and Notation (BPMN) [26]. Other popular notations are Event-driven Process Chain (EPC), Extended Event-driven Process Chain (eEPC), UML activity diagrams and flow charts. Process modelling notations enable creation of graphical representations of business processes, called process models or process diagrams.

In order to execute business processes, their implementation must include details providing an end-to-end IT support for the process. Processes can be implemented with BPMN 2.0, BPEL or other service orchestrating languages. Process implementation phase is typically done by SOA developers in contrast to the modelling phase, which is performed by domain experts and process owners.

BPM in relation to SOA

In order for executable business processes to orchestrate services, such services must first exist and be available for use by process execution engines. SOA provides architecture guidelines on how to build information systems that expose organization's IT assets as reusable business services [12].

Business services provide access to business assets and are crucial for building loosely-coupled process-driven IT systems, which provide execution support for identified business processes. They are the enablers of reuse principle, which is a key to fast and efficient development of new software solutions. The end-to-end automation of business processes is only possible when sound, robust and reliable applications and services are available as groundwork for automated business processes [13].

SOA is the key enabler for integration of various resources into business IT systems and its business processes. Standardised service interfaces can be used to interconnect different IT systems, applications, services, devices and people, enabling creation of automated business processes, that can interact with wide enterprise environment.

Next section describes integration of IoT with SOA based systems, orchestrated by business processes.

2.1.3 Business processes with support for IoT

IoT provides the means for creation of processes that can interact with objects from real world. Such processes can sense the state of physical entities or even act on them and change their state. Such processes are called real-world processes.

SOA enables integration of different systems, applications and services. With emergence of IoT into the area of SOA [31], its integration capabilities now also connect entities and devices from IoT with existing systems and applications.

SOA's loosely-coupled integration capabilities and event processing techniques provide possibilities for integrating existing IT systems with IoT devices, which will play an important role in business processes of the future. Large number of events generated by IoT can now be handled and analysed by SOA systems, that support business events and event processing since the early days [12].

As described in 2.1.1, enterprise domain is an important application area for future IoT applications. SOA provides solutions to integrate IoT into existing IT systems, but in order to provide an end-to-end support, IoT has to become a part of business processes. To successfully model real-world business processes, which are made possible by integration of IoT with SOA, business process modelling notations have to be extended to support modelling of characteristics of real-world business processes.

Inclusion of IoT into business processes enables businesses to perform business analytics of real-world processes. With continuous monitoring and analysing, such processes can be constantly improved, which minimises the costs and increases the productivity of the business.

Next section focuses on Business Process Model and Notation (BPMN), mainly on the aspects significant for this thesis.

2.2 Business Process Model and Notation

This section first explains our decision to focus on extending BPMN. Secondly, it introduces BPMN metamodel. Lastly, it describes the process of extending BPMN used in this thesis.

2.2.1 Extending BPMN for integration of process-driven applications with IoT

In section 2.1.3 we stated, that process modelling notations have to be extended in order to support modelling of real-world process. In this thesis we are focusing on extending Business Process Model and Notation (BPMN) [26].

BPMN is a widely used process modelling notation, developed and maintained by Object Management Group (OMG). Since version 2.0, it also provides process execution semantics, which enables BPMN processes to be executed by process execution engines.

Authors in [16] performed a study on the acceptance of BPMN, purposes of its usage and its main advantages and disadvantages. Results were obtained by performing Systematic Literature Review (SLR). Results show, that BPMN holds the title of *de-facto* standard for business process modelling. BPMN's acceptance, popularity and high usage in process-driven applications are among the reasons, that we are focusing on extending BPMN for integration of process-driven applications with IoT.

Further reasons are based on the work [24], where authors analyse different process modelling notations in terms of their current support for modelling properties of real-world processes. BPMN, eEPC and UML 2.3 are analysed, and BPMN is identified as a notation with support for the largest number of real-world properties and as the most suitable for extending, to support modelling of currently unsupported properties.

2.2.2 BPMN 2.0 metamodel

BPMN version 2.0 brought almost zero changes to the notation, i.e. graphical model, but instead focused mainly on defining a metamodel, the formal specification of the semantics of the notation [29].

Metamodel describes the semantics of the elements from the graphical model and relation between them. Each element from the graphical model has a corresponding class in the metamodel. All valid BPMN models must conform to the specifications of the metamodel.

Metamodel consists of object classes with corresponding attributes. Some classes are abstract and do not represent any element from the graphical model. Classes can be subtypes of other classes and inherit their attributes. Each class have a set of attributes, which can be either required or optional. They are used for storing the information about the modelled element.

When proposing extensions to BPMN, we talk about extending the metamodel with new classes, or about extending existing classes of the metamodel with new attributes.

2.2.3 Process of extending extending BPMN

Main goal of this thesis is to define new modelling approaches with corresponding extensions to BPMN. Different possible ways of proposing extensions to BPMN are described in this section.

Simple extensions can be implemented with BPMN's extensibility mechanism, introduced in BPMN version 2.0. It allows extending standard BPMN elements with additional attributes [26]. All extended elements shall not contradict the semantics of existing BPMN elements and shall maintain the look and feel of BPMN. There are two types of extensions; mandatory extensions have to be understood by implementation of process execution engine. Optional extensions may be ignored by process execution engine and are used only to improve the readability of created diagrams.

Some proposed solutions cannot be realised with BPMN extensibility mechanism and may require a deeper change in the BPMN metamodel. This can include defining new classes in order to realize proposed modelling approaches, or even redefining relationships between existing classes. For such extensions to be executed, new implementation of process execution engine is needed.

Next section presents the state of the art research in the fields of IoT and its integration with process-driven applications.

2.3 State of the art research

This section presents the state of the art research on IoT and its integration with process-driven applications. First, the IoT domain model and its main components are described. Secondly, it focuses on internet oriented vision for IoT. Thirdly, the IoT characteristics relevant for modelling of real-world processes are described. Lastly, it presents the state of the art extensions for modelling real-world processes in BPMN.

2.3.1 IoT domain model

In order to discus IoT, we first need standardised terminology describing concepts from IoT domain. This section provides a description of concepts from IoT domain model that are used in this thesis.

As stated before, IoT is applicable in many different areas and applications. For demonstration purposes in this thesis, we use scenarios related to home automation system. Note that it is not the purpose of this work to optimize home automation, but we use it only as a device to depict a concrete application of our approach. Similar IoT applications could be developed for support of factory maintenance or supply maintenance in enterprise environments, which are typically orchestrated with business processes.

Our knowledge of IoT domain model is based on two papers. First, [9] defines the most important terms from IoT domain model. It provides definitions for *physical entity*, *IoT device*, *IoT resource* and *IoT service* and defines relationships between them. Similar ideas are then further defined and expanded in work [28].

Physical entity

With IoT, we are talking about interaction between digital and physical worlds. We want to connect physical objects with digital applications. To do so, we first have to define a term *physical entity*.

Physical entity, or sometimes also called *entity of interest* [9], is a real-world entity, which we are interested in from a process modelling point of view [24]. *Physical entity* can be anything that is a part of our surroundings, or even a location [33]. Examples of *physical entities* from a home automation system scenario are rooms, windows, doors etc.

In generic IoT scenario, *user* interacts with a *physical entity* [28]. *User* can either be a person or a software agent, that has interest in sensing or changing the state of a *physical entity* in its environment.

Digital proxy and smart object

Digital proxy is a digital representation of a *physical entity* in digital world [28]. It must be bidirectionally associated with a *physical entity* it represents. It only represents a set of properties of the *physical entity*, which *user* is interested in. All changes to *physical entity* must be reflected on *digital proxy* and vice versa.

Authors in [28] also define a term *smart object* as a *physical entity* with an associated *digital proxy*. Changes in the properties of a *smart object* must be represented in both physical and digital world.

IoT device

IoT device is a technical communication device, which can sense the state or interact with a *physical entity*. It can be attached to a *physical entity* or it can be installed in its environment.

From the functional point of view, there are three types of *IoT devices* [28]:

- **Sensors** are sensing or monitoring a *physical entity* and providing *user* with information about entity's state. We say, that they are performing *sensing activities*. Examples of *sensors* from a home automation system scenario are temperature sensor, window status device, luminosity sensor, air-quality sensor etc.

- **Actuators** are devices that can modify the physical state of the *physical entity.* We say, that they are performing *actuating activities.* Examples of *actuators* from a home automation system scenario are heating, ventilating, and air conditioning (HVAC) system, window automation device, smart lightning system etc.

- **Tags** are identifiers, attached to *physical entities* in order to identify them and collect further information about them from an information system. They can be read by special *sensor* devices called *readers.* Examples of tags are QR codes, radio-frequency tags etc.

Sensing and *actuating activities*, performed by sensors and actuators, are called *IoT activities.*

Sensors, *actuators* and *tags*, together with wireless communication capabilities, form a *wireless sensor and actuator network (WSAN)*, sometimes called *WSN. WSANs* are networks, consisting of a number of network nodes with sensing and actuating capabilities, that provide wireless communication channel. Backend systems can access WSANs through a special node, that acts as gateway [28]. In this thesis we mainly focus on individual *IoT devices* and not on WSANs.

IoT resource

IoT resource is a software component hosted by *IoT device* [28]. It provides computational capabilities supporting *IoT activities* performed by *IoT devices.*

In case of *sensing activity*, IoT resource is providing retrieval of physical properties of associated *physical entity*, captured through *senosrs.* In case of *actuating activity*, *IoT resource* is providing modification of physical properties of associated *physical entity* through the use of *actuators.* Each *IoT device* can be able to perform many different *IoT activities* and therefore can host more than one *IoT resource.*

Authors in [22] have identified *IoT resource* as a process resource from a process modelling perspective. *IoT resources* have process execution responsibilities during process execution time.

Example of an *IoT resource* from a home automation system scenario is a software component running on temperature sensor, providing measured temperature of a *physical entity* that it is associated with.

IoT service

Implementations and interfaces of *IoT resources* are dependent on underlying technologies and manufacturers, and are therefore highly heterogeneous. In order to provide interoperability, access to *IoT resources* is usually provided in the form of *IoT services* [28].

Results of a survey performed in [32] show, that there is still no common nomenclature and that the term *IoT service* is used differently among different projects. In this thesis, we use the term *IoT service* for a service with a standardised interface (e.g. SOAP, REST etc.) exposing *IoT resources* in order to enable integration with SOA applications. Exposed resources provide functionalities to perform *IoT activities* on *physical entities*. Our definition is based on work presented in [32], [6] and [7].

Interoperability, provided by *IoT services*, is highly important, when we want to include IoT into enterprise environments, which are typically build as SOA and orchestrated with business processes. In order for IoT to become a part of business processes, we have to service-enable *IoT resources* [11].

An example architecture for integration of IoT in enterprise environments is presented in [31]. Proposed architecture is based on WS-* standards and demonstrates the benefits of service-enabling *IoT resources* for integration with enterprise applications.

Example of *IoT service* from a home automation system scenario is a service with a standardized interface, which is exposing the *IoT resource* running on temperature sensor, providing measured temperature of associated *physical entity*.

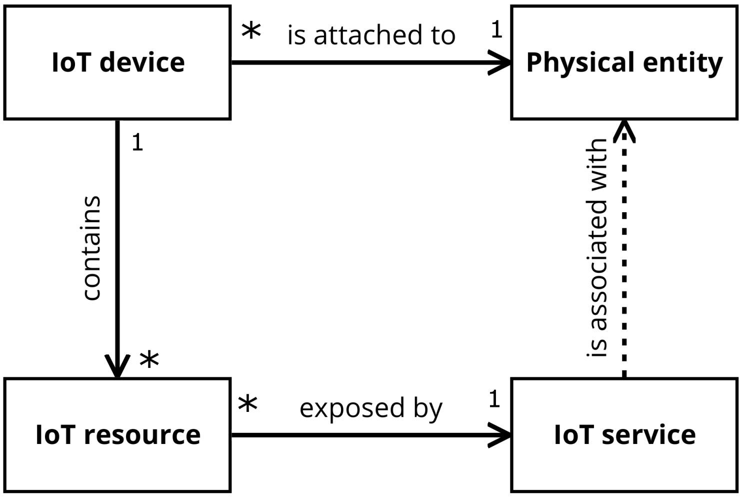

Figure 2.2: Relationships between concepts from IoT domain model.

Relationships between concepts from IoT domain model

In order to have a complete picture of IoT domain model, it is important to understand relationships between described concepts. Relationships are defined as follows:

One or more *IoT devices* are attached to a *physical entity* or installed in its environment. Each *IoT device* contains one or more *IoT resources*, which provide functionalities for performing *IoT activities* on *physical entities. IoT resources* are exposed by *IoT services*. Each *IoT service* can expose one or more *IoT resources* [9]. Described concepts from IoT domain model and relationships between them are represented in figure 2.2.

In next section we take a look at how individual IoT building blocks are interconnected in order to become a part of the internet.

2.3.2 Internet oriented IoT vision

Authors in [1] define three IoT visions: things oriented, focusing on providing everyday objects with computing and networking capabilities, internet oriented, defining communication for smart objects using standard internet protocols and their integration with existing web services, and semantic oriented, focusing on organising large number of interconnected devices and storing, organising, analysing and representing vast amount of data generated by them. In this theses we focus on internet oriented vision, as we want to integrate IoT with process-driven applications built on SOA principles. This section describes building blocks supporting the internet oriented vision of applications in our thesis.

IoT middleware

There is a gap between internet and complex distributed IoT infrastructure, comprised of numerous heterogeneous devices. To simplify the development of IoT-enabled applications and services, that gap has to be breached with use of middleware [18]. IoT middleware is a software layer that provides transparency with regard to the heterogeneous components and therefore offers common services for applications and eases application development [27].

Furthermore, IoT middleware enables *IoT resources* to be exposed in form of *IoT services*, which enables integration of heterogeneous sensor and actuator networks with SOA applications [8].

Similarly as in other contexts, SOA based middleware architectures for IoT are gaining popularity in last years. Adoption of SOA principles allows decomposition of complex systems, usage of well defined interfaces and standardised protocols, and reusage of software and hardware. Process-driven applications can be built on top of SOA, making it easier for an enterprise to adapt itself to the changes imposed by the market evolution [1].

IoT middleware provides support for building IoT applications consisting of IoT edge devices and a backend system. IoT edge devices are *IoT devices* such as sensors or actuators or networks such as WSNs or WSANs. Backend system is responsible for executing the business logic of an IoT application. It can be implemented using SOA standards and can include components such as process execution engine. Backend system can be cloud based, as explained in next section.

IoT cloud

With service-enabling IoT, an important step has been made towards integration of IoT with cloud computing. Looking at *IoT devices* and *IoT activities* as *IoT services* enables building of IoT applications, which consist of many heterogeneous *IoT devices*, with cloud in the middle to connect them and provide foundation for building modular, interoperable and scalable SOA applications.

Cloud-centric vision for implementation of IoT is presented in [8]. Cloud is an ideal backend solution for storing, analysing and visualizing enormous amounts of data generated by IoT [18]. Only with cloud computing in the centre, realization of full potential of IoT is possible. Cloud provides highly interoperable environment for experts from many different fields to come together and build scalable IoT applications, releasing full potential of *IoT devices* installed in surrounding environment. In such example, sensors provide data, that is stored in cloud data centres, analysed by data mining experts, visualized by graphic designers and acted upon by business domain experts.

Cloud-based IoT applications can be applied in many different application areas. By combining services from multiple stakeholders and scaling abilities to support a large number of users, they can be used to improve home environments, transport, community environments, nationwide utilities and infrastructure, making it possible for emergence of smart homes and smart cities. Creation of such application have to be supported by provisioning domain specific programming tools and environments.

With process-driven applications running on top of cloud based IoT systems, users are enabled to create applications supporting different use cases from their domains, using modelling notations they are already familiar with from modelling business processes in a more traditional sense.

Types of communication used in IoT

IoT devices must be able to communicate with each other (M2M) and with the backend system. Edge devices can be connected to the internet directly, or through an IoT gateway. Different communication protocols are used in IoT, differing by communication type, underlying transfer protocol, support for quality of service, support for security mechanisms, overhead size and architecture type [15].

Communication types in IoT can correspond to different sequence of exchanged messages as a result of various use cases. Messages exchanged between backend system and *IoT devices* can follow a request/response paradigm or can be transmitted only in cases of predefined events.

A request/response communication paradigm is used in scenarios, where backend system requests some data from an *IoT device*, then *IoT device* returns a response (e.g. request current temperature from temperature sensor). This type of communication enables application to get desired information at any point in time.

An event-based communication is used in scenarios, where *IoT device* sends data in cases of predefined events, such as time intervals or fulfilment of the conditions (e.g. temperature sensor sends temperature every 10 minutes or temperature sensor sends temperature when temperature is higher than 25° Celsius). This type of communication can decrease the number of exchanged messages in certain use cases, compared to a request/response approach. Selection of a communication type used in an application depends on the use case.

In the next section we describe IoT characteristics that are relevant and need to be taken into account when modelling real-world business processes.

2.3.3 IoT characteristics relevant for process modelling

To successfully integrate IoT with business processes, we first need to identify and understand IoT characteristics, which need to be considered when modelling real-world processes, i.e. processes that can interact with real-world environment. IoT characteristics relevant for modelling real-world processes are identified and analysed in [24]. Identified characteristics are:

- **Entity-based concept:** Process modelling notations supporting IoT integration need to implement a concept of *physical entity*, which is a core concept of IoT. Notations shall provide elements for modelling *physical entities* and process activities related to them. In this thesis we try to improve BPMN's support for this characteristic.

- **Distributed execution:** With number of communication-enabled mobile devices and their computing capabilities rising, responsibility for performing an individual process tasks can be moved to them. This way we get highly distributed business processes, which are executed on the edges of the infrastructure. Support for distributed process execution shall become part of process modelling notations.

- **Interactions:** Because of a vast number of *IoT devices* and their ability to communicate with each other, exists a large number of possible interactions in business processes. Furthermore, different types of interactions and communication types are used, e.g. a request/response paradigm and an event-based communication. Business process modelling notations shall provide support for such interactions. In this thesis we try to improve BPMN's support for this characteristic.

- **Distributed data:** Similarly as with distributed execution, IoT infrastructure can provide possibilities for distributed data storage. Support for modelling distributed data shall become part of process modelling notations.

- **Scalability:** IoT application can consist of many different entities and devices. Complexity of modelled business processes must be independent from the number of IoT participants. Similarly, the execution of the process shall not be affected by the number of IoT participants.

- **Abstraction:** As stated in 2.3.1, each *physical entity* can be associated with one or more *IoT devices*, which provide one or more *IoT resources* exposed by *IoT services*. Process modelling notations shall provide types of abstraction needed, to model *physical entities* and all associated *IoT services* in intuitive and efficient way. In this thesis we take into consideration BPMN's support for this characteristics.

- **Availability / Mobility:** In IoT, *physical entities* and associated *IoT devices* can be mobile. Process modelling notations shall provide means to model such mobility and execution engines shall provide support for execution of processes including mobility. Mobility in BPMN processes is analysed in [17], where authors describe the need for modelling mobility in BPMN, present methods for identifying mobile processes, introduce location based events and propose a way to model mobile participants.

- **Fault tolerance:** In IoT, availability of *IoT devices* cannot be guaranteed. Process modelling notations shall therefore provide means to model unavailability and alternative flows, which shall be executed in cases of unavailability.

- **Flexibility / Event-based:** Number of possible events occurring in IoT scenarios is much higher than in a typical business processes. Many changes in state of *physical entity* can be modelled as events. Business process modelling notations shall provide means for modelling IoT specific events. IoT specific events are analysed in [4]. In this thesis we try to improve BPMN's support for this characteristic.

- **Uncertainty of information:** *IoT devices* provide sensed information with different levels of accuracy. Business process modelling notations shall support modelling information certainty, so that process flow can be altered in cases of bad measurements. Examples of services and BPMN extensions with support for quality of information are presented in [23].

- **Real-time:** Because real-world business processes interact with entities from real world, it is important to take the time dimension into account. It is important, in which point in time a certain task is performed. Business process modelling notations shall provide means to model exact time points for process steps that interact with real world.

In this thesis we mainly focus on entity based concept, interactions, abstraction and event-based IoT characteristics. Next section introduces some state of the art approaches for modelling identified IoT characteristics in BPMN.

2.3.4 Modelling IoT characteristics with BPMN

As described in the previous section, there are many IoT characteristics, which need to be addressed by process modelling notations in order to successfully and efficiently model real-world processes. Current research has so far been mainly focused on including an entity-based concept into BPMN. In this section we present proposed state of the art extensions for BPMN, which are addressing IoT specific concepts.

Physical entity

Modelling of *physical entity* was one of the first things presented in related research work. First attempts for modelling of *physical entity* in BPMN were presented in works [24] and [30], where authors proposed modelling of *physical entity* with a text annotation or with a newly defined element *PhysicalObject*, respectively.

In [21], a few different extension are proposed and evaluated based on the predefined requirements. Based on evaluation, authors propose an extension for modelling of *physical entity*, which meets all proposed requirements. They propose modelling of *physical entities* with the same graphical stencil as black box pool, with a small cow as a self explanatory marker, expressing that a

Figure 2.3: Proposed graphical stencil for modelling of *physical entity*.

physical entity may even be alive. BPMN metamodel is extended with a new class *PhysicalEntity*.

Proposed graphical stencil for modelling of *physical entity* is presented in figure 2.3.

IoT device

Authors in paper [30] identify *IoT devices* as *participants* in a collaboration diagram. Concept of *participant* is in BPMN defined as an entity which executes a process. *IoT devices* have a responsibility to execute IoT related process activities. Based on that, authors propose modelling of *IoT devices* as lanes.

Idea is further pursued in [22], where *IoT devices* and their *IoT resources* are analysed for their roles in business process. *IoT devices* and *IoT resources* are both recognised as business process resources because of their roles as performers. Authors again propose modelling of *IoT devices* as lanes.

In [33], authors present a WSN-specific extension for BPMN. They propose modelling of a set of *IoT devices* as a separate pool, for better distinguishably between WSN and other process performers.

IoT activity

As described in 2.3.1, *IoT devices* perform *IoT activities* on *physical entities*. To successfully model real-world processes, we need a way to model those activities in process diagrams.

Authors in [30] define *IoT task* as IoT specific BPMN task, used for modelling of *IoT activities*. They extend the BPMN metamodel with two

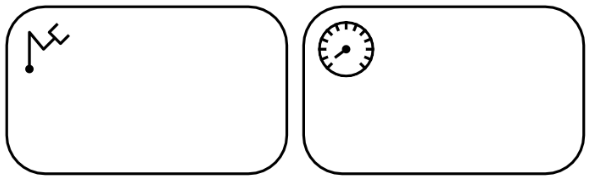

Figure 2.4: Proposed stencils for modelling of *IoT actuating tasks* and *IoT sensing task*, respectively.

subclasses of class *Task*; *SensingTask* for *IoT sensing task* and *ActuationTask* for *IoT actuating task*. Extensions to graphical model are presented in form of two new stencils for modelling sensing and actuating tasks.

Stencils for *IoT sensing task* and *IoT actuating task* as proposed in [30] are shown in figure 2.4. Stencil for sensing task is marked with a symbol of a gauge and stencil for actuating task with a symbol of a robot arm.

Connecting IoT tasks with physical entities

In [30], authors define a new BPMN class *PhysicalAssociation*, derived from class *BaseElement* and analogous to class *DataAssociation*. It has two subclasses. First, class *ActuationAssociation* is a directed connection from an *IoT actuating task*, modelled with class *ActuationTask*, to a *physical entity*, modelled with class *PhysicalObject*, on which the modelled actuator acts on. It represents a flow of physical interaction. Second, class *SensingAssociation* is a directed connection from a *physical entity*, modelled with class *PhysicalObject*, to an *IoT sensing task*, modelled with class *SensingTask*. It represents a flow of physical information.

For modelling of physical associations in BPMN diagrams, authors in [30] propose using the same stencil (dotted directed arrow) as it is defined for a *DataAssociation*.

Future work of the same authors introduces a new class for modelling of *physical entity*. Instead of class *PhysicalObject*, class *PhysicalEntity* is used. Therefore we imply, that class *PhysicalAssociation* now refers to class

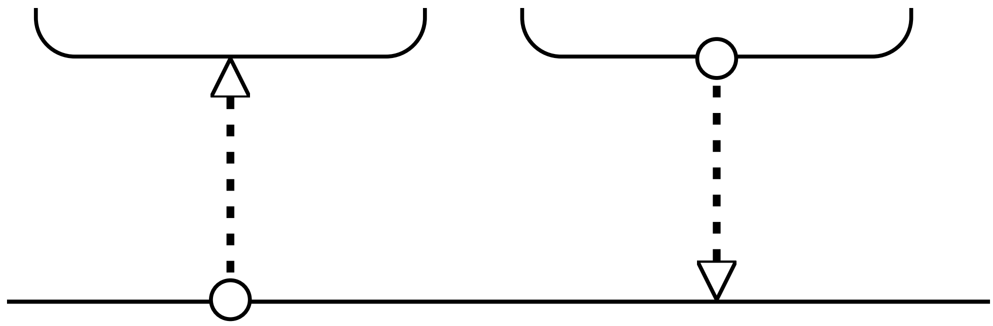

Figure 2.5: Proposed stencils for modelling of sensing association and actuating association, respectively.

PhysicalEntity instead of class *PhysicalObject*.

Proposed stencils for modelling *SensingAssociation* and *ActuationAssociation* are shown in figure 2.5.

Events

Authors in [4] analyse IoT characteristics relevant for modelling of real-world processes presented in [24] and summarized in 2.3.3. They focus on extending BPMN event element to support modelling IoT specific characteristics.

They propose extensions to *condition event*, *message event* and *error event* to support IoT use cases. They also propose a new type of event - *location event* for modelling of location changes of *physical entities*.

Presented state of the art modelling approaches are analysed and discussed in next chapter. Based on the analysis, we then describe open topics and propose research questions.

Chapter 3

Open topics and problem statement

This chapter first analyses the state of the art modelling approaches for modelling IoT characteristics described in section 2.3.4. Secondly, it identifies and describes three open research topics. Thirdly, based on presented open topics, we propose our research questions. Lastly, we describe methodology, which we use to find solutions for our research questions.

3.1 Analysis of the state of the art modelling approaches

In this section we analyse the state of the art modelling approaches for modelling of IoT characteristics of real-world processes presented in section 2.3.4. We identify modelling approaches, which we find suitable and which we use as a basis in this thesis. Performed analysis identifies open topics, which need to be addressed in order to provide better notations for modelling of IoT characteristics of real-world processes. Section concludes with a sample process, consisting of the state of the art modelling approaches, which we use as a basis in this thesis.

3.1.1 Modelling of *physical entity*

Modelling approach presented in 2.3.4 proposes modelling of *physical entity* with a graphical stencil similar to black box pool. BPMN metamodel is extended with a class *PhysicalEntity*.

Proposed approach is suitable for modelling of *physical entity*, as it enables user to model *physical entity* in a process diagram as a separate element. This element can then be addressed and interacted with by *IoT activities*. In this thesis we use this approach as a basis for modelling *physical entities*.

However, each *physical entity* can have multiple properties and we can be interested in more than one of them from the process modelling perspective. Proposed approach does not allow modelling of such properties. Further research is needed, in order to provide modelling approaches for solving described problem.

3.1.2 Modelling of *IoT device*

First modelling approach described in 2.3.4 proposes modelling of *IoT devices* as lanes in process diagrams. This enables users to clearly model one *IoT device* including all *IoT tasks*, for performing of which modelled device is responsible. We use this modelling approach as a basis for modelling *IoT devices* in this thesis.

However, modelling each *IoT device* with a separate lane may result in big, crowded and unreadable diagrams as the number of *IoT devices* in process diagram rises. In order to prevent that, further research is needed to provide a new way of modelling *IoT devices*.

3.1.3 Modelling of *IoT activity*

Modelling approach described in 2.3.4 proposes modelling of *IoT activities* with *IoT tasks*. New metamodel classes for *IoT sensing task* and *IoT actuating task* with corresponding graphical stencils are defined. Since BPMN tasks are used to model process activities, proposed approach is suitable for modelling of *IoT activities*. Sensing and actuating subtypes enable user to create clear and readable diagrams.

It is important that each *IoT task* has specified *IoT device*, which is responsible for its execution. Modelling of *IoT devices* as lanes enables clear and understandable way of defining *IoT device* responsible for execution of an *IoT task*.

In our work we use this modelling approach as a basis for modelling *IoT activities*.

3.1.4 Modelling of association between *IoT tasks* and *physical entities*

Modelling approach, described in 2.3.4, defines class *PhysicalAssociation* and graphical stencils, used for modelling of association between *IoT task* and *physical entity*.

Proposed modelling approach and stencils enable user to model interaction between *IoT task* and *physical entity*, including the direction of the association. We assess it as adequate for modelling of association between *IoT tasks* and *physical entities* and use it in this thesis.

3.1.5 Modelling of events

Proposed extensions to BPMN event element, presented in 2.3.4, are only the first steps in defining IoT-specific BPMN events. Presented solutions show the direction in which the research is going, but do not provide a complete and usable solutions.

Further research is needed, in order to provide extensions to BPMN's event element, which support identified IoT-specific characteristics. First step may be providing a modelling approach for modelling an event-based communication type, described in 2.3.2, with BPMN event elements.

3.1.6 Sample process

To better illustrate the state of the art modelling approaches analysed in this section, we model a sample real-world BPMN process. Modelled process consists of the state of the art modelling approaches, that we identified as suitable and that are used as a basis in the rest of the thesis.

A sample BPMN process is shown in figure 3.1. Process starts with a *start event* and then executes two *IoT tasks* in parallel. An *IoT sensing task* **sense temperature** measures the temperature of the *physical entity* **room** and is executed by *IoT device* **temperature sensor**. An *IoT sensing task* **sense window status** senses the state of the *physical entity* **window** and is executed by *IoT device* **window automation system**. Measured data is sent to decision management unit which executes a *decision task*, based on defined rules. Depending on *decision task* results, temperature adjustment may be executed, otherwise process ends with an *end event*. Temperature adjustment is performed in two steps. First, an *IoT device* **HVAC System** performs an *IoT actuating task* **adjust temperature** on *physical entity* **room**. Secondly, an *IoT device* **window automation system** performs an *IoT actuating task* **close window** on *physical entity* **window**. Process then ends with an *end event*.

A sample process illustrates current state in modelling of real-world processes and its shortcomings. The missing part is the support for modelling of properties of *physical entities* and event-based type of communication between *IoT devices* and a process engine. By modelling of each *IoT device* as a separate lane, process diagrams may become unnecessary big, crowded and unreadable. In next section, we focus on those problems and identify three open research topics, describe them, and based on them propose our research questions.

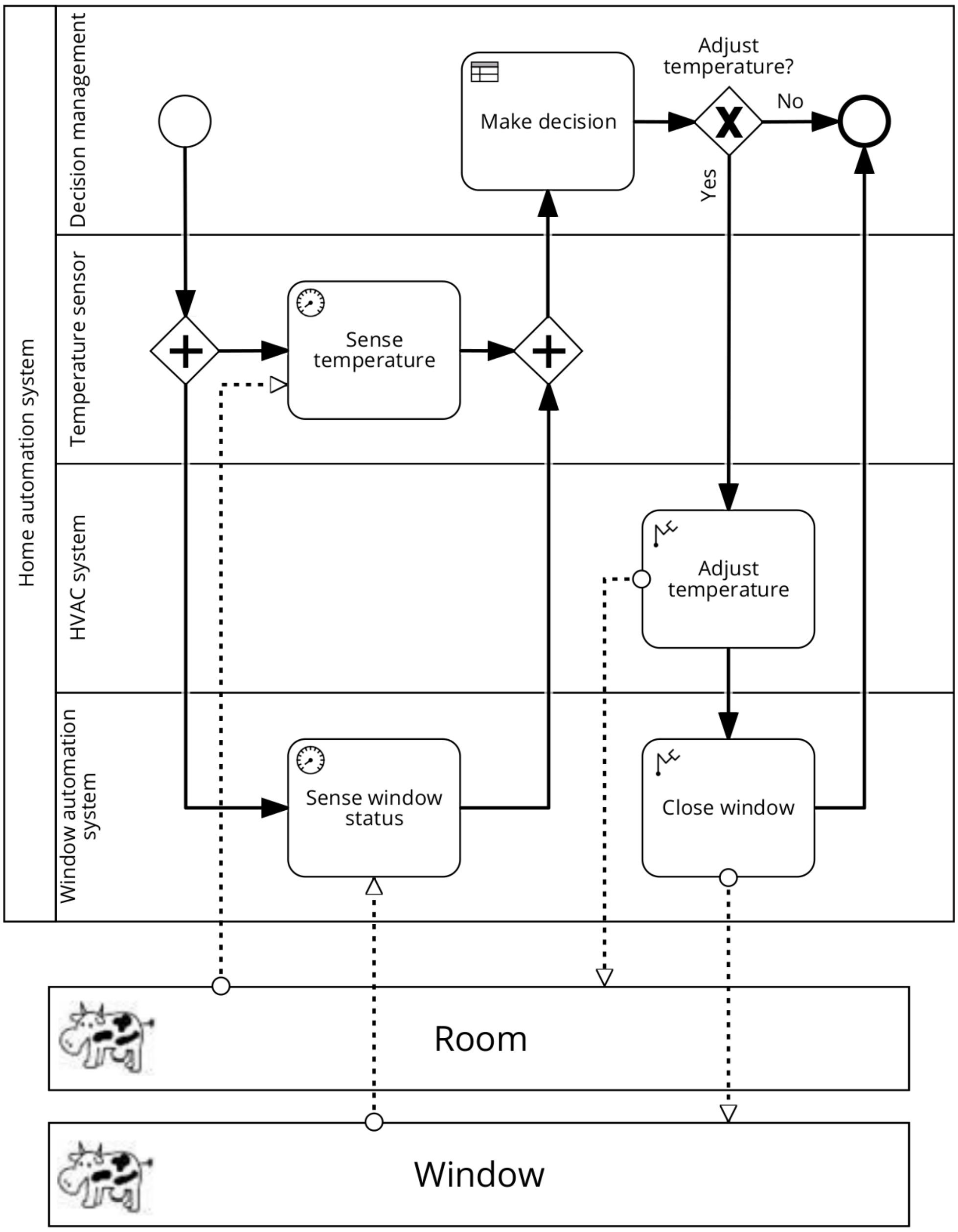

Figure 3.1: A sample BPMN process with the state of the art modelling approaches for IoT characteristics.

3.2 Open topics in the state of the art research

Based on the analysis of the state of the art modelling approaches performed in section 3.1, we identify three open topics which should be solved, in order to improve modelling of real-world processes. Open topics, that we base our research questions on, are presented in this section.

3.2.1 Modelling properties of *physical entity*

As analysed in section 3.1.1, each *physical entity* from real world has many different properties, that describe and define its state. So far, modelling of a *physical entity* has only been focused on an entity as a whole. We argue, that in order to support modelling of real-world processes, notations shall support modelling of different properties of *physical entity*.

Each *physical entity* may have many different properties defining it, but from the process modelling perspective, we are rarely interested in all of them. We define *property of interest* as a property of *physical entity*, that we are interested in from a process modelling point of view.

Lets take a look at an example from a home automation scenario. As mentioned before, an example of *physical entity* may be a **room**, that can have multiple *properties of interest* such as **temperature**, **air freshness**, **occupational status**, **luminance** etc. From a process modelling perspective, we may be interested in one, two or all of them.

In the state of the art modelling approaches, *properties of interest* can be modelled only implicitly. If *IoT sensing task* **measure temperature** is performed by *IoT device* **temperature sensor**, which is associated with *physical entity* **room**, it is implicitly stated, that we are interested in *property of interest* **temperature** of *physical entity* **room**. However, this approach do not provide a way to clearly and explicitly model *properties of interest* of a *physical entity* and may result in crowded and unreadable diagrams.

3.2.2 Improving the modelling of *IoT device*

Section 3.1.2 analyses the state of the art modelling approaches for modelling of *IoT device* and proposes modelling of *IoT device* with a lane as a basis in this thesis. However, concerns are raised, that such approach may result in crowded and unreadable diagrams when modelling a high number of *IoT devices*.

A sample process, described in section 3.1.6, shows that already with only three *IoT devices*, process diagram can become quite big and clumsy. We believe, that better approach shall be defined.

Modelling approach for modelling of *IoT devices* shall provide notation more flexible as lanes, which would support creation of clean and readable diagrams, independently of number of modelled *IoT devices*.

3.2.3 Modelling of an event-based communication between *IoT devices* and a process engine

When modelling IoT characteristics of real-world processes in BPMN, we have to think about types of communication used between *IoT devices* and a process engine. Section 2.3.2 describes two types of communication used in IoT, a request/response communication paradigm and an event-based communication. Section 3.1.5 then provides the idea of using BPMN events for modelling of event-based communication.

When performing *IoT actuating activities*, we always talk about request/ response, or even only request type of communication. When we want for application to perform a certain *IoT actuating activity*, process engine sends a request to *IoT device*, which may return a response.

With *IoT sensing activities*, we have to consider both request/response and event-based communication. When we want for an application to perform an *IoT sensing activity* in a certain point in time, we use a request/ response communication type. Process engine sends a sensing request and *IoT device* returns a result.

An event-based communication type is used, when an *IoT device* is providing sensing results without previous requests. Such *IoT device* is sending sensing results in certain events, such as specified time intervals (e.g. send sensed temperature every 10 minutes), or when certain conditions are satisfied (e.g. send sensed temperature, if sensed temperature is higher than 25° Celsius).

Modelling of event-based communication between *IoT devices* and a process engine is currently not supported in BPMN or the state of the art modelling approaches. Process engine should be made aware of communication type used between *IoT devices* and a process engine, therefore this information have to be included in process diagram. We argue, that in order to support modelling of real-world processes, notations shall support modelling of an event-based communication type between *IoT devices* and a process engine.

As mentioned in section 3.1.5, BPMN events may be a good start for modelling of an event-based communication type, since the moment, when a process engine receives a message from an *IoT device*, can be described as an event.

Next section proposes our research questions, based on the open topics described in this section.

3.3 Research questions

Based on the open topics described in 3.2, we define our research questions. Questions are focused on three aspects; modelling of *properties of interest*, improving the modelling of *IoT device*, and modelling of an event-based communication type between *IoT devices* and a process engine.

We merge those aspects together and get a problem of modelling of an *IoT activity* performed by a specified *IoT device* on identifiable *property of interest* of a *physical entity*. Meanwhile we also provide support for modelling of an event-based communication type used between *IoT devices* and a process engine.

Our proposed research questions are:

- How to provide support for modelling the *properties of interest* of a *physical entity*?
- How to improve the modelling of *IoT device* responsible for performing *IoT activities*?
- How to provide the support for modelling of an event-based communication type between *IoT devices* and a process engine?

Methodology, which we use for solving our research questions, is described in next section.

3.4 Methodology

To solve our research questions, we follow methodology proposed in this section. First, we define requirements for the proposed solutions. Secondly, we propose solutions for solving our research questions. Lastly, we evaluate proposed solutions against defined requirements.

The rest of the thesis is structured in chapters describing individual steps of proposed methodology.

3.4.1 Defining requirements

We try to solve our research questions by proposing different modelling approaches and corresponding extensions to BPMN. Before proposing solutions, we first define a set of requirements, which shall be addressed and satisfied by proposed solutions, in order to successfully solve our research questions.

Defined requirements guide us through the process of proposing possible solutions and help us evaluate them.

3.4.2 Proposing solutions

Process of solving our research questions continues by proposing solutions. In order to develop solutions, we identify possible approaches for modelling business processes, which solve our research questions.

Some proposed approaches can be modelled using standard BPMN notation or modelling approaches presented in the state of the art research work. In some cases, we need to propose extensions to BPMN. Proposed extensions may extend existing BPMN elements or define new ones.

Identified modelling approaches, together with proposed extensions, form our proposed solutions.

3.4.3 Evaluating proposed solutions

In the process of proposing solutions for our research questions, we try to address as many defined requirements as possible. However, all solutions cannot address all requirements. Therefore, proposed solutions are evaluated against requirements. Proposed solution, which satisfies the highest number of requirements, is our final solution.

In next chapter we perform the first step of our methodology, i.e. we identify and define requirements for proposed solutions.

Chapter 4

Requirements for proposed solutions

In this chapter we first describe the process of identifying requirements for proposed solutions to our research questions. Secondly, we define four sets of requirements, which later help us develop and evaluate proposed solutions.

4.1 Identifying requirements

We propose solutions for our research questions in form of different possible modelling approaches for modelling of real-world processes, together with necessary extensions to BPMN. Requirements, defined in this chapter, guide us through the process of developing proposed solutions and later help us evaluate them.

Defined requirements are influenced by works [21] and [33] where authors propose requirements for defining a new modelling element for *physical entity*, and for modelling of WSANs, respectively. Proposed requirements from related work are taken as a basis and then modified to provide guidance through the process of solving our research questions.

We identify and define four sets of requirements, which are based on aspects of modelling business processes, validity of created diagrams, process execution and process of extending BPMN. Defined requirements represent guidelines and limitations in order for us to develop appropriate and adequate solutions for our research questions.

First set of requirements supports the process of modelling business processes. Modelling of business processes is usually performed in two phases. First, domain expert develops a descriptive process model, that considers the process flow. Secondly, more technical system developer implements the process, including all execution details [33] [24]. In the implementation part of developing a real-world process, system developer specifies *IoT services* responsible for performing *IoT activities* and other implementational details, needed for process engine to successfully execute business process.

Second set of requirements is based on the aspect of process validity. Process diagrams, modelled using proposed extensions, should be valid, therefore we define requirements supporting validity of modelled business processes. Requirements from this set focus mainly on the relationships between the newly defined elements.

Third set of requirements is based on the aspect of process execution. Proposed extensions should be adequate for execution in process execution engines. Process execution is based on *IoT services*, which are responsible for performing *IoT activities*. For each *IoT activity* in the process, specified *IoT service* is invoked by execution engine.

Fourth set of requirements is based on the aspect of extending BPMN. Since version 2.0, BPMN supports custom extensions. Proposed extensions should use BPMN's support for extensions wherever possible. If deeper changes to the BPMN metamodel are needed, metamodel shall be changed as little as possible.

All four sets of requirements are defined and described in next section.

4.2 Defined requirements

In this section, we define four sets of requirements, based on the identification of requirements in previous section. Defined requirements guide us through the process of proposing solutions and help us evaluate them.

Each requirement is stated in a bold font, followed by its rationale.

4.2.1 Requirements supporting process modelling

Proposed solutions shall define modelling approaches and extensions to BPMN that:

- **Provide a way to create uncrowded and readable diagrams.** According to scalability characteristics, described in section 2.3.3, created diagrams shall be simple and understandable, even with high number of *physical entities* and *IoT devices*.

- **Provide a way of representing an isolated *property of interest* of a *physical entity*.** Process modeller must be able to model each *property of interest* of *physical entity* in a clear and understandable way.

- **Provide a way to model availability of *property of interest* for different types of *IoT activities*. *Property of interest* can be available for *sensing*, *actuating*, or both.** Depending on the available IoT infrastructure, certain *properties of interest* may be available only for one kind of interaction.

- **Provide a way to associate *property of interest* with a *physical entity*.** *Properties of interest* are not a self-standing elements and must be associated with a *physical entity*.

- **Provide a way to associate *property of interest* with an *IoT activity*. The direction of association must be expressible.**

IoT devices are performing *IoT activities* on *properties of interest* of *physical entity*, therefore such association must exist.

- **Provide support for one *property of interest* having multiple associations to different *IoT activities*.** Each *property of interest* may be accessed by more than one *IoT activity*.
- **Provide a way to model an event-based communication type for *IoT sensing activities*.** As explained in section 3.2.3, *IoT sensing activities* may use an event-based type of communication.
- **Provide a better way to model an *IoT device* responsible for execution of an *IoT activity* in comparison to the state of the art modelling approaches.** Each *IoT activity* must have defined *IoT device* responsible for its execution. As described in section 3.2.2, modelling of *IoT devices*, as described in the state of the art research, shall be improved.
- **Include a set of predefined *properties of interest*, that are common with all *physical entities* (e.g. ID tag, temperature, location, owner etc. [6]).** A predefined set of *properties of interest*, available for modeller in modelling tool, would simplify and improve efficiency of process modelling.

4.2.2 Requirements supporting validity of modelled processes

Proposed solutions shall define modelling approaches and extensions to BPMN which ensure that:

- **Each modelled *physical entity* in process diagram is associated with at least one modelled *property of interest*.** Process diagram is valid, if each *physical entity* have at least one *property of interest*. *Physical entity* without *properties of interest* cannot be acted upon and is therefore unnecessary and invalid.

- **Each modelled *property of interest* is associated with exactly one *physical entity.*** *Physical entity* with all its *properties of interest* is a separate entity, which can be reused. Each *property of interest* is therefore part of exactly one *physical entity*. *Properties of interest* without associated *physical entity* are meaningless and therefore invalid.

4.2.3 Requirements supporting process execution

Proposed solutions shall define modelling approaches and extensions to BPMN that:

- **Provide a way to define an *IoT service* responsible for execution of an *IoT activity.*** Each *IoT activity* must have an *IoT service* defined in order to be executed by process engine.

- **Provide a way to define the type of communication used between *IoT devices* and a process engine, when performing *IoT activities.*** Process engine have to be aware of possible event-based communication type in order to wait for incoming messages from *IoT devices*.

4.2.4 Requirements supporting process of extending BPMN

Proposed solutions shall define modelling approaches and extensions to BPMN that:

- **Respect semantics of existing BPMN elements.** Newly defined elements shall be intuitive and easy to use for an experienced BPMN modeller.

- **Make use of BPMN's extension mechanism whenever possible.** BPMN's extension mechanism, described in section 2.2.3, provides a native support for defining extensions and shall be used wherever possible.

- **Minimize the number of changes to existing BPMN metamodel.** If extension mechanism cannot be used, the number of changes to existing BPMN metamodel shall be minimal in order to make the implementation of new extensions as simple as possible.

4.2.5 Summary of proposed requirements

We can summarise proposed requirements as follows: *User* must be able to model, which *IoT device* is responsible for performing which *IoT activity* on which *property of interest* of which *physical entity.* For *IoT sensing activities*, *user* must be able to model an event-based communication type between *IoT devices* and a process engine, when applicable.

Based on the requirements defined in this chapter, in next chapter we propose a set of solutions for solving stated research questions.

Chapter 5

Proposed solutions

In this chapter we identify and propose solutions for our research questions. Solutions are proposed in form of different possible modelling approaches, which try to solve our research questions. Each modelling approach may include extensions to BPMN, if necessary. Proposed solutions are evaluated in chapter 6.

Based on our research questions, we try to identify different possible ways to model *properties of interest* of a *physical entity* and an event-based communication type, and improve modelling of *IoT devices*.

We are proposing six different solutions, each one consisting of a proposed modelling approach with corresponding extensions to BPMN. Proposed solutions are described in more detail in the following sections.

5.1 Modelling a single *IoT device* as a single lane

This solution is based on the state of the art research presented in [22] and [21] and analysed in section 3.1.

5.1.1 Proposed modelling approach

In this modelling approach, *physical entity* is represented with a class *PhysicalEntity*, introduced in [21], and modelled with the same graphical stencil as a black-box pool with an additional marker. Each *IoT device* is modelled as a separate lane. Lane contains *IoT activities* modelled as *IoT tasks* that are executed by modelled *IoT devices*. Association between *IoT task* and *physical entity* is modelled using *PhysicalAssociation*, described in section 2.3.4 and analysed in section 3.1.4.

Modelling of *properties of interest* is in this case implicit. *Property of interest* is specified as a combination of *IoT device*, *IoT task* and *physical entity*. If a temperature sensor is performing sensing task on room, *property of interest* is temperature of a room.

For modelling of a request/response type of communication between *IoT devices* and a process engine, we propose using *IoT tasks* connected with the sequence flow. Sequence flow represents execution of BPMN tasks in sequential order and is therefore suitable for modelling of request/response type of communication.

An example of a process with an *IoT sensing activity* using a request/ response type of communication is shown in figure 5.1. Process, modelled in figure, starts with a start event and continues with two parallel *IoT sensing tasks*, **sense temperature**, performed by *IoT device* **temperature sensor**, on *physical entity* **room** and **sense window status** (it can either be opened or closed), performed by *IoT device* **window automation system**, on *physical entity* **window**, using a request/response type of communication. Next, decision management performs a decision task based on predefiend rules. If temperature adjustment is not needed, process ends. Otherwise, *IoT device* **HVAC system** performs an *IoT actuating task* **adjust temperature** on *physical entity* **room**, then *IoT device* **window automation device** performs an *IoT actuating task* **close window** on *physical entity* **window**. Process ends with an end event.

For modelling of an event-based type of communication between *IoT de-*

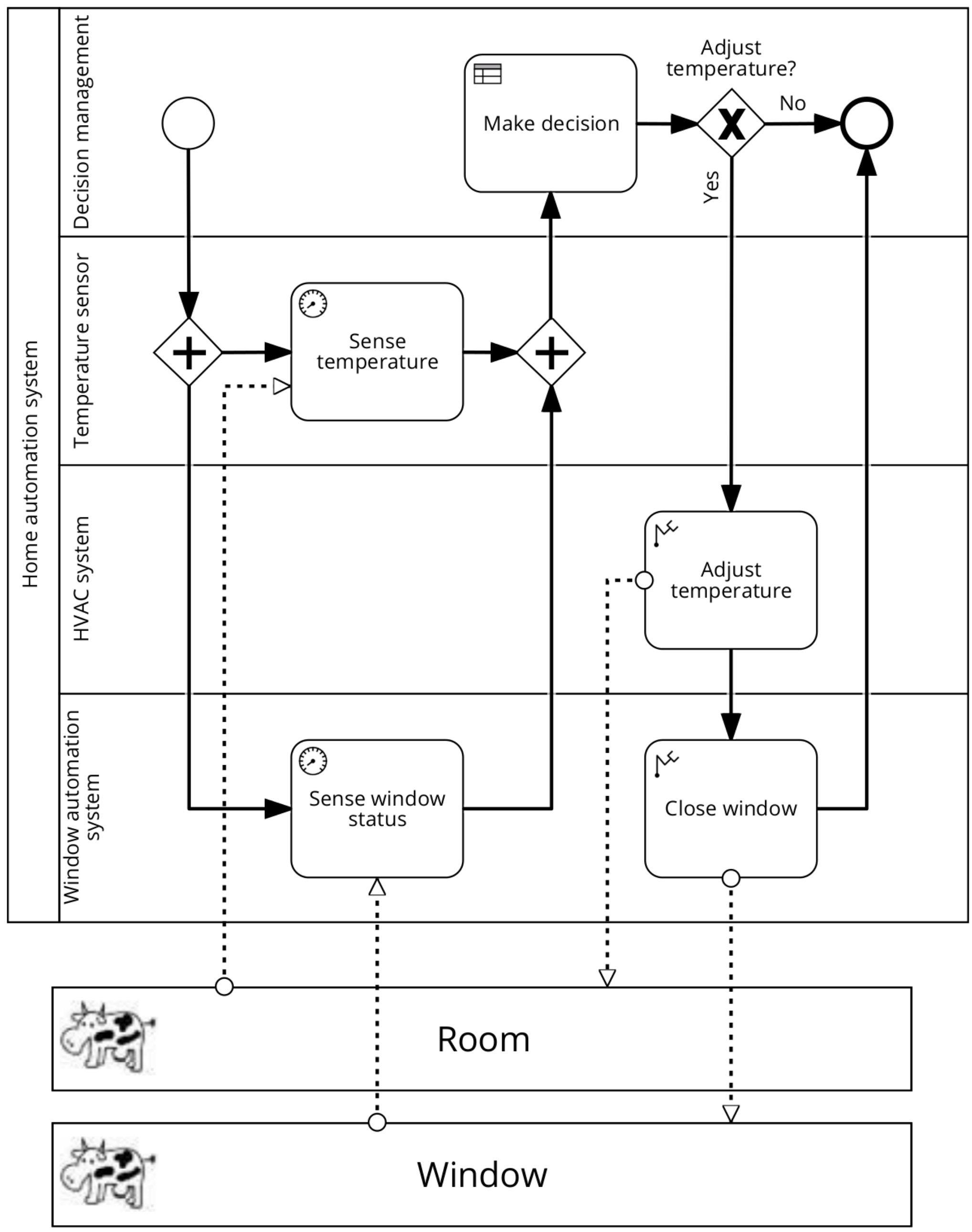

Figure 5.1: Proposed solution for modelling each *IoT device* as a single lane, using a request/response communication type.

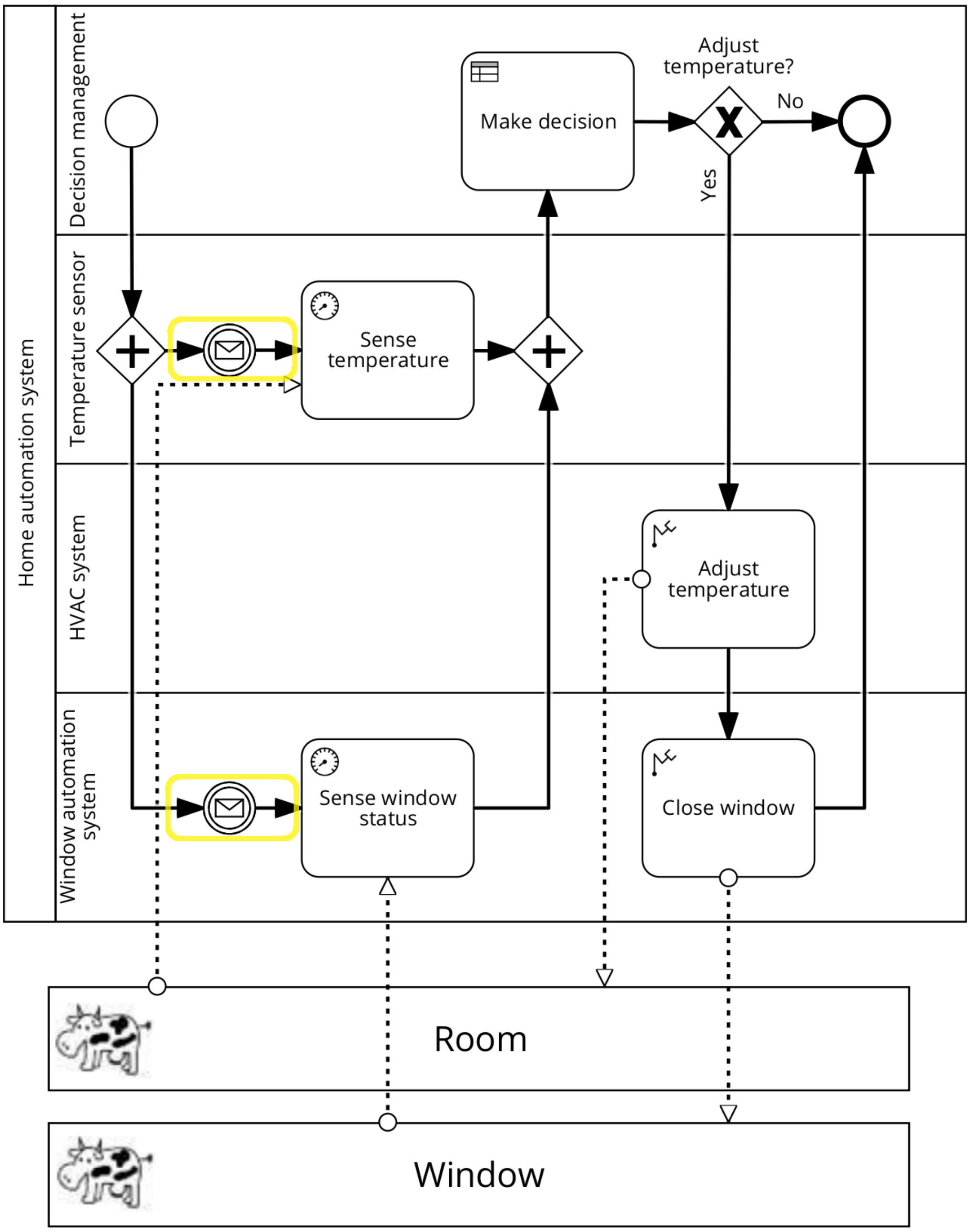

Figure 5.2: Proposed solution for modelling each *IoT device* as a single lane, using an event-based communication type.

vices and a process engine, we propose using BPMN events. More particularly, we are interested in an event that occurs when an *IoT device* performs an *IoT activity* and sends measured data to process engine. We propose using an *intermediate catching message event* to model a point in process, where process engine is waiting to receive measured data from an *IoT device*.

An example of a process with an *IoT sensing activity* using an event-based type of communication is shown in figure 5.2. Process, modelled in figure, is similar that the one with a request/response communication shown in figure 5.1. The only difference[1] is an *intermediate catching message event* before each *IoT sensing task*. It is used to model the point in process, where process waits for an *IoT device* to send data to process engine.

5.1.2 Proposed extensions to BPMN

According to requirements supporting process execution, user has to be able to specify an *IoT service* responsible for execution of an *IoT activity* and a type of communication used, in order to enable execution of defined processes in process execution engines. Therefore, we extend classes *SensingTasks* and *ActuatingTasks*, defined in work [30].

We extend both classes with two additional attributes; *iotServiceRef* and *communicationType*. Attribute *iotServiceRef* is used to specify an *IoT service*, responsible for execution of *IoT activity*. Attribute *communicationType* is used to specify the type of communication used by *IoT activity*. Additional attributes are defined using BPMN's extension mechanism.

We use extended definitions of two classes for all proposed solutions in the thesis.

5.2 Modelling all *IoT devices* as a single lane

This solution is based on solution presented in previous section (5.1). We try to reduce the complexity of modelled diagrams by modelling all *IoT devices*

[1] The difference is on the figure marked with a yellow square.

with a single lane.

5.2.1 Proposed modelling approach

In this approach, we propose modelling of all *IoT devices* using one process lane. In this case, performer is one *IoT device* from a pool of all *IoT devices* of an application. In comparison to previous approach, we lose information about the *IoT device* performing the *IoT activity* and have to model it some other way.

Property of interest can still be stated only implicitly as a combination of an *IoT task* and a *physical entity.* If an *IoT task* **measure temperature** is performed on *physical entity* **room**, *property of interest* is **temperature** of *physical entity* **room**.

In order to model an *IoT device* responsible for execution of an *IoT task*, we propose adding an annotation with device's name to the graphical stencil representing physical association between *IoT task* and *physical entity.* We extend class *PhysicalAssociation* later in this section.

A request/response type of communication is modelled with a sequence flow, the same way as in solution from section 5.1. An example of a process with a request/response type of communication is shown in figure 5.3. Process is the same as in solution from section 5.1. The only difference is modelling of *IoT devices*, which are in this case joined in one lane and modelled with an annotation on stencils representing physical association.

For modelling of an event-based type of communication we use events, the same way as in solution from section 5.1. An *intermediate catching message event* is used for modelling a point in a process, where process engine waits to receive sensed data from an *IoT device.* An example of a process using an event-based type of communication is shown in figure 5.4. Again, process, modelled in figure, is similar that the one with a request/response communication shown in figure 5.3. The only difference[2] is an *intermediate catching message event* before each *IoT sensing task.*

[2]The difference is on the figure marked with a yellow square.

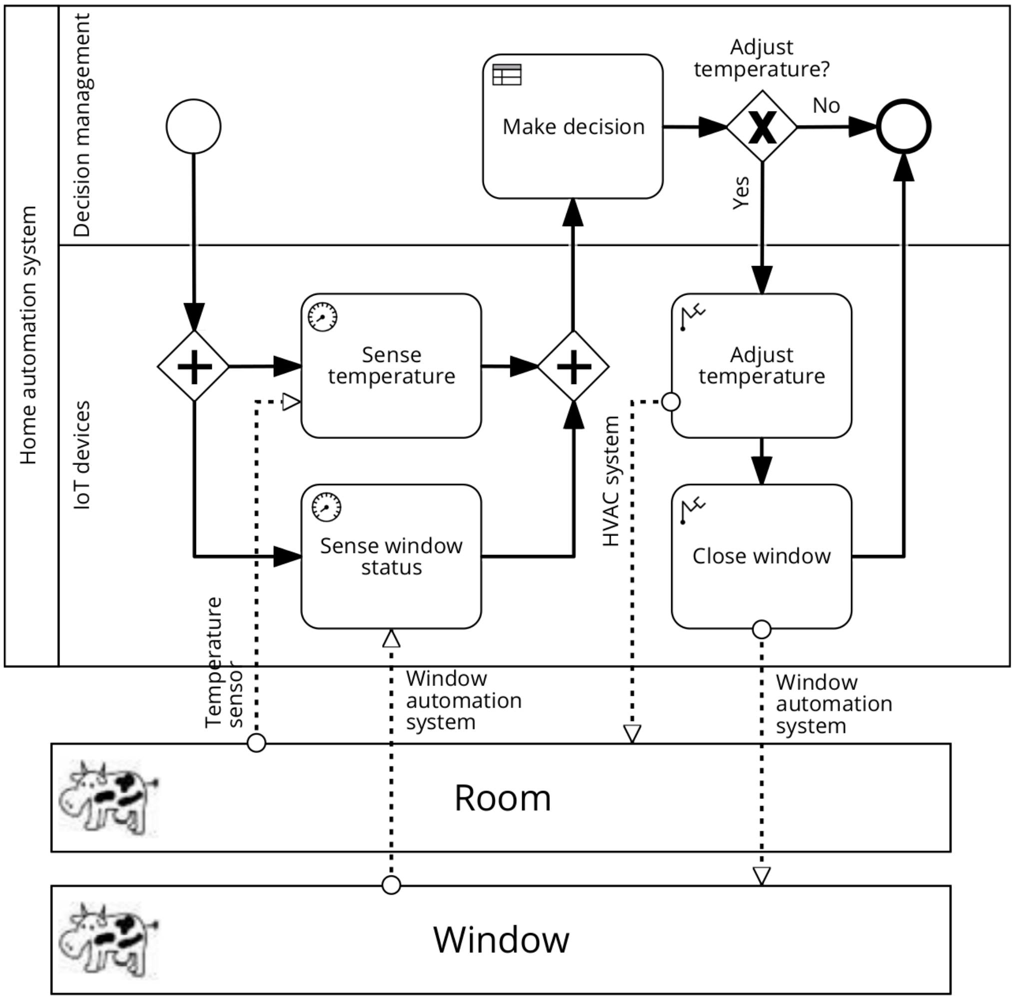

Figure 5.3: Proposed solution for modelling all *IoT devices* as a single lane, using a request/response type of communication.

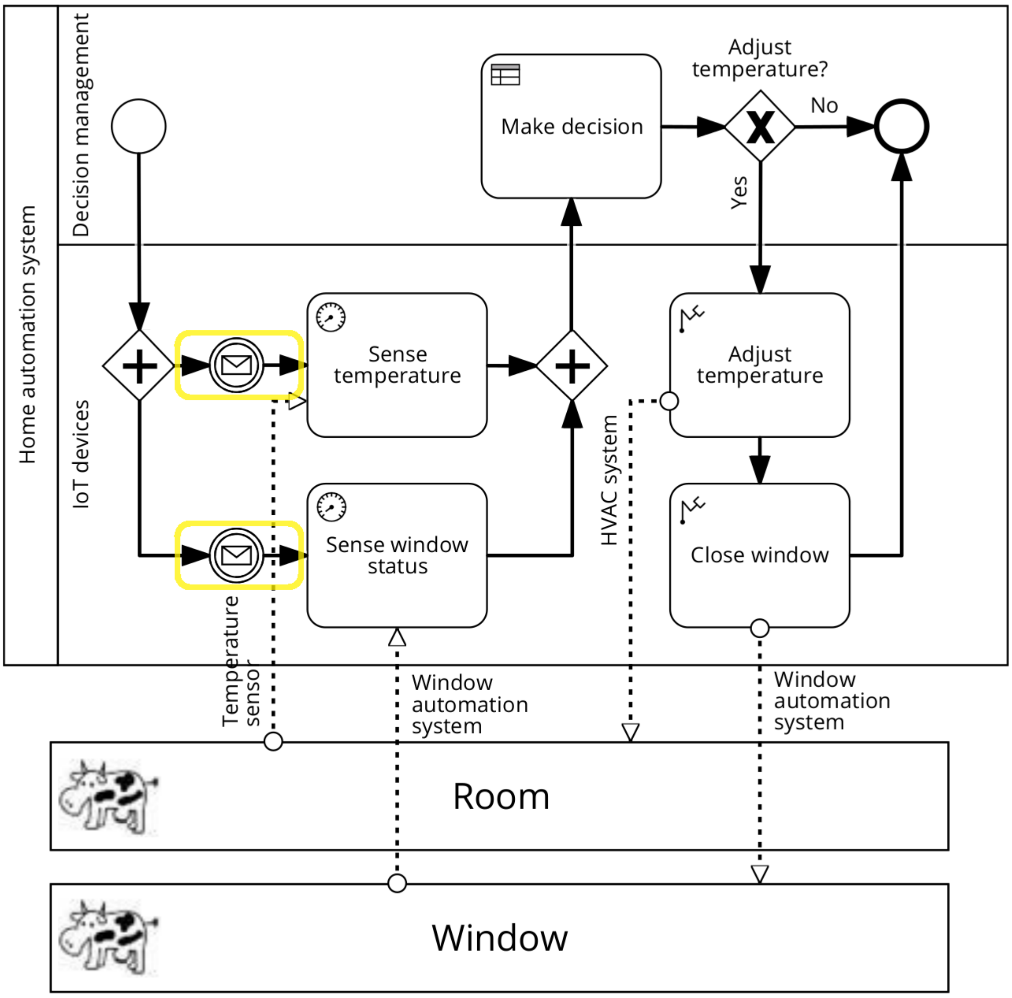

Figure 5.4: Proposed solution for modelling all *IoT devices* as a single lane, using an event-based type of communication.

5.2.2 Proposed extensions to BPMN

As stated before, by modelling of all *IoT devices* with one lane, we lose information about an *IoT device* performing specific *IoT activity*. We propose modelling a responsible *IoT device* in process diagram with an annotation on a stencil representing physical association. In order to do so, we have to extend class *PhysicalAssociation*.

We extend class *PhysicalAssociation* with an attribute *iotDeviceRef*. Attribute *iotDeviceRef* identifies an *IoT device* responsible for executing the *IoT task*, associated with an instance of *PhysicalAssociation*. Attributes are defined with BPMN's extension mechanism.

We use extended definitions of the class *PhysicalAssociation* for all solutions proposed further in the thesis.

5.3 Modelling sensors and actuators as two lanes

This solution is based on solution presented in previous section (5.2). We try to improve the readability of diagrams, while still trying to reduce the complexity in comparison with solution from section 5.1. We propose using two lanes for modelling of *IoT devices*.

5.3.1 Proposed modelling approach

In this approach we use two lanes for modelling of *IoT devices*; one for modelling sensors and one for modelling actuators. The division of *IoT devices* to sensors and actuators in process diagram may improve the readability of the diagram. This modelling approach generates slightly more complex diagrams than solution from section 5.2, but is far less complex than solution from section 5.1.

Property of interest is again stated only implicitly, the same way as in solution 5.2.

For modelling of an *IoT device* responsible for performing an *IoT activity*, we use the approach with an extended class *PhysicalAssociation* presented in section 5.2.

A request/response type of communication is modelled with a sequence flow, the same way as in solutions from sections 5.1 and 5.2. An example of a process with a request/response type of communication is shown in figure 5.5. Process is similar to process from previous section, shown in figure 5.3. The main difference in this solution is that we split the lane with all *IoT devices* into two lanes, one for sensors and one for actuators.

For modelling of an event-based type of communication we use events, the same way that in solutions 5.1 and 5.2. An example of a process using an event-based type of communication is shown in figure 5.6. Process, shown in figure, is similar that the one showed for a request/response type of communication. The only difference[3] is an *intermediate catching message event* placed before each *IoT sensing task*.

5.3.2 Proposed extensions to BPMN

Presented modelling approach do not require any new extensions to BPMN other than the ones already defined in this chapter. Process diagrams can be modelled with standard BPMN elements, elements from state of the art modelling approaches, analysed in section 3.1, and classes extended in this chapter.

5.4 Modelling a set of *IoT devices* as a pool

This solution departs from the approach of modelling *IoT devices* as lanes and instead proposes modelling of a set *IoT devices* as a pool.

[3]The difference is on the figure marked with a yellow square.

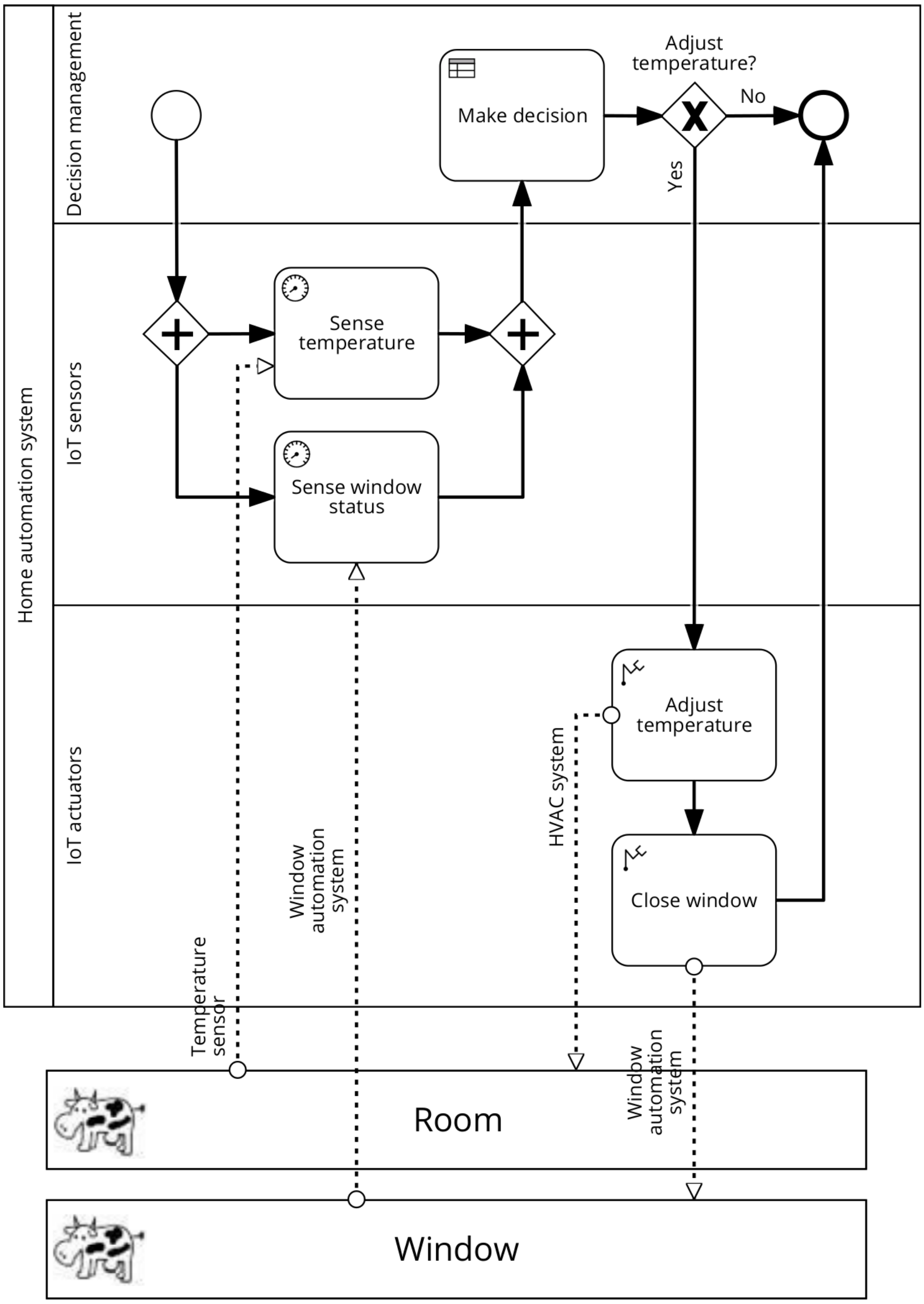

Figure 5.5: Proposed solution for modelling sensors and actuators with two lanes, using a request/response type of communication.

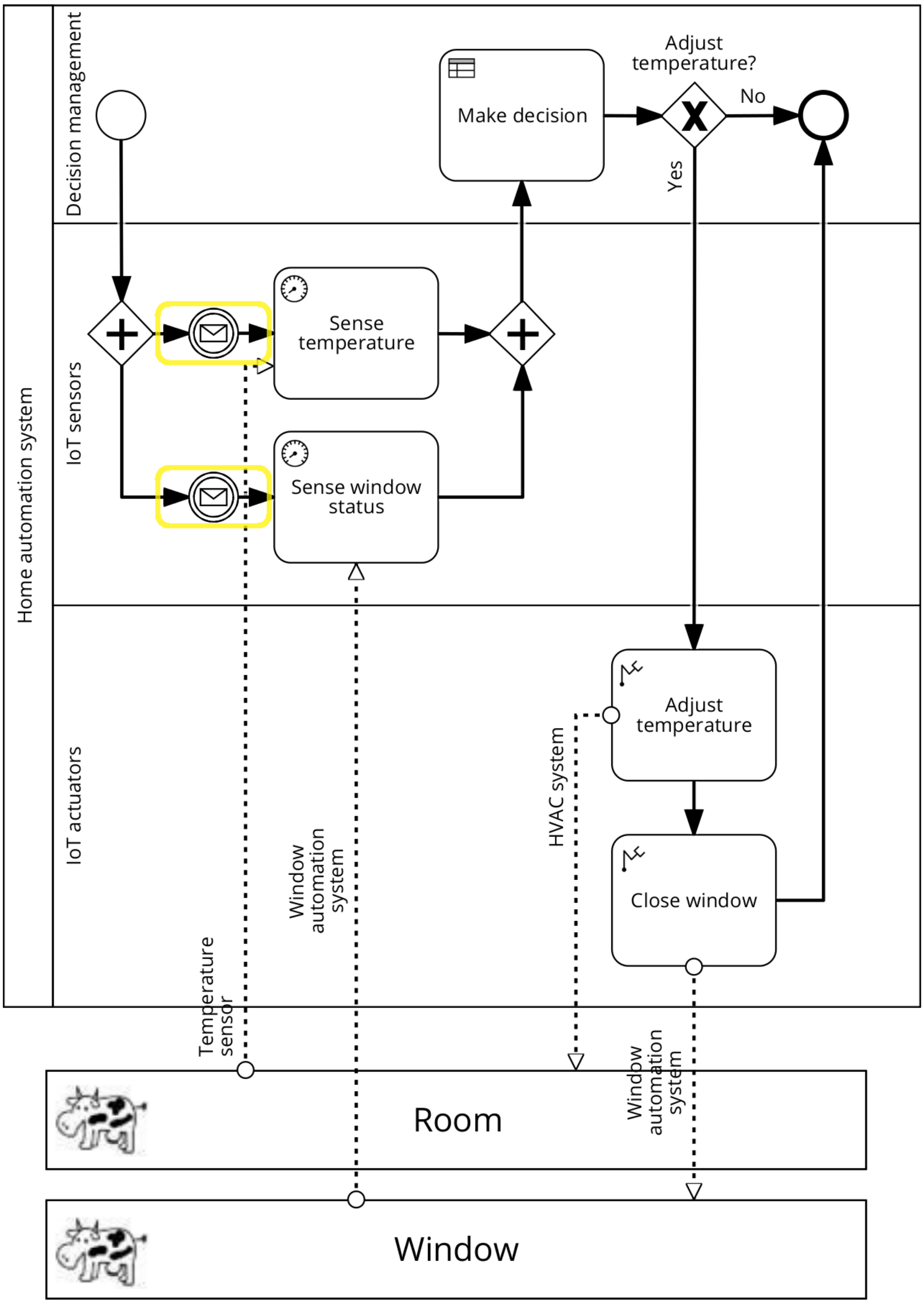

Figure 5.6: Proposed solution for modelling sensors and actuators with two lanes, using an event-based type of communication.

5.4.1 Proposed modelling approach

Idea of this modelling approach is to model a set of *IoT devices* as a pool. Such pool may represent all *IoT devices* of an IoT application or just a specific subset, e.g. a set of *IoT devices* connected to the same *IoT gateway*.

This approach enables user to model organizational structure of *IoT devices* as installed in real world. For example, home automation system may consist of a decision managements system, *IoT gateway*, that connects all *IoT devices* inside the house, and *IoT gateway*, for *IoT devices* outside the house. In such scenario, user may model decision management system using one pool, all *IoT devices* inside the house using second pool, and all *IoT devices* outside the house using third pool.

This approach is similar to approach 5.2, but it enables user to create structured diagrams that are better representing the structure of *IoT devices* from real world.

Property of interest is again stated just implicitly, the same way as in previous solutions.

For modelling of an *IoT device* responsible for performing an *IoT activity*, we use the approach with an extended class *PhysicalAssociataion* as presented in section 5.2.

Modelling of *IoT devices* as a pool comes with a limitation, i.e. sequence flow cannot cross the boundaries of a pool. This restriction forces us to propose new approach for modelling a request/response type of communication.

A request/response type of communication is still modelled with a sequence flow, but we have to use messages to model communication flow between pools. An example of a process with a request/response type of communication is shown in figure 5.7. Process, shown in figure, starts with a start event and then sends two messages to *IoT devices*, requesting measured data. *IoT device* **temperature sensor** performs an *IoT task* **sense temperature** on *physical entity* **backyard**. Similarly, different *IoT device* **temperature sensor** performs an *IoT task* **sense temperature** on *physical entity* **room**. Measured data is then sent to decision management system. If

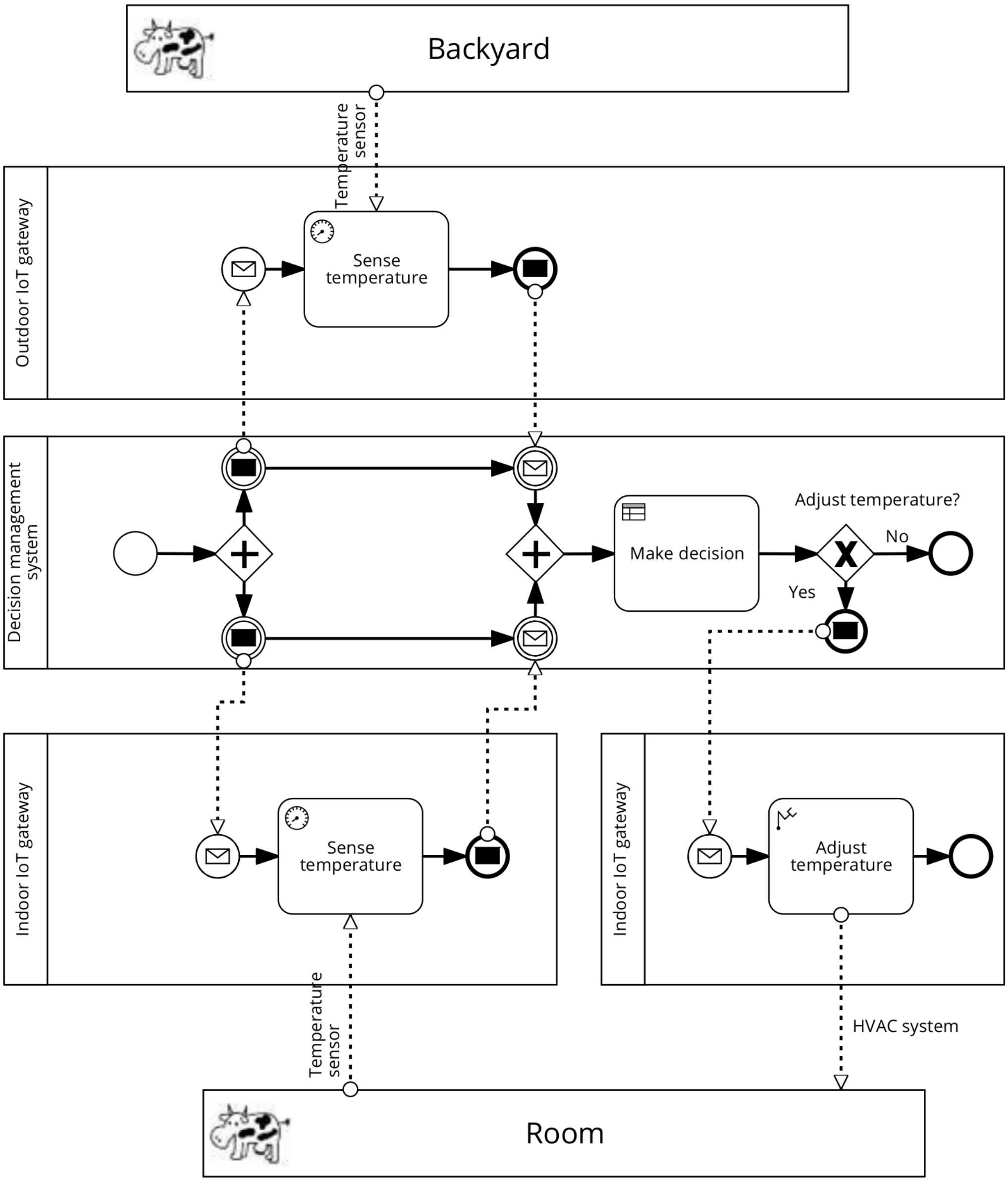

Figure 5.7: Proposed solution for modelling of a set of *IoT devices* as a pool, using a request/response type of communication.

temperature has to be adjusted, message is sent to *IoT device* **HVAC system**, which performs an *IoT task* **adjust temperature** on *physical entity* **room**. Otherwise, process ends with an end event.

For modelling of an event-based type of communication, we use *timer events* and *conditional events*, meaning that *IoT sensing activities* can start based on timer or other stated condition. An example of a process using an event-based type of communication is shown in figure 5.8. Process, shown in figure, is similar that the one showed for a request/response type of communication. Difference is, that we no-longer have to model sending requests from decision management system to *IoT devices*. Instead, *IoT sensing activities* are triggered by *timer event* or by *conditional event*. Process then performs an *IoT sensing task* and sends measured data to decision management system.

5.4.2 Proposed extensions to BPMN

No extensions are needed for described modelling approach. It can be modelled with standard BPMN elements.

5.5 Modelling event-based sensing with events

This solution is based on solution presented in 5.2. It focuses only on modelling of *IoT sensing activities* using an event-based communication type. We abandon the idea of using *IoT tasks* to model an event-based *IoT sensing activities*. Instead, we propose IoT-specific events.

5.5.1 Proposed modelling approach

We propose modelling of *IoT actuating activities* and *IoT sensing activities*, using a request/response communication type, the same way as in 5.2, that is with *IoT tasks*. For modelling of *IoT sensing activities*, using an event-based communication type, we propose using a newly defined *IoT sensing*

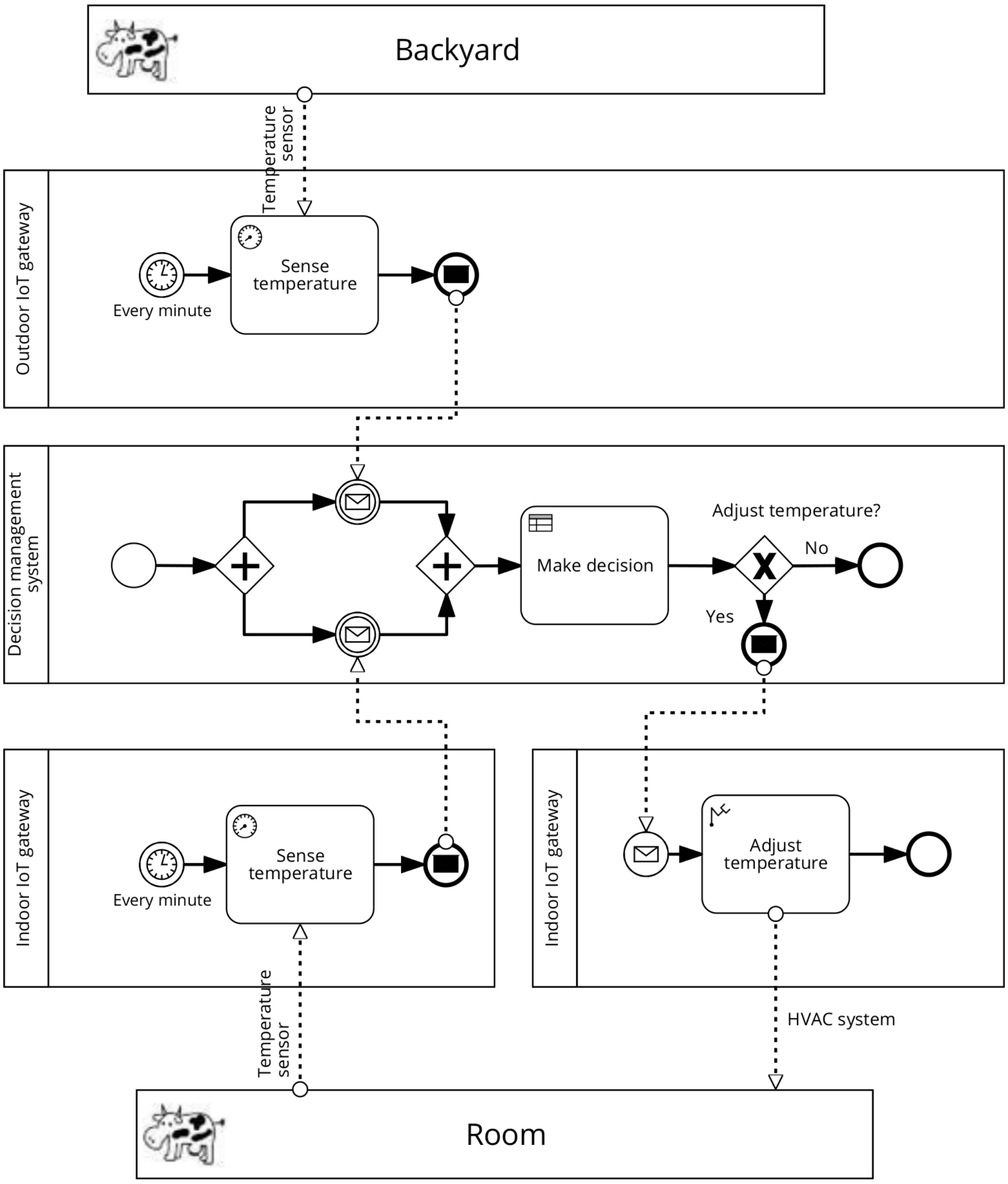

Figure 5.8: Proposed solution for modelling of a set of *IoT devices* as a pool, using an event-based type of communication.

intermediate event.

An *IoT sensing intermediate event* is a newly defined IoT-specific event, used for modelling of *IoT sensing activities* using an event-based communication type. It represents a point in the process, where process waits for *IoT device* to perform an *IoT sensing activity* and to send sensed data to process engine.

We propose a graphical stencil, standard for BPMN intermediate events, i.e. a circle drawn with a double thin line. Within a circle is a symbol of a gauge, same as used for *IoT sensing task*. Graphical stencil shall be annotated with a *property of interest*, which *IoT device* is measuring.

In this modelling approach, modelling of *properties of interest* is done with annotations on an *IoT sensing intermediate event*. It enables user to specify *property of interest* from event point of view, but not to explicitly define all *properties of interest* of a *physical entity*.

An example of a process using *IoT sensing intermediate events* is shown in figure 5.9. Process, shown in figure, starts with a start event and proceeds with two parallel *IoT sensing intermediate events*, waiting for measured *property of interest* **temperature** of *physical entity* **room**, measured by *IoT device* **temperature sensor**, and *property of interest* **status** of *physical entity* **window**, measured by *IoT device* **window automation system**. Process then continues with a decision task. If temperature adjustment is needed, *IoT device* **HVAC system** performs an *IoT actuating task* **adjust temperature** on *physical entity* **room**, then *IoT device* **window automation device** performs an *IoT actuating task* **close window** on *physical entity* **window**. Process ends with an end event.

5.5.2 Proposed extensions to BPMN

In presented modelling approach we define a new BPMN event *IoT sensing intermediate event*. Here we extend BPMN metamodel by defining a subclass *SensingEventDefinition* to class *EventDefinition*. It inherits the attributes and model associations of class *BaseElement* through its relationship

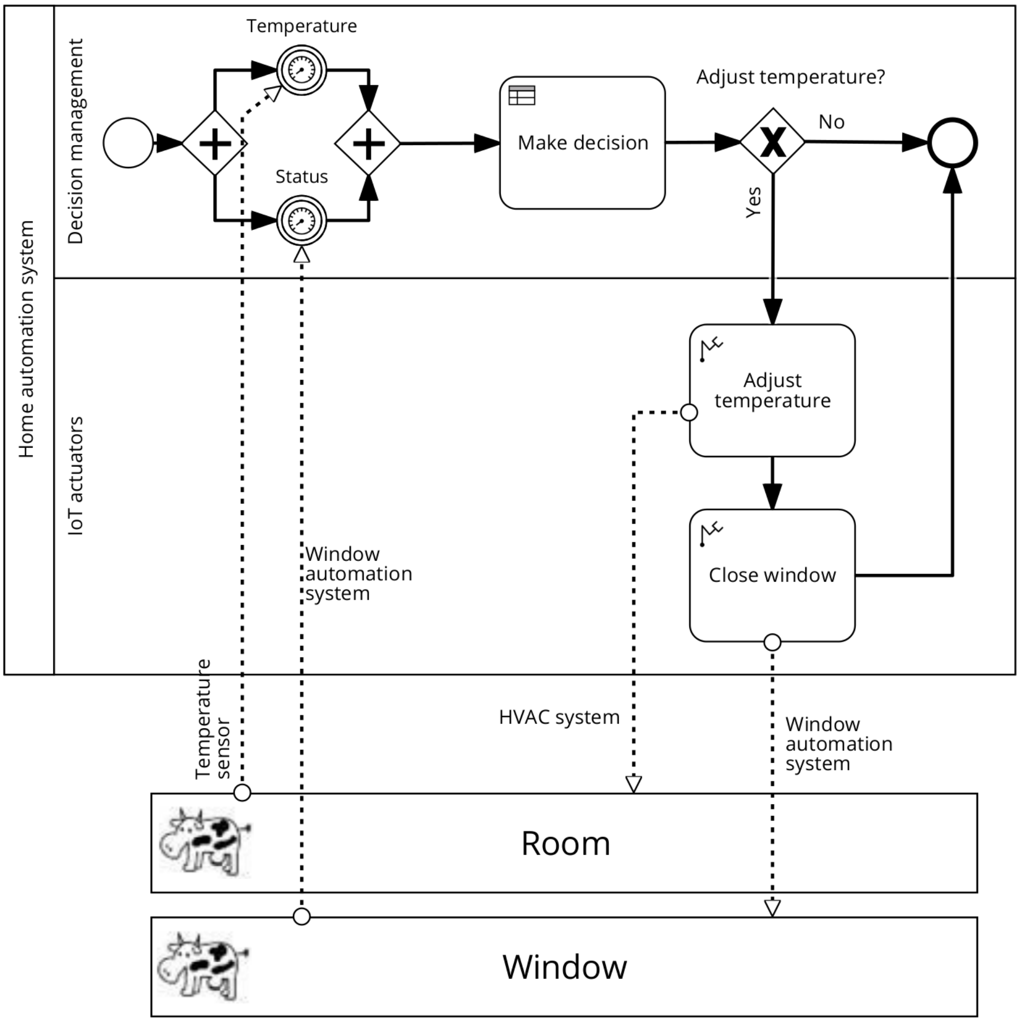

Figure 5.9: Proposed solution for modelling of an event-based *IoT sensing activities* with *IoT sensing intermediate events*.

to the class *EventDefinition*. We define additional attributes *iotServiceRef* and *communicationType*, analogously as for classes *SensingTasks* and *ActuatingTasks* in section 5.1. There is only one variation of event, i.e. *IoT sensing intermediate event*.

We have to extend class *PhysicalAssociation* in order to use it with *SensingEventDefinition*. We define a new subclass *EventbasedSensingAssociation*. Class *EventbasedSensingAssociation* is a directed connection from a *physical entity*, modelled with class *PhysicalEntity*, to an *IoT sensing intermediate event*, modelled with class *SensingEventDefinition*. *IoT device* is modelled with an attribute *iotDeviceRef*.

Those two extensions cannot be defined with BPMN's extension mechanism, as they add new classes to BPMN metamodel.

5.6 Modelling *properties of interest* with item-aware elements

Main idea of this solution is to extend modelling of *physical entity* with information about *properties of interest*.

5.6.1 Proposed modelling approach

In this modelling approach, we propose modelling of *properties of interest* with item-aware elements, attached to a pool representing a *physical entity*. This notation enables user to move information about *properties of interest* from process flow to a graphical stencil representing a *physical entity*. It improves readability of the diagrams and enables reuse of defined *physical entities*, since they now also include information specifying *properties of interest*.

Item-aware elements are used to store or convey items during process execution [26]. In case of real-world processes, they can store *properties of interest* of corresponding *physical entities*, which are mirrored with real-world objects with the use of *IoT devices*. Proposed modelling approach is therefore in line with BPMN semantics.

We define a new item-aware element for modelling of *properties of interest*. Each *physical entity* can be now modelled with all its corresponding *properties of interest*. Each *property of interest* can be marked as available for *IoT sensing activities*, *IoT actuating activities*, or both.

We propose using the same graphical stencils as for data object, with symbols of a gauge, that marks *property of interest* available for *IoT sensing activity*, and symbol of a robot arm, that marks *property of interest* available for *IoT actuating activity*. If *property of interest* is available for both sensing and actuating, graphical stencil includes both symbols.

Modelling of *properties of interest* is now explicit and focused on *physical entity's* point of view. User is now able to model *physical entity* with all its *properties of interest*.

We propose modelling of a request/response type of communication used by *IoT devices* the same way as proposed in solution 5.3. An example of a process with a request/response type of communication and specified *properties of interest* is shown in figure 5.10. Process, shown in figure, starts with a start event and then performs two *IoT sensing tasks* in parallel. *IoT sensing task* **sense temperature** measures *property of interest* **temperature** of *physical entity* **room**. *IoT task* is performed by *IoT device* **temperature sensor**. *IoT sensing task* **sense window status** measures *property of interest* **status** of *physical entity* **window**. *IoT task* is performed by *IoT device* **window automation system**. Process then makes decision based on defined rules. If temperature adjustment is needed, *IoT device* **HVAC system** performs *IoT actuating task* **adjust temperature** on *property of interest* **temperature** of *physical entity* **room**. Next, *IoT device* **window automation system** performs *IoT actuating task* **close window** on *property of interest* **status** of *physical entity* **window**.

For modelling of an event-based type of communication, we propose similar approach than in solution 5.5, Only difference being, that we no longer

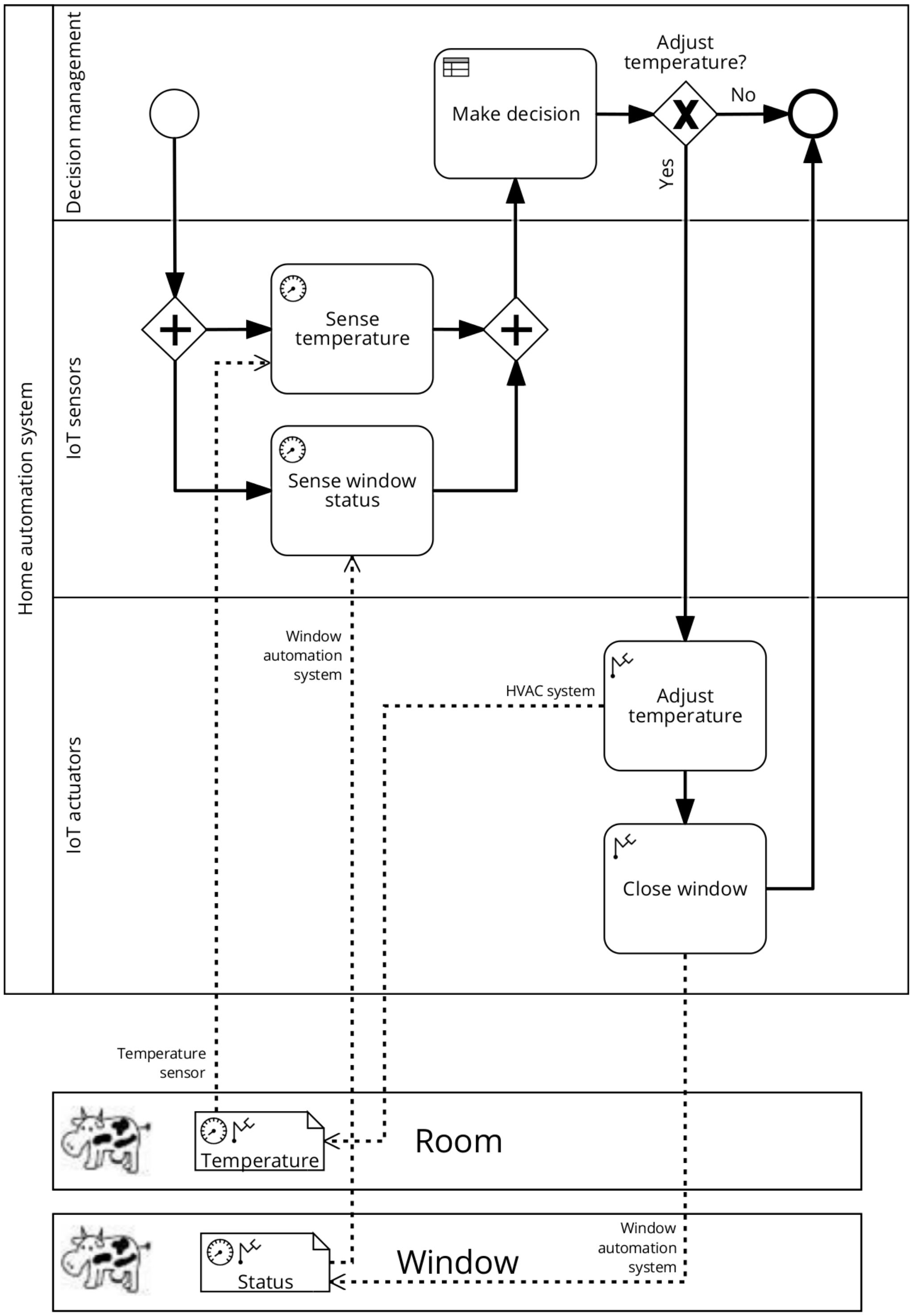

Figure 5.10: Proposed solution for modelling *properties of interest* with item-aware elements and a request/response type of communication used by *IoT devices.*

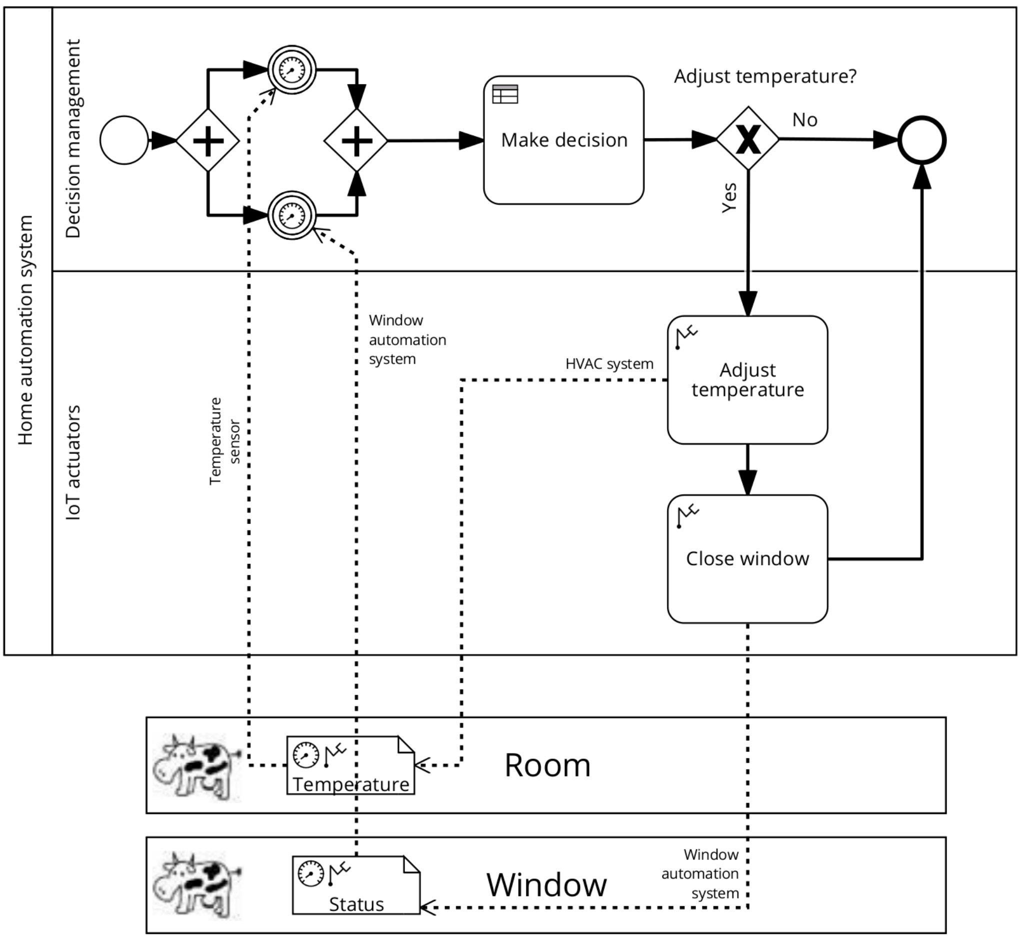

Figure 5.11: Proposed solution for modelling *properties of interest* with item-aware elements and an event-based type of communication used by *IoT devices*.

need to specify *property of interest* for an *IoT sensing intermediate event*. An example of a process with an event-based type of communication and specified *properties of interest* is shown in figure 5.11. Process, shown in figure, starts with a start event and proceeds with two parallel *IoT sensing intermediate events*, waiting for measured *property of interest* **temperature** of *physical entity* **room**, measured by *IoT device* **temperature sensor**, and *property of interest* **status** of *physical entity* **window**, measured by *IoT device* **window automation system**. Process then continues with a decision task. If temperature adjustment is needed, *IoT device* **HVAC system** adjusts *property of interest* **temperature** of *physical entity* **room**. Next, *IoT device* **window automation system** adjusts *property of interest* **status** of *physical entity* **window**. Process ends with an end event.

5.6.2 Proposed extensions to BPMN

In order to model *properties of interest* as item-aware elements, we define new class *PropertyOfInterest* which extends class *ItemAwareElement*. Class has attributes *physicalEntityRef* and *availability*. Attribute *physicalEntityRef* link *property of interest* with a *physical entity*, modelled with a class *PhysicalEntity*. Attribute *availability* can hold values *sensing*, *actuating* or *both* and marks *property of interest* as available for *IoT sensing activities*, *IoT actuating activities*, or both types of *IoT activities*, respectively.

Newly defined class shall be modelled with the same graphical stencil as data object, with symbols of gauge, robot arm, or both, for *properties of interest* available for *IoT sensing activities*, *IoT actuating activities* or both types of *IoT activities*, respectively.

We also have to redefine class *PhysicalAssociation*. It no longer represents the association to *physical entity*, but instead to *property of interest*. We redefine class *PhysicalAssociation*, so that it now represents association with class *PropertyOfInterest* instead with a class *PhysicalEntity*.

Proposed extensions cannot be defined with BPMN's extension mechanism, as they add new class to BPMN metamodel and change the relationships between existing classes, respectively.

In next chapter, we evaluate proposed solutions against defined requirements and identify solution, which solves our research problem in best way possible.

Chapter 6

Evaluation and final solution

This chapter first evaluates all proposed solutions against defined requirements. Secondly, it presents the final solution for solving our research questions.

6.1 Evaluation of proposed solutions

In this section, we evaluate all proposed solutions against defined requirements. Each solution is rated with the number of requirements it satisfies. Solution, which satisfies the highest number of requirements, is proposed as a final solution.

All proposed solutions are evaluated using a sample process, presented in section 3.1.6, representing a hypothetical scenario from a domain of home automation, which we already used as a sample domain in this thesis. Process models adjustment of room temperature based on measured data, with usage of three *IoT devices*. For each proposed solution, we try to model this process, using proposed modelling approaches and extensions. Created diagrams are analysed in order to determine which requirements are satisfied by proposed solution. Results of evaluation are presented in next sections.

6.1.1 Modelling a single ***IoT device*** as a single lane

This section evaluates a solution proposed in section 5.1. Proposed solution provides a modelling approach for modelling of a request/response and an event-based communication types. *Properties of interest* are however modelled only implicitly.

Proposed solution does not satisfy a majority of requirements supporting process modelling. It does not provide a way of representing an isolated *property of interest* and consequently it cannot be annotated for sensing and/or actuating, associated with a *physical entity* nor with *IoT activity.* There are also no predefined *properties of interest.* Diagrams, created by this modelling approach, become crowded and unreadable, when modelling a big number of *IoT devices.* Modelling approach does not improve the modelling of *IoT device* responsible for execution of an *IoT activity.* However, it provides a way to model an event-based communication type for *IoT sensing activities.* Solution satisfies 1 of 9 proposed requirements from this group.

Proposed solution does not support modelling of *properties of interest* and therefore support none of proposed requirements supporting validity of modelled processes.

Proposed solution supports both defined requirements supporting process execution. Reference to *IoT service* and a type of communication used for certain *IoT activity* are stored inside classes *SensingTasks* and *ActuatingTasks*, extended in section 5.1.2.

Proposed solution satisfies all three requirements supporting process of extending BPMN. BPMN specification restricts the usage of events to only those types of events, that affect the sequence or timing of activities of a process [26]. Usage of events, proposed in this modelling approach for modelling of event-based type of communication, is in compliance with BPMN specification, as in this case we use events to model a point in time when sensed data is received by a process engine. Extensions are realised with BPMN's extension mechanism and do not modify existing BPMN metamodel.

Altogether, proposed solution satisfies 6 of 16 proposed requirements.

6.1.2 Modelling all *IoT devices* as a single lane

This section evaluates the solution proposed in section 5.2. Proposed solution provides a modelling approach for modelling of a request/response and an event-based communication types and improves the readability of diagrams with a new approach for modelling of *IoT devices*. *Properties of interest* are however modelled only implicitly.

Proposed solution again does not satisfy a majority of requirements supporting process modelling. It does not provide a way of representing an isolated *property of interest* and therefore does not satisfy any of related requirements. Diagrams, created by this modelling approach, are uncrowded and readable even as the number of modelled *IoT devices* rises, as we no longer have to add a process lane for every *IoT device*. Modelling approach provides a way to model an event-based communication type. Modelling of *IoT device*, responsible for execution of an *IoT activity*, is improved and does no longer result in big and crowded diagrams, even with high number of modelled *IoT devices*. Solution satisfies 3 of 9 proposed requirements from this group.

Proposed solution does not support modelling of *properties of interest* and therefore support none of proposed requirements supporting validity of modelled processes.

Proposed solution supports both proposed requirements supporting process execution. Reference to *IoT service* and a type of communication used for certain *IoT activity* are stored inside classes *SensingTasks* and *ActuatingTasks*, extended in section 5.1.2.

Proposed solution satisfies all three requirements supporting process of extending BPMN. Newly defined extensions respect semantics of existing elements, extensions are realised using BPMN's extension mechanism and no changes to existing metamodel are needed.

Altogether, proposed solution satisfies 8 of 16 proposed requirements.

6.1.3 Modelling sensors and actuators as two lanes

This section evaluates the solution proposed in section 5.3. Solution proposes modelling approach similar to one presented in section 5.2, only difference being a division of *IoT devices* into two lanes, representing sensors and actuators, enabling creation of slightly more readable diagrams. However, sensors and actuators can easily be distinguished by different directions of arrows representing association between *IoT activity* and *physical entity*. Furthermore, using this approach results in slightly bigger diagrams, with no additional advantages. *Properties of interest* are still modelled only implicitly.

Because of the similarity of the two approaches, this approach satisfies the same set of requirements as solution from section 5.2. Similarly as evaluated in 6.1.2, this solution also satisfies 8 of 16 proposed requirements.

6.1.4 Modelling a set of *IoT devices* as a pool

This section evaluates the solution proposed in section 5.4. Proposed solution tries to introduce a modelling approach, which supports modelling of the structure of *IoT devices* as installed in the real world. However, proposed modelling approach results in big and clumsy diagrams with *properties of interest* again being modelled only implicitly.

Proposed solution tries to mirror a structure of *IoT devices* from real world to process diagrams. Proposed modelling approach enables modelling of such structure with the usage of pools, but results in crowded and unreadable diagrams. Therefore, proposed solution does not satisfy the requirement stating, that created diagrams shall be uncrowded and readable.

Proposed solution therefore satisfies 2 out of 9 requirements supporting process modelling; it provides support for modelling of an event-based communication type and improves the modelling of *IoT devices*, using extensions defined as a part of solution presented in section 5.2.

Since it does not provide a support for modelling of *properties of interest*, this solution satisfies none of the requirements supporting validity of modelled processes. It satisfies both requirements supporting process execution by using extensions defined in section 5.1.2. As no new extensions are defined for this solution, it satisfies all three requirements supporting process of extending BPMN. Proposed solution therefore satisfies 7 of 16 proposed requirements.

6.1.5 Modelling event-based sensing with events

This section evaluates the solution proposed in section 5.5. Solution proposes modelling approach for modelling of an event-based *IoT sensing activities* with events. It is a step in right direction, as event-based sensing clearly corresponds with the notion of events in BPMN.

Properties of interest are modelled as a part on *IoT sensing intermediate event*. They are stated explicitly, but from an event point of view and cannot be associated with a *physical entity*. Therefore, this approach again does not satisfy the requirements related to modelling of *properties of interest*. Defined *IoT sensing intermediate event* introduces a clear approach for modelling of an event-based sensing. Created diagrams are clear and readable, even with high number of *IoT devices*, which can be modelled more efficiently as in the state of the art approaches. Proposed solution therefore satisfies 3 of 9 requirements supporting process modelling.

Proposed solution does not satisfy any requirement supporting validity of modelled processes. It supports both requirements supporting process execution with usage of extensions defined in section 5.1.2.

Proposed solution supports all three requirements supporting the process of extending BPMN. Newly defined element respects the semantics of existing BPMN elements, as it represents the point in process, where process is waiting for an *IoT device* to perform an *IoT sensing activity* and to send measured data to a process engine. Solution defines two new classes, which cannot be implemented with extensibility mechanism. Metamodel have to be changed, but the changes are minimal, as we merely add two classes and do not change any relations.

Proposed solution therefore satisfies 8 of 16 proposed requirements.

6.1.6 Modelling *properties of interest* with item-aware elements

This section evaluates the solution proposed in section 5.6. Solution combines proposed modelling approaches for modelling of different types of communication. It proposes modelling of *IoT activities*, using a request/response communication type, with *IoT tasks*, and event-based *IoT sensing activities* with *IoT sensing intermediate events*. Proposed modelling approach satisfies the semantics of existing BPMN elements. Furthermore, it provides modelling approach for explicitly modelling *properties of interest* as parts of *physical entity*.

By providing support for modelling of *properties of interest* of a *physical entity*, this solution satisfies requirements supporting process modelling, which have so far been neglected. User is able to define a *physical entity* with all corresponding *properties of interest*. Each *property of interest* can be associated with one or more *IoT activities* and can also be marked according to its availability for sensing and actuating.

IoT device responsible for performing of an *IoT activity* is specified with an annotation on association between *IoT activity* and *property of interest* and is an improvement in comparison with the state of the art modelling approaches. Proposed solution provides an approach for modelling different types of communication, by combining best approaches from previously described solutions. Resulting diagrams are therefore clear, uncrowded and readable. Solution does not include a set of predefined *properties of interest*, but such set can be defined and included by each implementation of a modelling tool, supporting described modelling approach. Solution satisfies 8 of 9 proposed requirements supporting process modelling.

Proposed solution supports both requirements supporting validity of modelled processes. Each *property of interest* have reference to associated *physical entity* and can therefore be associated with only one *physical entity*. Imple-

Proposed solution	**Satisfied requirements**
Modelling a single IoT device as a single lane	6/16
Modelling all IoT devices as a single lane	8/16
Modelling sensors and actuators as two lanes	8/16
Modelling a set of IoT devices as a pool	7/16
Modelling event-based sensing with events	8/16
Modelling properties of interest with item-aware elements	15/16

Table 6.1: Number of requirements satisfied by each proposed solution.

mentations have to ensure, that each modelled *property of interest* has a reference to a *physical entity* and that each *physical entity* has at least one *property of interest* referencing it.

Solutions satisfies both requirements supporting process execution with extensions defined in section 5.1.2.

All three requirements supporting process of extending BPMN are satisfied. Newly defined elements respect semantics of existing BPMN elements. Extensions cannot be realised using BPMN's extensions mechanism, however, changes to BPMN metamodel are as small as possible, as only one class is added and one existing relation changed.

Altogether, proposed solution satisfies 15 of 16 proposed requirements.

Number of requirements satisfied by each proposed solution is presented in table 6.1.

6.2 Final solution

Based on the evaluation of proposed solutions described in previous section, we present our final solution for modelling of *properties of interest*, an event-based communication and improved modelling of *IoT devices*. Proposed solution, which satisfies the highest number of proposed requirements, is our final solution.

Based on the evaluation, our final solution is solution proposed in section 5.6, satisfying 15 of 16 defined requirements. It provides a solution for our research questions, solving all three identified open topics, which we address in this thesis.

Solution supports modelling of *properties of interest* by introducing a new item-aware element. *Properties of interest* can be attached to *physical entities* and addressed from *IoT activities*. A request/response type of communication is modelled with *IoT tasks*, similarly as in the approaches proposed in the state of the art research. For modelling of an event-based type of communication, we introduced a new IoT-specific event. Modelling of *IoT devices* is improved by extending class *PhysicalAssociation* and corresponding graphical stencil with a reference to an *IoT device*, responsible for execution of an *IoT activity*.

Solution presents an adequate and acceptable answer to proposed research questions. It improves modelling of real-world processes by introducing support for two much needed modelling aspects; *properties of interest* and an event-based type of communication. It also improves the existing modelling approach for modelling of *IoT devices*.

In next chapter we describe conclusions and future work on this topic.

Chapter 7

Conclusions and future work

This chapter describes conclusions of our work and provides a short overview of the future work.

7.1 Conclusions

It this thesis we described IoT, its application areas and open challenges. We presented motivation on why IoT shall become a part of process-driven applications. We argued, that in order to achieve that, business process modelling notations have to be extended in order to support IoT specific characteristics of real-world processes.

After analysing the state of the art research and identifying open topics, we focused our research on three particular problems. First, how to model individual properties of physical entities. Second, improving the modelling of *IoT devices*, and third, how to model an event-based type of communication. Based on those identified problems, we formed our research questions.

In the process of solving identified research questions, we first defined four sets of requirements, which shall be addressed by proposed solutions. We then proposed six solutions for stated research questions. Each proposed solution has been evaluated against defined requirements. Solution, which satisfied the highest number of requirements, was then proposed as our final solution for stated problem.

Our final solution successfully solves stated problem and presents an important step towards process modelling notations, supporting modelling of real-world processes. It addresses all problems stated in our research questions. Solution provides a process modelling approach together with corresponding extensions to BPMN. Future work shall implement those extensions into existing BPMN modelling tools and execution engines, in order to evaluate them on practical use cases.

This thesis presents a small but important part towards process modelling notations that support modelling of real-world processes. However, a lot of research still needs to be done, in order to fully integrate worlds of IoT and process-driven enterprise applications. Some ideas and directives for future work are presented in next section.

7.2 Future work

With research performed in this thesis, we made a small, but significant step towards process modelling notations, which support modelling of real-word processes. However, there is still a lot of work to be done for process modelling notations to fully support modelling of real-world processes. In this section we highlight a few topics, which should be addressed, following the work presented in this thesis.

In order to use the solution developed in this thesis, BPMN modelling tools and process execution engines need to be extended. Process modelling tools shall provide support for modelling approaches and extensions presented in this thesis and execution engines shall provide support for executing them. Future work shall extend one of the open source BPMN suits, extending the modelling tool and the execution engine. Extended tools would help with evaluating our solution on practical use cases from real-world scenarios.

Secondly, other IoT characteristics relevant for modelling of real-world processes, described in 2.3.3, shall be analysed. Future work shall focus on characteristics, which are most needed in order to model efficient real-world processes.

Another interesting research topic is IoT middleware, supporting availability, mobility and fault tolerance of *IoT devices*. Such middleware should act as a mediator between *IoT devices* and backend systems and minimise the impact of unexpected events on the operations of IoT applications.

With IoT being a new technology, there are still many other research opportunities and open topics which need to be addressed, some of them already described in section 2.1.1. A lot of research and investments from industry is still needed, before IoT can become a mainstream technology.

Bibliography

[1] Luigi Atzori, Antonio Iera, and Giacomo Morabito. The internet of things: A survey. *Computer networks*, 54(15):2787–2805, 2010.

[2] Debasis Bandyopadhyay and Jaydip Sen. Internet of things: Applications and challenges in technology and standardization. *Wireless Personal Communications*, 58(1):49–69, 2011.

[3] Joseph Bradley, Joel Barbier, and Doug Handler. Embracing the internet of everything to capture your share of $14.4 trillion. *White Paper, Cisco*, 2013.

[4] Hsiao-Hsien Chiu and Ming-Shi Wang. Extending event elements of business process model for internet of things. In *Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing (CIT/IUCC/DASC/PICOM), 2015 IEEE International Conference on*, pages 783–788. IEEE, 2015.

[5] Li Da Xu, Wu He, and Shancang Li. Internet of things in industries: A survey. *Industrial Informatics, IEEE Transactions on*, 10(4):2233–2243, 2014.

[6] Suparna De, Payam Barnaghi, Martin Bauer, and Stefan Meissner. Service modelling for the internet of things. In *Computer Science and Information Systems (FedCSIS), 2011 Federated Conference on*, pages 949–955. IEEE, 2011.

[7] Suparna De, Tarek Elsaleh, Payam Barnaghi, and Stefan Meissner. An internet of things platform for real-world and digital objects. *Scalable Computing: Practice and Experience*, 13(1):45–58, 2012.

[8] Jayavardhana Gubbi, Rajkumar Buyya, Slaven Marusic, and Marimuthu Palaniswami. Internet of things (iot): A vision, architectural elements, and future directions. *Future Generation Computer Systems*, 29(7):1645–1660, 2013.

[9] Stephan Haller. The things in the internet of things. *Poster at the (IoT 2010). Tokyo, Japan, November*, 5:26, 2010.

[10] Stephan Haller, Stamatis Karnouskos, and Christoph Schroth. *The internet of things in an enterprise context*. Springer, 2008.

[11] Stephan Haller and Carsten Magerkurth. The real-time enterprise: Iot-enabled business processes. In *IETF IAB Workshop on Interconnecting Smart Objects with the Internet*. Citeseer, 2011.

[12] Matjaz B Juric. Design principles for process-driven architectures using oracle bpm and soa suite 12c. 2015.

[13] Matjaz B Juric and Swami Chandrasekaran. *WS-BPEL 2.0 for SOA Composite Applications with IBM WebSphere 7*. Packt Publishing Ltd, 2010.

[14] Matjaz B Juric and Kapil Pant. *Business process driven SOA using BPMN and BPEL: from business process modeling to orchestration and service oriented architecture*. Packt Pub., 2008.

[15] Vasileios Karagiannis, Periklis Chatzimisios, Francisco Vazquez-Gallego, and Jesus Alonso-Zarate. A survey on application layer protocols for the internet of things. *Transaction on IoT and Cloud Computing*, 3(1):11–17, 2015.

[16] Mateja Kocbek, Gregor Jošt, Marjan Heričko, and Gregor Polančič. Business process model and notation: The current state of affairs. *Computer Science and Information Systems*, (00):6–6, 2015.

[17] Tomas Kozel. Bpmn mobilisation. In *Proceedings of the European Conference of Systems: World Scientific and Engineering Academy and Society, WSEAS*, 2010.

[18] In Lee and Kyoochun Lee. The internet of things (iot): Applications, investments, and challenges for enterprises. *Business Horizons*, 58(4):431–440, 2015.

[19] Heather Levy. What's new in gartner's hype cycle for emerging technologies, 2015. `http://www.gartner.com/smarterwithgartner/whats-new-in-gartners-hype-cycle-for-emerging-technologies-2015/`, October 2015. Access: 02.06.2016.

[20] Carsten Magerkurth, Klaus Sperner, Sonja Meyer, and Martin Strohbach. Towards context-aware retail environments: An infrastructure perspective. *Mobile Interaction in Retail Environments (MIRE 2011), Stockholm, Sweden*, 2011.

[21] Sonja Meyer, Andreas Ruppen, and Lorenz Hilty. The things of the internet of things in bpmn. In *Advanced Information Systems Engineering Workshops*, pages 285–297. Springer, 2015.

[22] Sonja Meyer, Andreas Ruppen, and Carsten Magerkurth. Internet of things-aware process modeling: integrating iot devices as business process resources. In *Advanced Information Systems Engineering*, pages 84–98. Springer, 2013.

[23] Sonja Meyer, Klaus Sperner, and Carsten Magerkurth. Towards real world aware enterprise systems-reflecting the quality information of physical resources in services and processes. In *Mobile Adhoc and Sen-*

sor Systems (MASS), 2011 IEEE 8th International Conference on, pages 843–848. IEEE, 2011.

[24] Sonja Meyer, Klaus Sperner, Carsten Magerkurth, and Jacques Pasquier. Towards modeling real-world aware business processes. In *Proceedings of the Second International Workshop on Web of Things*, page 8. ACM, 2011.

[25] Daniele Miorandi, Sabrina Sicari, Francesco De Pellegrini, and Imrich Chlamtac. Internet of things: Vision, applications and research challenges. *Ad Hoc Networks*, 10(7):1497–1516, 2012.

[26] Business Process Model. Notation (bpmn) version 2.0.2. *OMG Specification, Object Management Group*, 2013.

[27] Koosha Paridel, Engineer Bainomugisha, Yves Vanrompay, Yolande Berbers, and Wolfgang De Meuter. Middleware for the internet of things, design goals and challenges. *Electronic Communications of the EASST*, 28, 2010.

[28] Alexandru Serbanati, Carlo Maria Medaglia, and Ugo Biader Ceipidor. *Building blocks of the internet of things: State of the art and beyond.* INTECH Open Access Publisher, 2011.

[29] B. Silver. *BPMN Method and Style: With BPMN Implementer's Guide.* Cody-Cassidy Press, 2011.

[30] Klaus Sperner, Sonja Meyer, and Carsten Magerkurth. Introducing entity-based concepts to business process modeling. In *Business Process Model and Notation*, pages 166–171. Springer, 2011.

[31] Patrik Spiess, Stamatis Karnouskos, Dominique Guinard, Domnic Savio, Oliver Baecker, LMSD Souza, and Vlad Trifa. Soa-based integration of the internet of things in enterprise services. In *Web Services, 2009. ICWS 2009. IEEE International Conference on*, pages 968–975. IEEE, 2009.

[32] Matthias Thoma, Steffen Meyer, Klaus Sperner, Stefan Meissner, and Torsten Braun. On iot-services: Survey, classification and enterprise integration. In *Green Computing and Communications (GreenCom), 2012 IEEE International Conference on*, pages 257–260. IEEE, 2012.

[33] Stefano Tranquillini, Patrik Spieß, Florian Daniel, Stamatis Karnouskos, Fabio Casati, Nina Oertel, Luca Mottola, Felix Jonathan Oppermann, Gian Pietro Picco, Kay Römer, et al. Process-based design and integration of wireless sensor network applications. In *Business Process Management*, pages 134–149. Springer, 2012.

[34] Viveca Woods and Rob van der Meulen. Gartner says iot adoption is driving the use of platform as a service. `http://www.gartner.com/newsroom/id/3241817`, March 2016. Access: 02.06.2016.

List of Figures