



HoloFlows: modelling of processes for the Internet of Things in mixed reality

Ronny Seiger¹ · Romina Kühn² · Mandy Korzetz² · Uwe Aßmann²

Received: 28 November 2019 / Revised: 9 November 2020 / Accepted: 21 December 2020
© The Author(s) 2021

Abstract

Our everyday lives are increasingly pervaded by digital assistants and smart devices forming the Internet of Things (IoT). While user interfaces to directly monitor and control individual IoT devices are becoming more sophisticated and end-user friendly, applications to connect standalone IoT devices and create more complex *IoT processes* for automating and assisting users with repetitive tasks still require a high level of technical expertise and programming knowledge. Related approaches for process modelling in IoT mostly suggest extensions to complex modelling languages, require high levels of abstraction and technical knowledge, and rely on unintuitive tools. We present a novel approach for end-user oriented-*no-code*-IoT process modelling using Mixed Reality (MR) technology: *HoloFlows*. Users are able to explore the IoT environment and model processes among sensors and actuators as first-class citizens by simply “drawing” virtual wires among physical IoT devices. MR technology hereby facilitates the understanding of the physical contexts and relations among the IoT devices and provides a new and more intuitive way of modelling IoT processes. The results of a user study comparing HoloFlows with classical modelling approaches show an increased user experience and decrease in required modelling knowledge and technical expertise to create IoT processes.

Keywords Process modelling · Mixed reality · Internet of Things · IoT processes · End-user development

1 Introduction

Smart devices are increasingly pervading our surroundings in the form of interconnected sensors and actuators for home automation, interactive entertainment systems and mobile personal assistants [1]. These devices are part of the Internet of Things (IoT) and weaved more and more “into the fabric of everyday life until they are indistinguishable from it” as proposed in Mark Weiser’s vision of *Ubiquitous Comput-*

ing [2]. However, the interactions with these smart devices are usually limited to direct monitoring and control of IoT appliances and applications. Despite showing high potential for improved comfort and efficiency, more complex automated routines and processes are only rarely implemented in smart environments [3]. One of the main reasons for the limited adoption is that the creation and configuration of these *IoT processes* [4] quickly start to become complicated and often require advanced technical knowledge and programming skills, which is the point where most end-users start to struggle [5]. To overcome this issue, the concepts of *low-code* or even *no-code* development are increasingly adopted by development tools to also enable end-users and domain experts to create applications requiring only little to no programming skills [6]. The smart home is an excellent example of an IoT environment with end-users as main target group that needs simplified tools for process modelling and configuration to wire and coordinate existing IoT devices, automate basic repetitive tasks and customize simple routines—to *program* the IoT environment.

Different approaches and paradigms were applied to realize this kind of low-code development platforms easing the

Communicated by Jens Gulden and Rainer Schmidt.

✉ Ronny Seiger
ronny.seiger@unisg.ch

Romina Kühn
romina.kuehn@tu-dresden.de

Mandy Korzetz
mandy.korzetz@tu-dresden.de

Uwe Aßmann
uwe.assmann@tu-dresden.de

¹ University of St.Gallen, St.Gallen, Switzerland

² Technische Universität Dresden, Dresden, Germany

development of IoT processes. Despite it being not their main application purpose, classical business processes were proposed and extended to model and execute processes among the physical and virtual entities of an IoT environment [7]. Flow-based modelling tools are more focused on the low-code development and wiring of IoT applications (e.g. Node-RED¹) [8]. However, the underlying modelling concepts and tools require high levels of domain and modelling knowledge as well as an understanding of the abstract concepts used to represent the IoT devices and their relations (e.g. as activities and events in process pools in BPMN 2.0 [9], or as flows among service-based nodes in Node-RED). Both approaches are supported by desktop-based modelling tools featuring classical graphical user interfaces (GUIs) with WIMP interactions [10] and are targeted at IoT experts and process engineers.

In this work, we investigate the application of *Mixed Reality* (MR) [11] as new interaction paradigm to facilitate the modelling and configuration of processes among the typical devices of IoT environments for end-users. The goal is to develop a more intuitive and user-friendly approach for composing processes in smart spaces—namely *smart homes*—compared to process modelling approaches relying on “classical” GUIs. In the main part, we present the *HoloFlows* application that utilizes the physical context of IoT devices and everyday metaphors to provide an intuitive approach for the modelling of processes among the IoT devices in smart spaces. With *HoloFlows*, end-users are able to create these processes by drawing virtual wires between the respective physical sensors and actuators and thereby “program” the IoT environment without requiring programming knowledge. With this prototype for MR-based end-user development (EUD) of processes, we investigate the application of new technologies such as IoT and MR in the business process management (BPM) and software engineering domains and compare them to other process modelling approaches that use classical GUIs on desktop and mobile devices.

The paper follows the process of the design science research methodology (DSRM) introduced in [12]. It is structured as follows: Sect. 2 presents the *problem*, main *objectives* and associated opportunities we address with our work according to the DSRM. It also introduces background information on IoT processes, process modelling and MR. Section 3 discusses related work to identify gaps in current approaches. Following the DSRM, Sect. 4 presents the *design and development* of the *HoloFlows* modelling approach for IoT processes in MR with examples to *demonstrate* the artefacts. Section 5 presents a comprehensive user study as *evaluation* of the *HoloFlows* approach and a discussion. Sec-

tion 6 concludes the paper and shows starting points for future work.

This paper is an extended version of the contribution [13] published as part of the proceedings of the BPMDS 2019 working conference. The original paper has been extended as follows: the background section features more details on desktop-based process modelling and on mixed reality; the related work section includes additional approaches regarding processes in the context of IoT and MR; the modelling section features additional details about the individual artefacts of *HoloFlows*—including the underlying domain-specific language—and process examples; the evaluation section now includes a comprehensive user study based on additional example IoT processes (cf. “Appendix”) and additional discussions.

2 Background and objectives

Moving from the simple control of individual IoT devices to the creation of processes among multiple devices and services in an IoT setting to automate more or less complex routines quickly becomes a challenging task for end-users [14]. Apart from detailed device-specific and technical knowledge to configure the desired functionality and behaviour of an IoT device in a process, programming skills are required to combine the functionalities of different devices and thus to *develop* the actual IoT processes using current solutions [15]. These high requirements imposed on the end-users lead to IoT devices mostly being used and controlled in isolation and thus to a waste of the potential of combining all the available IoT devices in more sophisticated *smart* processes [3]. The goal of this work is to investigate different low-code and no-code approaches with respect to their suitability for EUD of IoT processes and thereby increase adoption of processes in smart spaces. The main focus hereby lies on evaluating *Mixed Reality* as a promising novel paradigm for direct interaction with the physical world (i.e., IoT devices) and thereby providing end-users with more intuitive means for developing IoT processes compared to “classical” GUI-based approaches.

2.1 Internet of Things devices and IoT processes

The Internet of Things (IoT) can be regarded as the “world-wide network of interconnected objects uniquely addressable based on standard communication protocols” [16]. Key components of IoT-enabled smart objects and IoT environments are sensors, actuators and micro-processors to consume data from the physical world, act on the physical world, process data and communicate with other smart objects and computers. With our work, we follow the suggestions of the *IoT Reference Model* [17] and view *Sensors* and *Actuators* as

¹ <https://nodered.org/>

main building blocks of IoT devices and environments. An IoT device may be composed of one or more of these components of varying complexity. Physical entities are represented by the virtual counterparts that are associated with these *IoT services* [17]. Being the main actors of processes in the context of this work:

- *Sensors* measure physical properties and produce discrete or continuous streams of events. We also view abstract event sources producing data as (virtual) sensors. In IoT, data from sensors are often consumed by clients on a publish-subscribe basis using standard Web-based protocols (e.g. MQTT²).
- *Actuators* receive commands and manipulate the physical or virtual world by executing these commands. In IoT, the functionality of actuators is often accessed remotely in a service-oriented manner using Web services based on the REST paradigm and HTTP protocol [18].

Following the notion of a *Process* viewed as “a collection of inter-related events, activities and decision points that involve a number of actors and objects” [19], we regard *IoT processes* as processes where the aforementioned IoT devices are the main actors performing the processes [4]. The following IoT process connecting sensors and actuators as key building blocks of IoT environments serves as one example to illustrate the modelling concepts of the different approaches that we will investigate in the course of this work. More exemplary IoT processes are found in “Appendix”.

IoT Process 1 In the process, data from a light sensor are continuously collected and analysed. Once the light value is over 150 Lux, two lamps are switched on in parallel and the process ends afterwards.

2.2 IoT process modelling on “Classical” GUIs

Process modelling is part of the design phase in a process-aware information system (PAIS) in which abstract models of the processes in the PAIS are created for later instantiation and execution [19]. Different formal and more graphically oriented approaches for describing the low-level behaviour and processes of IoT devices, production machines and robots exist [20], e.g. state charts [21], Petri nets [22] or declarative solutions [23]. However, all of these approaches still require a deep knowledge of the abstract underlying formalisms and are targeted towards domain experts and process engineers with programming expertise. In this work, we will investigate three different representative (IoT) process modelling approaches that are oriented more towards end-users, namely *Business Process-oriented Modelling* and *Flow-*

based Modelling as state-of-the-art approaches on “classical” GUIs and *Process Modelling in Mixed Reality*.

2.2.1 Business process-oriented modelling

Business process modelling aims at creating business process models at the conceptual and organizational level considering customer needs [19]. BPMN 2.0 is a general purpose process modelling language and defacto standard in the BPM field [9,24]. Key modelling elements of BPMN 2.0 comprise *Pools* and *Lanes* to specify participants. Different types of *Activities* are used to specify active parts of work for each participant. Various types of *Events* represent passive elements in a process that reacts to other activities, events and messages. *Messages* are used to describe communication among participants. Different sorts of *Gateways* split and merge the flow based on specified conditions. *Data Objects* are associated with data-aware BPMN elements to specify input and output data. Many of these typical process elements can also be found in IoT-related processes, which is why we investigate business processes as suitable approach for modelling processes in IoT [7,25]. The processes usually have to be refined with various additional parameters regarding data, messages, events and service calls to implement them and make them executable via the specific WfMS [19]. In this work, we use the *Camunda Modeler*³ to model IoT processes on a classical GUI using drag and drop interactions. Camunda provides a well-established BPM suite that implements the BPMN 2.0 standard.

Figure 1 shows the exemplary IoT Process 1 in BPMN 2.0. The IoT devices and systems—the light sensor, the smart home control system and two lamp controllers—are represented as pools. The light sensor repeatedly sends light events via a *Send Task* to the control system. Here these events are received via a *Receive Task*, and using an *Exclusive Gateway* the system either waits for new events from the sensor (light value ≥ 150) or issues two calls to the two lamp controllers via two *Service Tasks* (light value < 150). *Parallel Gateways* are used to split and join the flow here. The processes on the two lamp controller are started based on a *Message Start Event* as a result of the two service tasks. Both controllers use a *Task* to activate the individual light switch. Despite it not being its main purpose, the example shows that already standard BPMN 2.0 is suitable to model IoT processes on a conceptual level to a certain degree.

However, BPMN 2.0 is a very rich general purpose language targeted at business process and IT experts [26]. The example already shows that advanced knowledge is required to identify suitable process modelling elements to express the desired process behaviour and relations within a process. Process modellers have to find concepts within the

² <http://mqtt.org/>

³ <https://camunda.com/>

BPMN 2.0 language to abstract the IoT devices, their functionality and produced/consumed data as well as decisions and communication with other devices as virtual entities on a process level. This task is prone to quickly become overwhelming for non-expert modellers (e.g. in a smart home context) [27]. Nevertheless, a significant number of related work propose IoT-related extensions to BPMN 2.0 for modelling IoT-related aspects (cf. Sect. 3.1) [7,25], which is why we also investigate the suitability of BPMN 2.0-based approaches for end-user oriented IoT process modelling.

2.2.2 Flow-based modelling

With flow-based modelling/programming [8], we address approaches that are tailored towards the IoT and focus on the modelling of data flow among software components on a more technical level within an IoT process. We use IBM's *Node-RED* as one of the main subjects of our investigation as it is the most widespread state-of-the-art tool for flow-based modelling/programming [28]. Essential elements of the underlying domain-specific language are *Nodes* representing specific software modules and functionalities associated with concrete IoT devices and services that consume—*Sources*—and produce—*Sinks*—discrete and continuous streams of data. These nodes are flexibly wired together creating *Flows* among the nodes in self-contained processes [29]. The nodes and flows contain specific technical implementation-relevant parameters, which make them directly executable by the corresponding runtime system. Nodes also cover network functionality, splits and joins, and the parsing and persisting of data. They can be extended with custom nodes and scripts. The modelling tool provides a browser-based GUI that uses drag and drop as main form of interaction [30].

Figure 2 shows the exemplary IoT Process 1 in Node-RED. An *MQTT In* node is used and parameterized to subscribe to the data produced by the light sensor via the MQTT protocol. This requires an active MQTT broker to be available and configured as part of the node as well as the specification of the topic related to the light values. The data are directly wired to a *Switch Node* to decide whether the payload is below 150Lux and therefore to activate the two connected *HTTP Request* nodes or to wait for new sensor data. The HTTP request nodes controlling two IoT services that switch on two individual lamps have to be parameterized with the respective URLs, HTTP method and payload and are invoked in parallel. A *Link Out* node is used to indicate the end of the process after the HTTP requests.

While flow-based modelling/programming with Node-RED appears to be more intuitive as it provides means for directly connecting the flow of data among individual components [29], the modelling and configuration of the nodes serving as virtual representations of the IoT devices' functionalities remain on a very technical level and abstract. Users

have to know and configure the individual software components (e.g. MQTT clients, HTTP requests) that are used for accessing the device functionality and data, which also tend to be quickly overwhelming for non-expert users [31]. Still we will investigate its suitability for end-user oriented IoT process modelling as it is specifically tailored to IoT and the most wide-spread, related approach that is closest to our goal [28].

Apart from finding the suitable representations for IoT devices and data among the elements of the process modelling language, the GUI/WIMP-based modelling tools associated with business process-oriented and flow-based process modelling require end-users to perform an additional step of abstraction: the processes among the available IoT devices have to be composed on a virtual canvas. In IoT environments with a large amount of IoT device instances—sensors and actuators—the correlation of the virtual process elements with the concrete physical devices may quickly become very complex and confusing. Moreover, information about the IoT device's physical context (*Reality*) and its relation to other physical objects may be lost [32].

2.3 Process modelling in mixed reality

The main focus of this work is the investigation of *Mixed Reality* as a promising and novel interaction paradigm with lots of potential for many different domains [33]. We investigate its suitability to provide a more natural and intuitive way of modelling processes in IoT [10] compared to GUI-based approaches. According to Milgram “Mixed Reality (MR) visual displays, a particular subset of Virtual Reality (VR)-related technologies, involve the merging of real and virtual worlds somewhere along the ‘virtuality continuum’ which connects completely real environments to completely virtual ones” [11]. Being part of this virtuality continuum, Augmented Reality (AR) systems “supplement the real world with virtual (computer-generated) objects that appear to coexist in the same space as the real world” [34]. While we mostly focus on AR to project extended information about IoT devices and processes as overlay on the physical devices and environment, we will also integrate purely virtual information and holograms without a direct relation to the physical world, which makes the presented approach more generally related to MR.

The Microsoft[®] HoloLens[™] (1st generation) smart glasses are used as an exemplary MR platform in our work [35]. These head-mounted smart glasses use multiple cameras to create a spatial 3D map of their surroundings. Thereby, it is possible to display holographic images in the see-through displays of the glasses and also fix these holograms at specific locations in the virtual scene. The user interacts with the holograms by controlling a virtual mouse pointer via head and eye movements and performing air tap gestures (pinching of

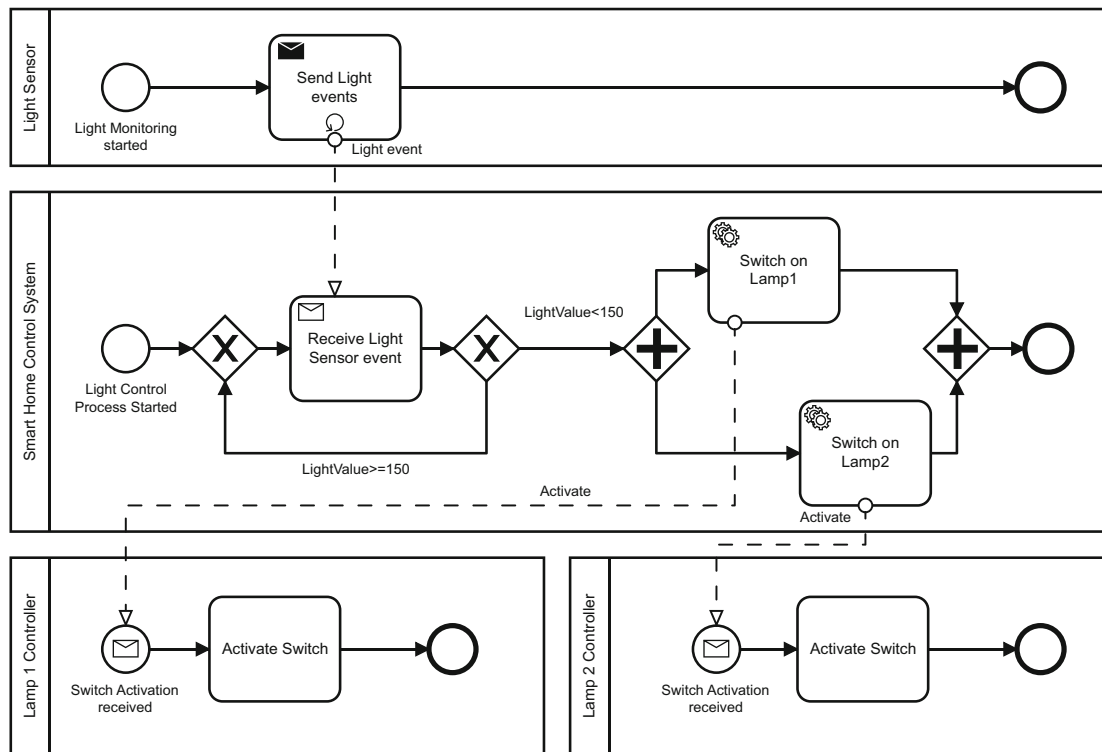
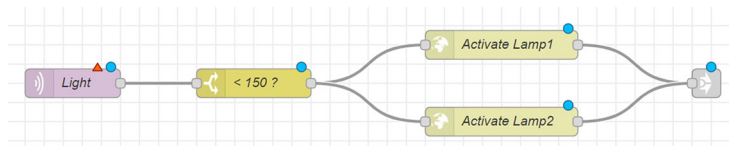


Fig. 1 Example IoT process 1 modelled in BPMN 2.0

Fig. 2 Example IoT process 1 specified in node-RED



thumb and index finger) to “click” the mouse pointer. We will present our mixed reality-based modelling approach for IoT processes—*HoloFlows*—in detail in Sect. 4. The HoloLens thereby merely serves as a demonstration platform for the developed MR-based concepts that can also be transferred to other devices capable of MR interactions and visualizations.

2.4 Objectives

The main goal of this work is to enable end-users to configure and combine the functionality of IoT devices to form IoT processes in an intuitive way without requiring a deep understanding of programming and process modelling.

End-users should be able to easily connect and combine sensors and actuators to either directly control the actuators based on the sensor values or set specific sensor related conditions to activate the actuators. We view these types of connections as the main IoT process (workflow) patterns to be supported. The complexity of the IoT processes should thereby comprise only a few sensors and actuators as end-users tend to specify rather simple processes [14] (*Objec-*

tive O1: Simple IoT process modelling language). However, these IoT processes will already be beneficial for end-users to increase the interplay and automation among the available IoT devices, which will lead to higher levels of efficiency and comfort in the smart home [1,3].

When creating these kind of processes in an IoT environment, the physical relations and contexts of IoT devices should be preserved and help end-users with understanding the components and “hidden” processes of the IoT environment (*Objective O2: Visualizing physical relations*), which will facilitate the adoption of new IoT technologies by the users [36]. MR hereby plays an essential role as enabling technology [11].

End-user acceptance and learning of new concepts/applications also increase with the ability of experimenting in concrete scenarios and direct feedback [5]. Therefore, users should be able to easily create, manipulate and adapt existing IoT processes and directly execute them to experience their immediate effects within the IoT environment (*Objective O3: Experimentation and direct feedback*).

Classical desktop-based process modelling tools following the WIMP paradigm tend to be overloaded with abstract icons, menus, options and frames or windows imposing high cognitive efforts on process modellers, especially on non-experts [10,32]. We aim at designing an intuitive and simple user interface for IoT processing modelling, which reduces information and options to what is required for the current task and uses everyday metaphors and more natural interactions to increase usability (*Objective O4: End-user friendly interface*).

3 Related work

The application of business process management (BPM) technologies in IoT and cyber-physical systems (CPS)—and vice versa—has been vibrantly discussed over the recent years. Various works identified new challenges that emerge with the tighter coupling of the physical and virtual worlds in BPM [3,37–40]. With our work, we address the simplified modelling of processes in IoT for end-users (*end-user development, EUD* [41]), which includes the concretion of abstract process models, the breaking down of end-to-end processes, and the bridging of the gap between event-based and process-based systems from the BPM-IoT Manifesto [3,38]. The focus domain for our work is thereby the *smart home*. A plethora of related works exist that address the modelling and execution of IoT processes relying on conventional applications and GUIs. The application of MR technology to provide an end-user friendly way of modelling IoT processes as a combination of transformative technologies for BPM and software engineering is a novel approach.

3.1 IoT process modelling and execution on “Classical” GUIs

Literature surveys regarding the modelling of IoT-aware business processes can be found in [7,25,42]. Many approaches propose extensions of business process notations with new elements for IoT-/CPS-related sensor and actuator tasks and entities. Sensor-related tasks and conditions are introduced to the WS-BPEL language in [43]. The majority of work discusses extensions of BPMN 2.0 to support new IoT- and CPS-related features. Business process tasks related to sensor networks are addressed by the *BPMN4WSN* extension [44]. Meyer et al. discuss the integration of IoT devices and things as process resources in the form of dedicated lanes [45,46]. Specific new process tasks for sensing and actuating in an IoT/CPS context as well as dedicated pools and lanes are discussed in various works [47–49]. Complementary work discusses ways of integrating IoT devices as process resources (e.g. by Friedow et al. [50] and Suri et al. [49]). While most of these approaches propose the

integration of IoT-related aspects into business processes independent of a specific domain, various works address for example the industrial IoT/high-tech manufacturing domains (*Industry 4.0*) specifically and propose (business) process modelling, implementation and integration concepts with smart factories, production lines, their machines, robots and supply chains [51–54].

All these approaches propose new formalisms and extensions to integrate IoT-related tasks, events, activities and devices as resources into business processes. They extend existing process notations and modelling tools accordingly. These approaches require domain experts and process engineers with a deep knowledge about the underlying formalisms to model the processes (cf. Objectives O1, O4), which then have to be deployed and executed on a special BPM system not providing immediate feedback and means for experimentation (cf. Objective O3). Within the process modelling languages, pools and lanes as well as special tasks are used for representing the actions of IoT devices, which requires a high level of mental abstraction from modelling the IoT process based on a formal representation with a desktop tool to correlating it with the actual execution in the physical world (cf. Objectives O1, O2).

In contrast with business process-oriented modelling, approaches towards a more end-user oriented flow-based programming of IoT environments emerge, e.g. as proposed in [55] where requirements for installations of industrial IoT appliances can be modelled with the help of Node-RED; or with the *CharIoT* end-user programming environment that enables the specification of high-level events from sensor data in combination with actions in automation rules [56] similar to the IFTTT tool⁴. Still these IoT process frameworks require a high degree of technical knowledge and an abstract understanding of the IoT devices and their relations (cf. Objectives O1, O2, O4). We propose to use MR to decrease this complexity and simplify the creation of IoT processes by taking the physical contexts of the devices into account and directly present offered functionality and states at the IoT devices’ physical locations.

An overview of process execution systems for IoT processes and discussion of challenges can be found in [4]. Various works propose architectures and frameworks for executing (business) processes that include IoT-related aspects [57–59], e.g. tasks related to sensor streaming and processing [60] as well as actuator execution and automatic planning of adaptations [61,62]. The integration of specific event streaming mechanisms for sensors into processes is proposed by Appel et al. in [63]. A more advanced architecture relying on digital twins and service orchestrations to execute processes in a smart factory is proposed in [64]. Peng et al. discuss a process execution approach using a Cloud-based process

⁴ <https://ifttt.com/>

orchestrator and mobile clients [65]. For execution of the modelled IoT processes, we rely on a specific system for adaptive processes in IoT that was designed with a focus on the smart home domain [66].

3.2 Mixed reality in process-based IoT applications

MR technology is currently mostly used in process contexts for guidance and advisory purposes, e.g. in manufacturing [67], assembly [68], maintenance [69] and medical applications [70]. General discussions of current and future applications as well as challenges of AR, MR and XR (eXtended Reality) can be found in [71] and in [72]. An approach for the automatic device recognition and process-based configuration in smart environments with the help of AR is proposed in [73]. In [74] the authors present a prototype of an augmented reality collaborative modelling tool for business processes based on BPMN 2.0. Additional works on collaborative business process modelling show the application to new forms of interaction devices (e.g. tabletops [75] or display walls [76]) and also in virtual reality [77]. The system presented by Pryss et al. in [78] is accompanied by 3D augmented reality application to configure and visualize assistance workflows without exploiting the physical contexts of participants. In [79], the authors illustrate an approach for setting up new IoT devices based on formal workflow descriptions in MR. This work can be seen as a preceding step to be integrated into the HoloFlows approach. The configuration and deployment of IoT devices in a smart home with the help of a ubiquitous interaction device for scanning and identifying appliances are presented in [80]. After successful deployment, the device is augmented with additional information and services.

Interactive business process management in combination with mobile devices, IoT and Mixed Reality has so far only been discussed by few works. Approaches for mobile business process management and guidance in the context of IoT are proposed in [78,81–83]. In [84,85], the authors propose BPMN 2.0 extensions for interactive processes and new interaction devices as event sources for processes, as well as patterns for new interaction techniques (e.g. augmented reality) based on these devices. The modelling of industrial systems and inference of software configurations and processes as recommendations in AR are presented in [86].

In general, we see approaches applying MR technology to improve understanding and collaboration by visualizing important data in location-related contexts. The physical world contexts play a key role when designing, modelling and operating IoT environments as they enable end-users to relate virtual data and information about IoT devices to the physical counterparts and contexts (cf. Objective O2). The direct augmentation of IoT devices with digital data (e.g. sensor states, quality of service levels, etc.) and control is proposed in [87]

in a static setting and in [88] with dynamic identification and tracking of the devices. The works discussing the application of AR to BPM for process modelling [74,78] still rely on abstract workflow/process notations and do not make use of the location contexts of process participants and devices. In contrast with these approaches, we will use MR technology to directly compose processes among the available IoT device instances at their physical locations without requiring an understanding of abstract process modelling concepts, thus facilitating end-user development. Especially in smart home environments, end-users—not being domain or modelling experts—have to be provided with an intuitive and easy way to explore and manage their IoT infrastructure as well as the devices' interactions and processes [2] (cf. Objectives O1, O2, O3, O4).

4 HoloFlows: mixed reality-based IoT process modelling

In this section, we present *HoloFlows*—a new IoT process modelling approach that goes beyond classical desktop-based modelling by leveraging MR technology to support end-users with the modelling tasks. First, we introduce the meta-model of the underlying IoT process modelling language. Second, the associated MR user interface is presented. We then demonstrate both artefacts via example IoT processes created following the HoloFlows approach. In the last part, we elaborate on the operation of and process execution in the HoloFlows application [89].

4.1 HoloFlows IoT process modelling language

As the goal of our work is to develop an approach for end-users to model and program their IoT environment with the help of IoT processes, the first objective is to design a simple domain-specific process modelling language (DSL) with low complexity and easy-to-understand concepts (*Objective O1*). The discussions in Sect. 2.2 revealed that business process-oriented and flow-based modelling approaches rely on abstract concepts (e.g. pools/swimlanes or special service-based tasks/nodes) for representing data and functionality of IoT devices, which are not suitable for non-experts. Following the suggestions in [5,31], we regard the IoT devices as first-class citizens within the HoloFlows approach so that end-users can directly relate to the individual devices and their components. Figure 3 shows the HoloFlows meta-model in detail.

Similar to the IoT reference model [17], an *IoT Device* is a composition of one or multiple *IoT Components*. IoT components have a name, an identifier and an icon representing the device in the user interface. *Sensors* and *Actuators* are the atomic units of composite IoT devices.

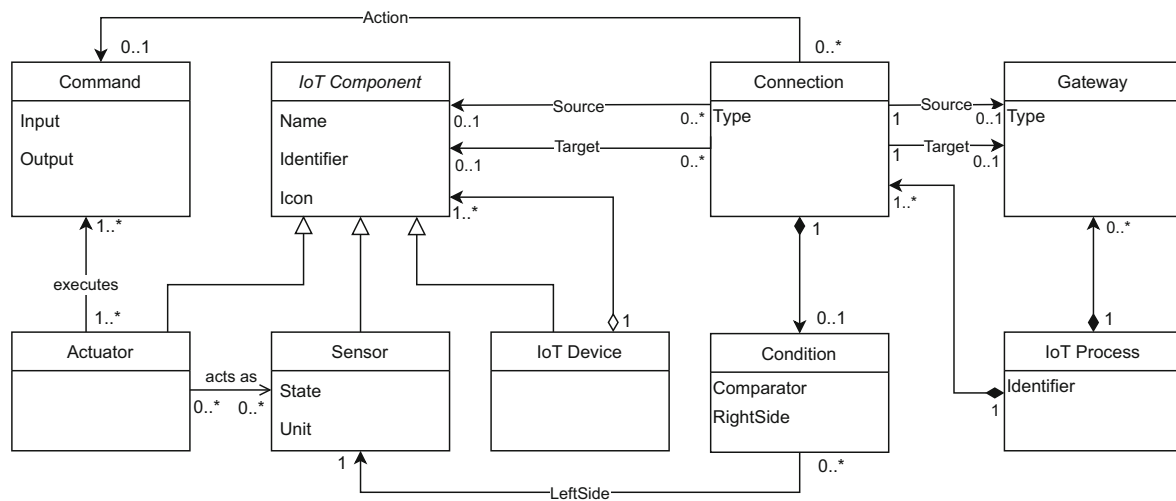


Fig. 3 Meta-model of the IoT process modelling language in HoloFlows

Sensors have a state and an associated unit of measure. Actuators are able to execute one or more Commands that require input and output parameters. An actuator can produce output data (e.g. have a state) and therefore may also act as one or more sensors.

The representation of an IoT Process also follows a minimal approach [90]. An IoT process can be composed of Gateways and has to have at least one Connection. A connection has one source and one target, which can both be either an IoT device or a gateway. A gateway is of a specific type (i.e., OR, AND). A connection may have an associated activation Condition. This condition refers to the state of a sensor (left side), a comparison operator, and a specified threshold (right side) defining the activation of the connection. Following the ECA (Event-Condition-Action) paradigm [91], a connection can also have an actuator command specified as action to be executed as result of its activation. Depending on the nature of source and target and existence of a condition, we distinguish between different types of connections (i.e., direct sensor–actuator connection, conditional sensor–actuator connection, actuator–actuator connection, connection with a gateway). With this simplified meta-model, processes of varying complexity based on single connections—possibly with an activation condition—between two instances of IoT devices or a device and a gateway can be developed. The expressiveness should thereby be sufficient for most modelling tasks performed by end-users in IoT environments [14].

4.2 HoloFlows mixed reality user interface

In line with the IoT process meta-model, the user interface for end-user enabled modelling of IoT processes is the second important artefact of the HoloFlows approach. The goal is to

simplify the modelling of basic IoT processes by using MR technology to provide a better understanding of the physical contexts, relations and effects of the individual devices and their actions. With our target group being end-users in the respective IoT environments, we rely on the following user interface design principles (*Objectives O2, O3, O4*) [92].

4.2.1 Design principles

Ubiquitous exploration and experimentation

HoloFlows allows users to explore their surroundings: all available instances of IoT devices are augmented with extra information including the devices' states and functionality. The information is shown as holograms at the physical location of the device, which also enables the discovery of hidden physical devices embedded into the probably unknown smart space (*Ubiquitous Systems* [2]). Existing IoT processes can be explored in a similar way. Thereby, the end-user is able to get an understanding of the IoT surroundings including available devices, their physical context and spatial relations with other devices as well as IoT processes in the room [87]. Direct control of IoT devices and processes, and modifications to the processes for experimentation are possible, too. The smart glasses facilitate this exploration in a mobile and hands-free manner with easy-to-perform and embodied natural interactions [32] to directly interact with the IoT devices and processes.

Reduced information

When presenting information regarding IoT devices and processes to the user in MR, we reduce the amount of details to only relevant information necessary to understand the properties and states of the devices and the configurations of the existing processes [93]. For process modelling, we perform automatic pre-checking of the compatibility of IoT

devices based on the device selected as source of a connection to then only show compatible targets and options to create the specific connection, which reduces the amount of information and possible configurations.

Everyday metaphors and feedback

To reach a high level of usability and create a steep learning curve, we apply various metaphors [94] from everyday life for natural interactions in the HoloFlows UI [32,95]. Examples include virtual buttons displayed at the respective IoT devices that can be “pushed” in the MR scene to trigger the execution of the respective functionality; and the “drawing” of virtual wires between the devices that should be connected as part of a process. Arrows indicate the direction of control or data flow from source to target. In addition, we provide visual feedback by highlighting actively focused and selected UI elements to facilitate navigation and control in the MR scene [93].

Correlation and creation

With the design of the HoloFlows UI, we put focus on having an easy-to-learn tool for exploring and controlling the IoT environment. MR helps end-users to directly manipulate and correlate the virtual IoT devices with the physical devices and to understand their properties [87]. When controlling the devices and creating processes, the users do not have to rely on complex modelling tools that require an abstract understanding of the physical relations of the devices (e.g. represented as swimlanes [45] or nodes). In HoloFlows, they can simply connect and correlate the desired devices with each other via virtual connections between the physical devices.

4.2.2 IoT process visualization and interactions

Exploration

Figure 4 shows a schematic excerpt of an MR scene during exploration and the corresponding meta-model elements (cf. Sect. 4.1). Sensors (here: a barometer and a humidity sensor) and actuators (here: a door) are augmented with their name, a representative icon and their current (live) states. Actuators feature virtual buttons representing available commands, which can be directly triggered. Devices can be standalone or part of a composite IoT device with branches for each sensor and actuator. Each sensor and actuator hologram has a *Connector Box* enabling the specification of a connection between it as a source or target device. Moreover, each IoT device features an *Anchor Box*, which allows the manual placement of the holographic element in the MR scene. Figure 4 also depicts an existing connection between the barometer sensor and door actuator with an arrow indicating the direction from source to target. A button for process control—start, stop, delete, modify—is positioned over the middle of a connection line. On the connection’s sensor side

the current triggering condition is displayed; on the actuator side the action to be executed is shown.

A connection between IoT devices is created by activating the source’s connector box via an air tap first and then activating the target’s connector box. Compatibility of devices is checked automatically in the HoloFlows app; the set of targets is reduced to only compatible ones after selection of the source. Depending on type of source and target we distinguish between different types of connections.

Creating conditional sensor–actuator connections

Figure 5 shows a schematic excerpt of an MR scene during the creation of a conditional sensor–actuator connection (i.e., an ECA rule). Upon connecting a sensor with an actuator, the user first configures the sensor-related condition—the current value of the sensor as left side, the comparator, and the threshold as right side—and then selects the actuator’s action to be executed when the specified condition becomes true.

Creating direct sensor–actuator connections

The output values of some sensors can serve as direct input to control the states of actuators in a continuous control loop of an IoT process. Upon selecting a compatible sensor first, the user has to select if it should be used to create a conditional sensor–actuator connection or a *direct* sensor–actuator connection. In the latter case, the user then connects the sensor directly to a compatible actuator without the need for additional configuration. Figure 6 shows an example with the colour detected by a colour sensor being directly set as new state of the lamp actuator. The data produced by the sensor have to be compatible with or at least convertible to input data for the respective actuator commands for these types of connections.

Creating actuator–actuator connections

The connection between two actuators to invoke two actions in a row is the third type of supported connection in HoloFlows as illustrated in Fig. 7. Actuators may often serve as sensors and actuators at the same time due to having a state as output data. To create a sequence of actuator actions to be executed, the user has to select the first actuator (here: the door) and choose between it acting as an actuator or as a sensor in case the device offers both functionalities. Actuators can only be connected to other actuators or gateways as targets. Following, the user selects the second actuator to connect it with (here: the lamp). The action to be executed by the first actuator is then selected from a list of available operations (here: open), followed by the same step for the second actuator (here: activate).

Creating connections with gateways

When creating connections, the user is presented with a menu of additional logic elements as gateways to split or join the flow of activations in a process—currently OR and AND gateways. A gateway can serve as source or target of a connection. The user selects and places the respective OR and

Fig. 4 Overview of the HoloFlows UI for exploration

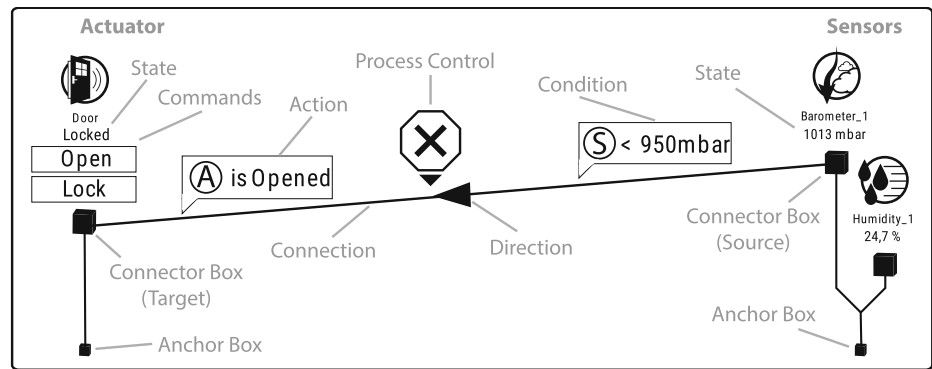


Fig. 5 Overview of the HoloFlows UI for creating a conditional sensor–actuator connection in a process

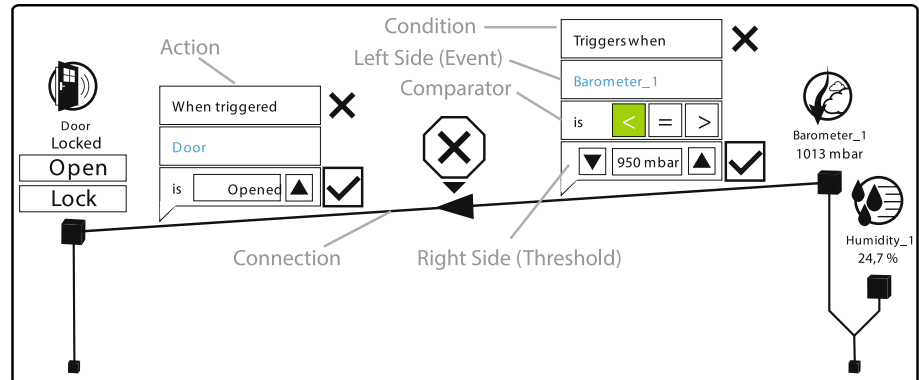
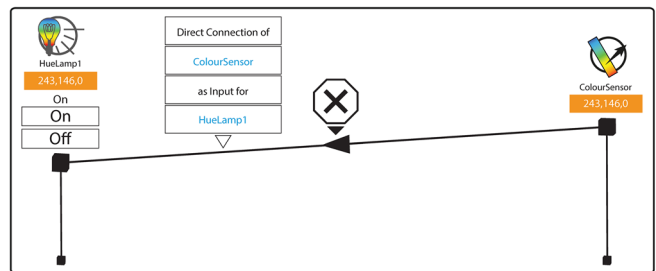


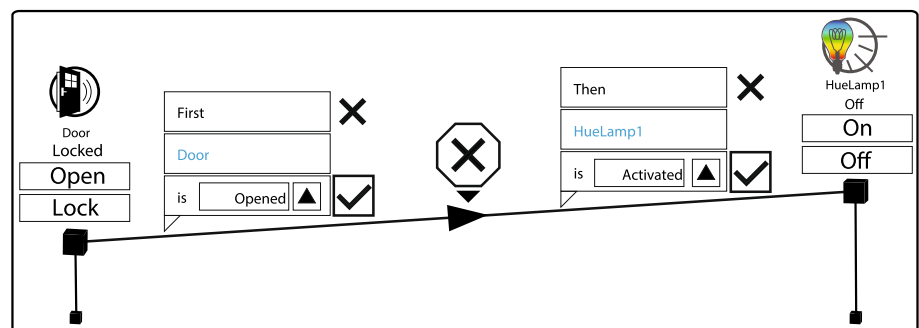
Fig. 6 Creating a direct sensor–actuator process



AND cube holograms freely within the holographic scene. Connections from and to a sensor or an actuator can be created by either first selecting the gateway and then the sensor/actuator or by first selecting the sensor/actuator and then the gateway. This way processes with multiple sensor-related conditions joined via an AND or OR gateway and parallel actuator calls split via an AND gateway can be defined.

These types of connections and modelling elements form the basic building blocks of IoT processes in the context of this work. More complex IoT processes can be composed by creating longer chains of connections among IoT devices and gateways following the presented modelling approach and interactions.

Fig. 7 Creating an actuator–actuator process



4.3 IoT process examples in mixed reality

This section demonstrates the developed artefacts—the IoT process modelling language and the MR user interface—of the HoloFlows approach based on example processes in a smart home setting:

- Figure 8 shows a live view from the HoloFlows app with two simple conditional sensor–actuator connections being modelled: one defining that if the light levels are below 150 Lux, then the coffee maker should start, and one defining that if the temperature is above 26°C, then the power level of the fan should be switched to 100%. Other sensors integrated into the HoloFlows app comprise an NFC tag reader (e.g. to create an authentication process to open the door when a certain NFC tag is present) and a wearable hand movement sensor (e.g. to switch on the light via a simple swipe gesture).
- Figure 9 shows an exemplary direct sensor–actuator connection between a potentiometer as sensor (values from 0 to 100 %) to the power level of a dimmer switch controlling a fan (or lamp) (values from 0 to 100 %).
- Figure 10 shows the live view of an IoT process with a conditional sensor–actuator connection and an AND split defining that if the light levels are below 100 Lux, then the power levels of Lamp 1 should be increased to 50 % AND of Lamp 2 to 79 % in parallel.
- Figure 11 shows an IoT process with two conditional sensor–actuator connections and an OR join defining that the door should be opened if the humidity exceeds 65.8 % OR the temperature is above 28.4°C.

4.4 Execution of IoT processes

With *Objective O3*, we highlighted the importance of experimentation and immediate feedback for end-user development [5], i. e., the modelled IoT processes should be directly executable such that users can study their effects and adjust/tune the processes. The implemented HoloFlows app allows for on-device process execution and server-based process execution.

Figure 12 depicts the system architecture of the IoT backend system and interactions with the HoloFlows mixed reality app⁵ developed for the Microsoft HoloLens (1st generation). We rely on the *OpenHAB*⁶ middleware to connect and unify the heterogeneous set of sensors and actuators of the IoT environment. A semantic model in the knowledge base describes the properties, functionalities and relations of the individual IoT devices [96] and the associated IoT

services [45]. The IoT middleware provides service-based RESTful interfaces to all devices to retrieve data and directly send commands from the HoloFlows app.

4.4.1 On-device execution in the HoloFlows app

The HoloFlows app provides the possibility to execute instances of modelled IoT processes directly on the respective device (i. e., HoloLens). The operation of the HoloFlows app thereby relies on a state machine consisting of the following operational modes, which depend on the current user tasks and interactions.

Exploration mode

The application's default mode is the *Exploration Mode* where the user is able to explore and directly control all IoT devices IoT processes in the vicinity. Holograms above the respective devices show general information, their current states and control functionality that can be triggered via an air tap (cf. Fig. 4). The list of available devices is automatically retrieved from the IoT middleware, and the holograms are shown based on the positions set in the manual placement mode. The user is always presented with a small unobtrusive menu to switch to the *Manual Placement Mode* or *Process Modelling Mode*.

Manual placement mode

In the *Manual Placement Mode*, the user is able to place holograms for the individual IoT devices at their physical locations in the MR scene. Upon selecting the hologram's anchor box (cf. Fig. 4), the hologram can be moved to the desired location and its position fixed with another air tap on the anchor box. The hologram will then stay at this exact position. We currently rely on a manual placement of these holograms by the user, i. e., the hologram positions have to be adjusted in case an IoT device is moved.

Process modelling mode

The *Process Modelling Mode* allows the user to create variations and chains of different types of connections and processes described in Sect. 4.2. A menu is displayed once this mode becomes active from which the user can select additional processes elements (gateways), save the current state of the process model, delete the connections and exit the process modelling mode.

Process mode

The *Process Mode* enables the user to control individual processes, which have been created in the process modelling mode. Upon clicking on the respective control panel of a process (cf. Fig. 4), an expanded control menu for the process is presented to the user. Here the user is able to edit the process configuration (enter the process modelling mode), delete it, directly execute it by entering the process execution mode or deploy it to an external WfMS.

Process execution mode

⁵ <https://github.com/IoTUDresden/HoloFlows>

⁶ <https://www.openhab.org/>

Fig. 8 Live view of two conditional sensor–actuator connections and an air tap

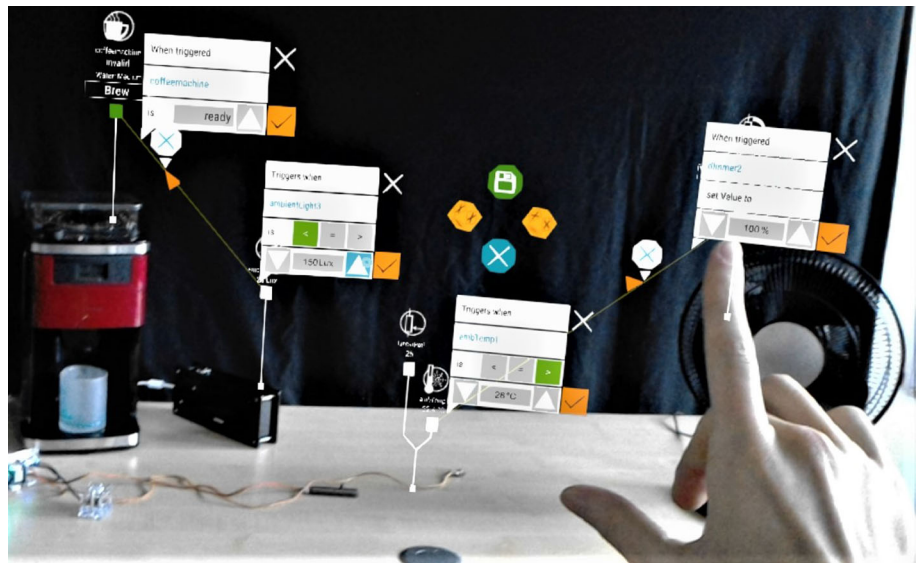


Fig. 9 Live view of a direct sensor–actuator connection

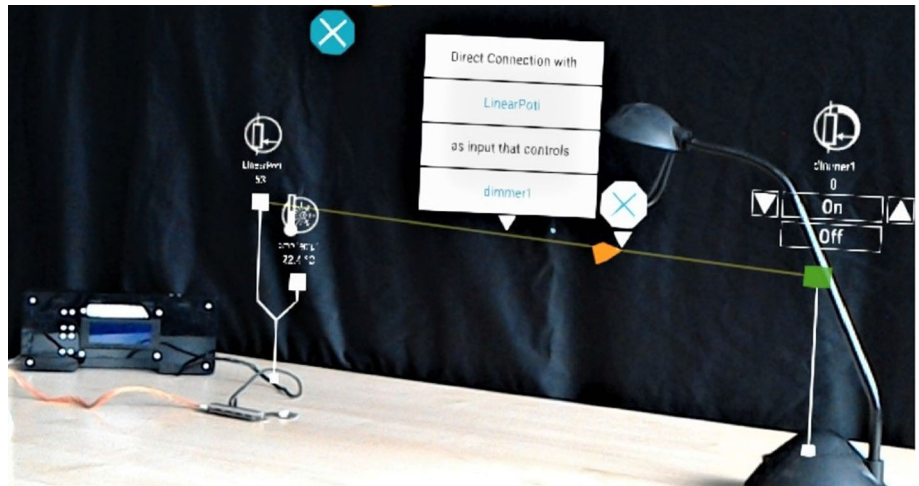


Fig. 10 Live view of an IoT process with an AND split

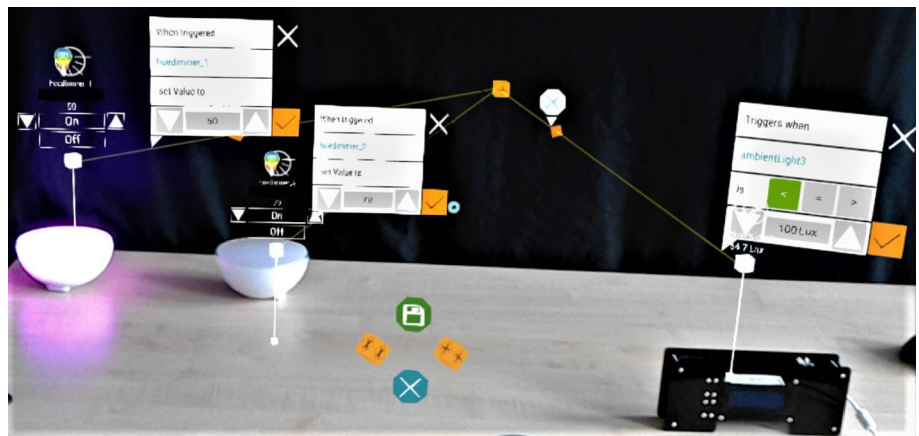


Fig. 11 Live view of an IoT process with an OR join

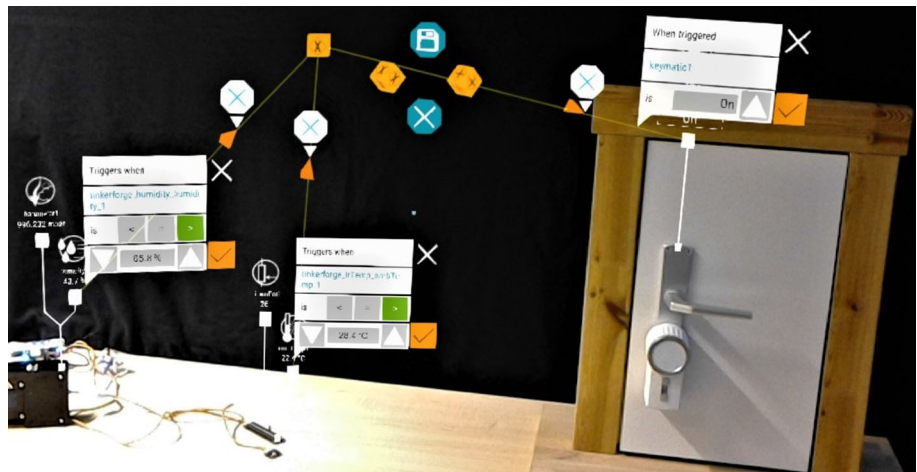
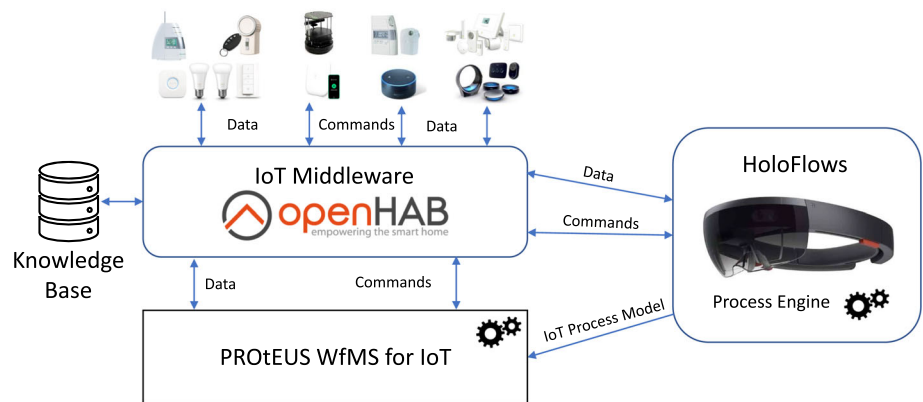


Fig. 12 Interactions between the HoloFlows App, IoT Middleware and Workflow Management System (WfMS) for Process Execution



Each modelled process features a “Play” button to switch to the *Process Execution Mode* and to execute an instance of this process directly on the device. The local process engine then activates the components according to the connections defined in the IoT process model. Sensor data are continuously retrieved from the IoT middleware via service-based interfaces. Actuator functionality is invoked directly from the device by calling RESTful Web services offered by the middleware. Depending on the type of connections and process, an instance stops after executing the last activities and needs to be instantiated anew, or the process instance runs continuously (for direct sensor–actuator connections) until the user decides to stop the instance.

4.4.2 Server-based execution

Once the IoT processes have been configured, tuned and sufficiently tested by the end-users on the device, they can be deployed to a full-fledged workflow management system (WfMS) [19] and executed independent of the MR device and HoloFlows app. The IoT process models created with HoloFlows are serialized based on the IoT process modelling language described in [90] and deployed to an IoT WfMS called “PROtEUS” [66] (cf. Fig. 12).

5 Evaluation

The goal of HoloFlows is to provide end-users with a simple intuitive approach for modelling and executing processes in IoT environments using MR technology (*Objectives O1, O2, O3, O4*). We claim that MR helps users to better understand the IoT environment, its devices and their relations. Users are able to put devices in their physical context and directly correlate them with others in the form of simple IoT processes without requiring deep IoT, process modelling or programming knowledge. Based on a preliminary study [89] and previous experience with MR applications [69], we designed and performed a user study to evaluate the HoloFlows approach including the process modelling notation and MR user interface artefacts. We let end-users model the same three exemplary IoT processes introduced in Sect. 2.1 and “Appendix” and thereby compare IoT process modelling in MR (with HoloFlows) with the two modelling approaches and associated tools introduced in Sect. 2.2: *Business Process-oriented Modelling* (with Camunda) and *Flow-based Modelling* (with Node-RED).

5.1 End-user study setup

The aim of our user study was to answer the following questions that arose from the literature and a preliminary study [89] to evaluate the suitability of the three approaches for end-user development of IoT processes:

1. Which modelling approach and tool performs best in terms of task completion time to create an IoT process?
2. Which approach and tool performs best concerning task load?
3. What is the perceived level of required experience to create IoT processes in terms of IoT knowledge, process modelling knowledge and programming knowledge for the three approaches?
4. What are advantages and disadvantages of the approaches and tools?

5.1.1 Participants

Thirteen participants were recruited personally or via e-mail to participate in our user study. Eleven participants had an academic background in terms of working at the university. Their ages ranged from 21 to 36 years ($M=30.1$, $SD=4.0$). Participants assessed their experience with (i) IoT, (ii) process modelling, (iii) programming, and (iv) augmented reality (AR), each on a 5-point scale with “0=no experience” to “4=expert”. With the answers “3” and “4”, three participants had prior experience with IoT; three participants had experience with (business) process modelling; eleven participants had experience with programming; and two participants had experience with AR.

5.1.2 Apparatus

Both tools Camunda Modeler (Version 3.4.1) and Node-RED (Version 1.0.2) were installed and presented on a Windows 10 PC with two displays (Fig. 13a). The left display showed textual instructions regarding the tool as well as the processes that had to be modelled. The right display presented each tool and was used for modelling. To introduce each approach, the study leader demonstrated the tool by modelling a sample process. The participants used a corded keyboard and mouse for interacting with the PC. For using HoloFlows, a first generation Microsoft HoloLens was utilized. Participants used the air tap gesture to interact with the HoloLens instead of using the HoloLens’s clicker device. The IoT devices were positioned in the middle of the laboratory where the user study took place (Fig. 13b–d). To introduce HoloFlows, the study leader started the HoloLens live stream on an external screen and demonstrated the usage of HoloFlows by modelling a sample process (Fig. 13c, e). Furthermore, the study leader used the live stream to check the participants’ progress

and to give hints when necessary. Figure 13 illustrates the overall study setting in our lab.

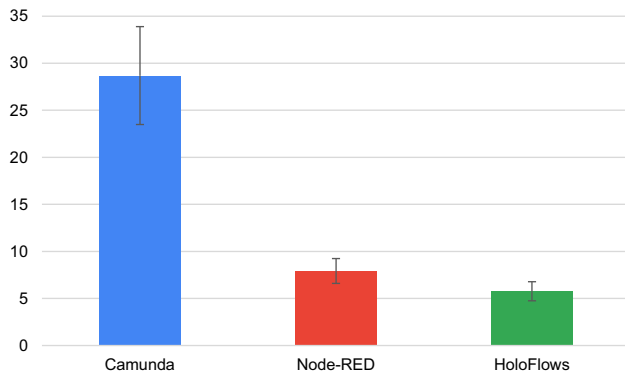
5.1.3 Procedure

To investigate the above-mentioned approaches and tools accordingly, participants performed the user study as follows. After the participants arrived in our lab, we explained the global procedure and the goal of the user study. The study leader explained the IoT setup, the three recurring IoT processes that the participants had to model (cf. IoT Process 1 in Sect. 2.1; IoT Process 2 and 3 in “Appendix”) and he described and demonstrated the first tool. After that, participants started to model the processes with the first tool. For each IoT process and each tool, detailed instructions were given in textual form for the users to follow on how to model the specific processes including the modelling elements to use and parameters to configure. If necessary, the study leader provided hints and guidance and gave final feedback for each modelled process. Afterwards, participants completed the NASA-TLX (Task Load Index) questionnaire [97]. This procedure was repeated with the other two tools.

A session concluded with a final questionnaire asking for an evaluation of required knowledge regarding (i) IoT, (ii) programming, and (iii) process modelling, each rated on a 5-point scale with “0=no experience” to “4=expert” and demographic data. It also included individual questions with free form text answers regarding the personal opinion about: (1) advantages/disadvantages of each tool; (2) experienced troubles with each tool; (3) advantages/disadvantages of modelling with desktop-based tools; (4) advantages/disadvantages of modelling in AR/MR; as well as (5) potential improvements of the HoloFlows app. During a session an observer took times and additional notes regarding a participant’s performance and remarks. A session took about 78 min ($M=77.7$, $SD=14.0$).

5.1.4 Design

The user study was performed as within-subjects study where each participant was tested on each condition. To avoid learning effects, we counterbalanced the tool testing order. We used the tool as independent variable with the three test conditions *Camunda*, *Node-RED* and *HoloFlows*. The dependent variable was the task completion time. Furthermore, we examined the workload of each tool using the NASA-TLX questionnaire and enquired about the experience with and advantages/disadvantages of each tool.

Fig. 13 Study setting in our lab**Fig. 14** Means and standard deviation of task completion time in minutes per tool

5.2 End-user study results

5.2.1 Task completion time

We compared the task completion time per approach. Modelling the three processes with Camunda took participants between 21 and 37 min ($M=28.7$, $SD=5.2$), with Node-RED between 6 and 10 min ($M=7.9$, $SD=1.3$), and with HoloFlows between 4 and 8 min ($M=5.8$, $SD=1.0$). Figure 14 illustrates the results, which also include the time for introducing the individual tool by the study leader. Performing an analysis of variance (ANOVA), we found that the effect of the tool on task completion time was statistically significant ($F_{2,24}=276.262$, $p<.0001$).

The results show a significantly longer task completion for the modelling of the IoT processes with Camunda. As the majority of participants was not familiar with BPMN 2.0, not only the introductory part required more time but also the modelling part. Participants had to be provided with detailed process descriptions, which took longer to study and transfer to the according process model. Despite being almost semantically identical to the Node-RED and HoloFlows processes, the BPMN 2.0 representations of the IoT processes show a higher degree of complexity due to the business process-oriented view and modelling concepts. There are less differences in the task completion times between modelling

in Node-RED and HoloFlows. In HoloFlows, devices are already integrated and set up such that users only have to create connections and specify the conditions and actions that are part of the process (cf. *Objectives O1, O3*). In Node-RED, users also have to create the corresponding nodes for devices and set parameters (e.g. topics and service URIs).

5.2.2 Comparison of task load

After completing the modelling of the three IoT processes following one approach, participants were asked to fill out the NASA-TLX to assess the perceived individual task load. We decided to let users evaluate their work load based on the NASA-TLX [97] as this index is more in line with the objectives we are investigating regarding task difficulty as well as mental and physical efforts of IoT process modelling, in contrast with for example the technology acceptance model (TAM) [98] or the system usability scale (SUS) [99] focusing on the perceived usefulness and usability of a system or application. Especially the new interaction forms in MR introduced by the HoloLens glasses may add to the physical efforts of the modelling tasks, which we are interested in investigating with the NASA-TLX. Figure 15 presents a comparison of the TLX means for all approaches. We used ANOVA to investigate the statistical significance for the overall task load. The results show that the effect of the tool on overall task load was statistically significant ($F_{2,24}=18.329$, $p<.0001$).

Due to the high complexity of the BPMN 2.0 language, its limited suitability to represent important IoT concepts in an easy way (e.g., continuous publish-subscribe to sensors) and the lack of BPMN 2.0 knowledge among the majority of the study participants, the overall task load and most of the individual factors are significantly higher for the Camunda modelling tool. For the HoloFlows tool, especially the physical demand is higher compared to the WIMP-based modelling tools as the HoloLens is rather heavy, uncomfortable to wear and has only a small field of view. In addition, the air tap gestures necessary to interact with the IoT devices involve more arm movements, which quickly become physically demanding. The usage of the clicker

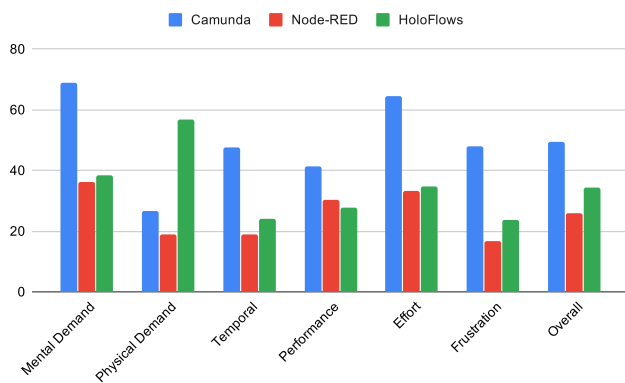


Fig. 15 Comparison of TLX means for modelling IoT processes with all tools

device accompanying the HoloLens may mitigate this factor, which was also mentioned by the participants. The second generation of the HoloLens will further reduce some of the aforementioned limitations. Compared to Node-RED, also the level of frustration for working with HoloFlows was rated a bit higher, which can be explained by the reliability of the gesture recognition and some interface elements of HoloFlows being too small for a precise selection. We will improve this issue in future releases.

5.2.3 Required experience to model IoT processes

As part of the questionnaire to be filled out after completing the modelling tasks, participants were asked to assess the knowledge/experience required to model IoT processes in Camunda, Node-RED and HoloFlows with regard to the areas of IoT, (business) process modelling and programming. Figure 16 shows the means of these three aspects for each tool. Again, we applied an ANOVA to validate the results. The effect of the tool on each kind of experience was statistically significant: IoT experience ($F_{2,24} = 11.573, p < .0005$), process modelling experience ($F_{2,24} = 74.548, p < .0001$) and programming experience ($F_{2,24} = 5.414, p < .05$).

Camunda being a classical business process modelling tool was perceived as mostly requiring process modelling knowledge as well as a certain level of programming and almost no level of IoT experience. Due to the focus on specifying functional processes (flows) among specific IoT-related entities that also have to be parameterized in a very detailed manner, Node-RED was perceived as requiring a higher level of IoT and programming knowledge. HoloFlows required the lowest level of experience in the areas of process modelling and programming as users only have to create simple connections among the existing IoT devices and specify *what* should be happening *when* (cf. *Objective O1*).

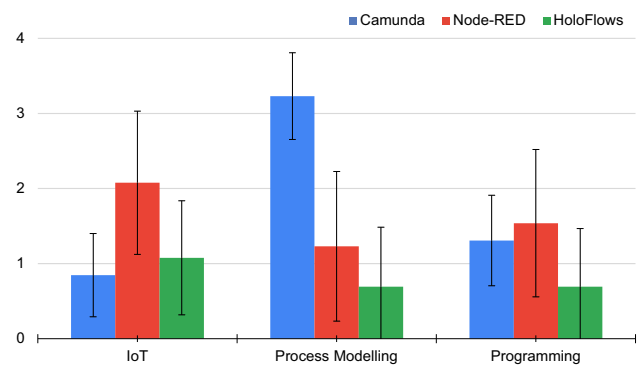


Fig. 16 Means and standard deviation of required IoT process knowledge per tool as perceived by the participants (“0=no experience”, “4=expert”)

5.2.4 Observations

From the observer’s notes taken during the individual experiments, we can summarize additional findings as follows:

- 1) We observed different kinds of learning curves for all three approaches. Participants without prior knowledge regarding business process modelling showed a steep learning curve after completing the modelling of the first exemplary IoT process. While the majority of participants struggled especially with modelling the split and join gateways in the first BPMN-based process, they were able to quickly transfer this newly gained knowledge to similar parts of the second and third process. The study leader had to provide hints and assistance mostly during the modelling of the first and third IoT process. After modelling the three processes in BPMN 2.0, the participants felt quite confident regarding their level of expertise in business process modelling. For the Node-RED- and HoloFlows-based approaches, the introductions and demonstrations by the study leader at the beginning of each modelling task were sufficient for the users to get started and complete the modelling without much assistance (cf. task completion times and NASA-TLX results). Due to the higher amount of interactions and configurations, participants preferred to use keyboard shortcuts and copy-paste when modelling processes in the Camunda and Node-RED tools.
- 2) Prior knowledge was mostly helpful for participants regarding the business process-oriented modelling tasks as these were the most complex processes (cf. NASA-TLX results). Participants with prior process modelling knowledge needed less assistance and time to model the IoT processes following the textual instructions and experienced a lower task load than other participants. As all three approaches aim at low-/no-code development of processes, existing programming knowledge did not have

a significant influence on the tasks, which corresponds to the users' evaluations of the level of required programming knowledge. Also, prior IoT knowledge was not entirely necessary as detailed instructions on how to configure the respective IoT devices were given as part of the textual process descriptions. Users were especially surprised about the short amount of time it took to model the IoT processes using the Node-RED and HoloFlows tools compared to Camunda (cf. task completion times).

5.2.5 Advantages and disadvantages of the modelling approaches and tools

As part of the final questionnaire the participants were asked to give their opinions about advantages and disadvantages of the individual approaches and comparison of desktop-based modelling using a "classical" GUI vs. modelling in MR.

Camunda (Business Process-oriented Modelling) According to the participants the **advantages** of business process-oriented modelling include that BPMN 2.0 as a standard facilitates knowledge exchange and a common understanding of (business) processes. Pools and lanes provide a good overview of the process participants, their relations and the overall process flow. Processes can be specified in a detailed manner and almost no IoT knowledge is necessary for modelling processes, which is also reflected in the quantitative results of the user study. The Camunda tool relies on known WIMP interaction principles, which is why participants felt confident in terms of interactions when using the tool. Compared to modelling in AR/MR the tool can be used independent of the physical location to create process models. **Disadvantages** of the approach that users perceived include the complexity of the set of BPMN 2.0 modelling elements requiring a profound knowledge of the language to find a possible way of modelling specific aspects and level of abstraction. Examples helped to understand and learn BPMN 2.0 better. Due to the ambiguity of modelling elements and different ways of representing specific aspects, users stated that they would rather feel unconfident about the correctness of modelled processes without having a detailed description provided, which can also be observed quantitatively in the results of the NASA-TLX analysis.

Node-RED (Flow-based Modelling) The **advantages** of the Node-RED tool include the use of common WIMP concepts for modelling interactions and the ability to create IoT processes independent of the current physical location. Participants found the tool easy to understand, well arranged and intuitive for flow-based modelling of processes among the individual nodes. They also highlighted the support of many relevant IoT concepts and technologies via the functional nodes. **Disadvantages** were seen in the high degree of technical and IoT-related expertise required to configure the individual nodes within flow-based modelling and the large

amount of nodes not always clearly stating their purpose, which made the tool's UI too complex for some users (cf. Performance aspect in NASA-TLX results). Similar to business process-oriented modelling on classical GUIs, the specification of IoT processes requires an abstract understanding of the devices' physical contexts as there is no direct physical correlation possible.

HoloFlows (Mixed Reality-based Modelling) Participants especially highlighted the interactions in MR via the HoloLens as most joyful and innovative way of directly interacting with the IoT environment (cf. *Objective O3*). Regarding the **advantages** of the HoloFlows approach, participants listed the direct correlation of the physical devices with their virtual representation as one of the major points (cf. *Objective O2*). The approach was also perceived as being very intuitive, easy to learn and in general a positive experience (cf. *Objectives O1, O4*). The users enjoyed interacting with the IoT devices via the tool in this rather unfamiliar way. Among the **disadvantages** of the tool, the participants listed the increased physical activities and demand (cf. results of NASA-TLX) as well as the current technical limitations of the HoloLens. With many IoT devices and processes the participants fear that the overview can be lost due to many distributed and probably overlapping holograms bloating the MR user interface. To use HoloFlows the corresponding MR device and physical presence are required.

5.2.6 Validity and limitations

The group of participants comprised test subjects with a computer science background mostly from academia and in the age range from 21 to 36 years. These subjects are well-educated and open to work with new technologies (e.g. MR and IoT) as early adopters. They also possess the ability of thinking in abstract models and algorithms from their professional experience, which is why most of the participants did not have much troubles with modelling the IoT processes. While this rather small—but already statistically significant (cf. results of ANOVA)—user group is suitable for a first study to gain insights into IoT process modelling with classical GUIs and MR [100], e.g. regarding required knowledge and possible improvements of the processes and applications, further user studies have to be conducted with a wider variety of participants representing end-users from different backgrounds and with different levels of experience and skills. We expect that the general trends and relations regarding task completion time, task load and perceived required knowledge will also appear within the results in these future studies.

For this first study, we relied on detailed descriptions specifying how the individual IoT processes were supposed to be modelled with the respective tools. Users followed these descriptions and were guided by the study leader in case of major deviations, errors or questions. The resulting pro-

cess models for each participant are therefore very similar. In future studies, these instructions should be reduced to the general descriptions of the individual IoT process as given in Sect. 2.1 and “Appendix” and users should try to model these processes on their own. That way the correctness and functionality of the created IoT process models can also be evaluated. For this task, we expect HoloFlows to perform significantly better than the other tools as there is no need for configuring technical details and parameters and process modelling is more straightforward.

The three IoT processes presented in the Background and “Appendix” sections as basis for the modelling tasks in the user study are semantically equivalent for each approach. However, they are of different complexity due to the specific concepts the modelling approaches are based on. HoloFlows relies on IoT devices already being integrated and presented in the MR scene so that users only have to create and configure the connections among them to specify executable processes (cf. *Objectives O1, O3, O4*). In the Camunda and Node-RED based approaches, the IoT entities also have to be modelled and configured to be part of an IoT process. The approaches and results can therefore only be partially compared. However, the insights and trends observed via the study serve as a good starting point to analyse the approaches in more detail and improve HoloFlows.

5.3 Discussion of HoloFlows

5.3.1 Objective O1: simple IoT process modelling language

The process modelling notation in HoloFlows is a DSL designed for the IoT domain. It views IoT devices as first-class citizens and connections among these devices as basic building blocks of IoT processes. These concepts facilitate the end-user oriented development as they are closer and directly related to the real world objects [5,31]; thus, end-users are more familiar with these concepts from everyday life experiences [94], which was also observed in the user study. The other two approaches—business process-oriented and flow-based modelling—rely on more abstract and generic representations and proxies of the IoT devices in the form of pools, lanes, events, (service) tasks/activities [7], or nodes for protocol-dependant software modules and services [29]. These abstractions are more generic and allow for integrating a wider variety of process entities and software components. However, they require expert knowledge for configuration and parameterization during modelling. In HoloFlows, these steps are offloaded to pre-modelling steps where IoT experts have to integrate the IoT devices into the IoT middleware and knowledge base (cf. Sect. 4.4). End-users are then only presented with the available instances of IoT devices for process modelling in the HoloFlows app.

The concepts of the HoloFlows DSL are kept at a minimum to reduce complexity, still its expressiveness is sufficient to create simple IoT processes among the sensors and actuators of an IoT environment to automate most of the basic routines that end-users will create [14]. With connections and gateways as basic concepts of an IoT process, their complexity can simply be extended to longer sequences of connections to be executed as part of a process. However, the DSL’s expressiveness is limited to automation rules in IoT processes, which is similar to the flow-based modelling with Node-RED. Business process-oriented modelling features more general purpose languages and elements to express business-related and organizational aspects [19]. Extensions and combinations of these modelling elements with IoT-related aspects for integration purposes along the automation pyramid [101] have been proposed in various works (cf. Sect. 3.1). While these business-related aspects are not relevant for end-user oriented modelling of IoT processes [14], a mapping and abstraction of the elements of the HoloFlows DSL to a BPM language (e.g. BPMN 2.0) are straightforward [13], which makes the processes created with HoloFlows also available in a BPM context.

5.3.2 Objective O2: visualizing physical relations

One of the key features of the HoloFlows is that it leverages MR technology to augment the physical IoT devices with additional holographic information about the devices and existing processes [102] enabling the exploration of the smart space, its existing IoT devices and processes (instances) [87]. The IoT process examples in MR presented in Sect. 4.3 show that the overlay of information according to the HoloFlows user interface (cf. Sect. 4.2) works well in environments with few simple IoT devices that are spatially distributed. However, with an increasing number and more complex IoT devices (e.g. in a smart factory or smart car) the MR user interface may quickly become overloaded and confusing, which requires additional visualization concepts for grouping to provide a better overview and the possibilities to zoom in for details [103]. Due to relying on MR for modelling in HoloFlows, end-users have to be physically present in the respective IoT environment and can only create processes among the available IoT device instances [102].

The flow-based and business process-oriented modelling approaches rely on abstract representations of the IoT devices and on classical GUIs for process modelling. The physical relations of the IoT devices/process participants are therefore not directly visible—modellers have to create the corresponding mental abstractions [10] based on the connections between process elements in pools or between nodes of a flow. While HoloFlows focuses on the creation of IoT processes between concrete instances of devices, the business process-oriented and flow-based approaches usually spec-

ify the process participants on a more abstract type level and do not require the devices to be available during modelling. In contrast with HoloFlows, the representations of process elements in the desktop-based tools enable the modelling of processes from remote locations and provide better overviews in complex scenarios and process landscapes.

5.3.3 Objective O3: experimentation and direct feedback

The ability to experiment and receive direct feedback about the created processes increases acceptance in end-user development [5]. In HoloFlows, on-device execution and MR enable end-users to directly control IoT devices and execute modelled processes with immediate feedback about the process execution in the physical world. Based on this feedback, processes can be easily modified and tuned to suit the users' needs. These possibilities were positively received by the study participants who were eager to continue using HoloFlows and create new processes.

The Node-RED tool for flow-based modelling of IoT processes also supports the direct execution and modification of created processes—without the direct physical correlation achieved via MR, though. The same holds for business process-oriented modelling in Camunda. However, here additional steps and configurations are necessary to make a model executable, deploy it to a WfMS and to then instantiate it for execution [19]. Changes to the modelled processes require a re-deployment, which may quickly become annoying for fine-grained parameter tuning and testing.

5.3.4 Objective O4: end-user friendly interface

The results of the user study show that the developed MR user interface in HoloFlows leads to a positive user experience, steep learning curve and high effectiveness and efficiency when it comes to modelling IoT processes by end-users. The UI elements are closely aligned with the elements of the simple process modelling language, which does not introduce additional overhead and UI concepts within the HoloFlows app. HoloFlows simplifies the process modelling in IoT to the act of “drawing” virtual wires between devices and setting some parameters—a deep knowledge of programming, IoT or modelling concepts is not required as shown in our study, which complies with Fryling's et al. idea of low/no-code development [6].

Our user study also shows that both the flow-oriented and the business process-oriented approaches suffer from the complex underlying process modelling languages and associated UI modelling elements and configuration options in the rather bloated modelling applications. This resulted in inexperienced users quickly becoming overwhelmed with all the modelling choices and options in the WIMP-based tools. Both approaches and tools—Camunda and Node-RED—do

not seem to be completely suitable for end-user oriented (no-code) development as they still require a certain amount of modelling and technical knowledge.

5.3.5 Application domains

The presented *HoloFlows* approach is mainly targeted at supporting end-users with modelling IoT processes in the smart home domain. Here end-users usually do not possess deep programming skills, domain expertise or technical knowledge regarding the IoT devices and their configuration. However, consumer IoT appliances for the smart home are usually affordable mass market products and therefore relatively widespread [1], which is why there is a lot of potential for going beyond the control of individual devices and having the devices interact in *IoT processes* [3]. Moreover, the offered functionalities of common smart home devices are of rather low complexity and designed to be used more easily. This goes in line with the simplified HoloFlows DSL aiming at creating rather simple IoT processes that make use of these high-level functionalities and data (cf. *Objective O1*), but is still sufficient to cover most of modelling and automation aspects in this domain [14]. The concepts developed as part of HoloFlows—mostly regarding the DSL but also the visualizations and interactions (cf. *Objective O4*)—can be applied to other smart spaces where simple high-level functionality and data should be visualized and combined as part of discrete IoT processes (e.g. smart offices, smart buildings and smart cities). The suitability of HoloFlows in expert IoT domains with more complex individual devices and continuous or safety-critical processes (e.g. smart factories or smart vehicles) has to be further evaluated and extended. It is currently limited to the aforementioned less complex IoT devices with discrete high-level actions and data that benefit from a visualization of their physical (spatial) relations in MR (cf. *Objective O2*), whereas the actual IoT process implementation and execution in these domains should be realized on much lower levels of the automation pyramid [101] (cf. *Objective O3*).

6 Conclusion and future work

In this work, we investigated the application of MR technology to support end-users with modelling of basic processes in IoT domains and compared it to traditional desktop-based process modelling approaches. The developed *HoloFlows* approach consists of a minimal IoT process modelling DSL, a mixed reality user interface aligned with the DSL and an application for modelling and executing the IoT processes. HoloFlows can be viewed as a novel, intuitive and easy-to-use approach for IoT exploration and end-user programming. It enables non-experts to create processes in the

respective IoT domain in a *no-code* fashion without process or modelling knowledge. In contrast with related work and approaches focusing on traditional GUIs and BPMN 2.0 extensions that rely on abstract representations of process participants and other elements, HoloFlows exploits MR technology and the physical contexts of devices to directly connect and coordinate instances of sensors and actuators of the IoT environment to form basic real-world processes that can be composed to more complex IoT processes. With HoloFlows, we present a vision of combining new emerging technologies—Holograms, MR and IoT—with the BPM domain for automation. The results of a user study conducted in the context of a smart home attest HoloFlows with providing an increased usability (e.g. better task completion times) and user experience (e.g. positive feelings while using HoloFlows) compared to desktop-based (IoT) process modelling approaches, which makes HoloFlows a suitable approach for end-user development in BPM and IoT.

Future work includes a revision of the MR user interface with respect to the suggestions by the study participants; the extension of supported IoT devices (e.g. smart speakers) and process elements also including human tasks and purely virtual entities; and a simplification of the UI to further ease the parameter configurations and provide better overviews of IoT environments with many devices and processes. Porting the application to the next generation of smart glasses will further

improve the user interface due to more mature MR technology. We will also work on a collaborative use of HoloFlows with multiple parallel users as well as means for combining HoloFlows with desktop-based modelling tools, e.g. to share and visualize process models and instances among the applications. Besides more extensive user studies in the smart home, we will also investigate HoloFlows in smart factories and smart facility management as IoT domains with more complex IoT devices and processes.

Acknowledgements This research has received funding under the Grant No. 100268299 by the European Social Fund (ESF) and the German Federal State of Saxony.

Funding Open Access funding provided by Universität St.Gallen

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

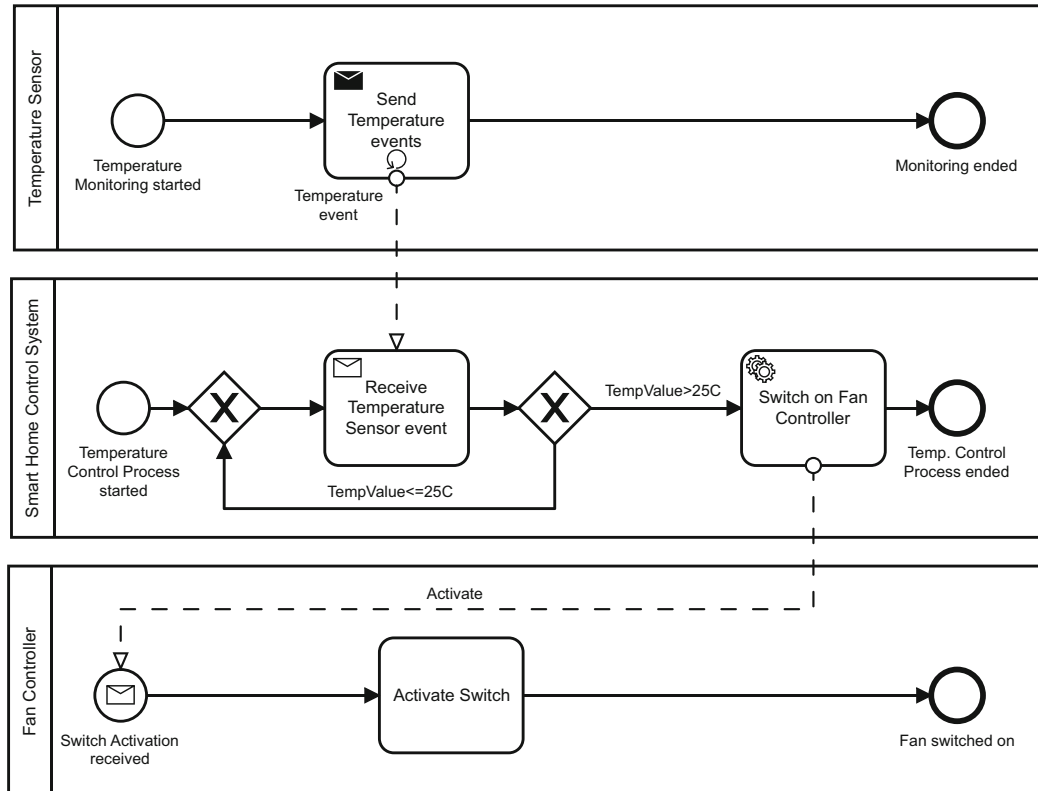


Fig. 17 Example IoT process 2 modelled in BPMN 2.0

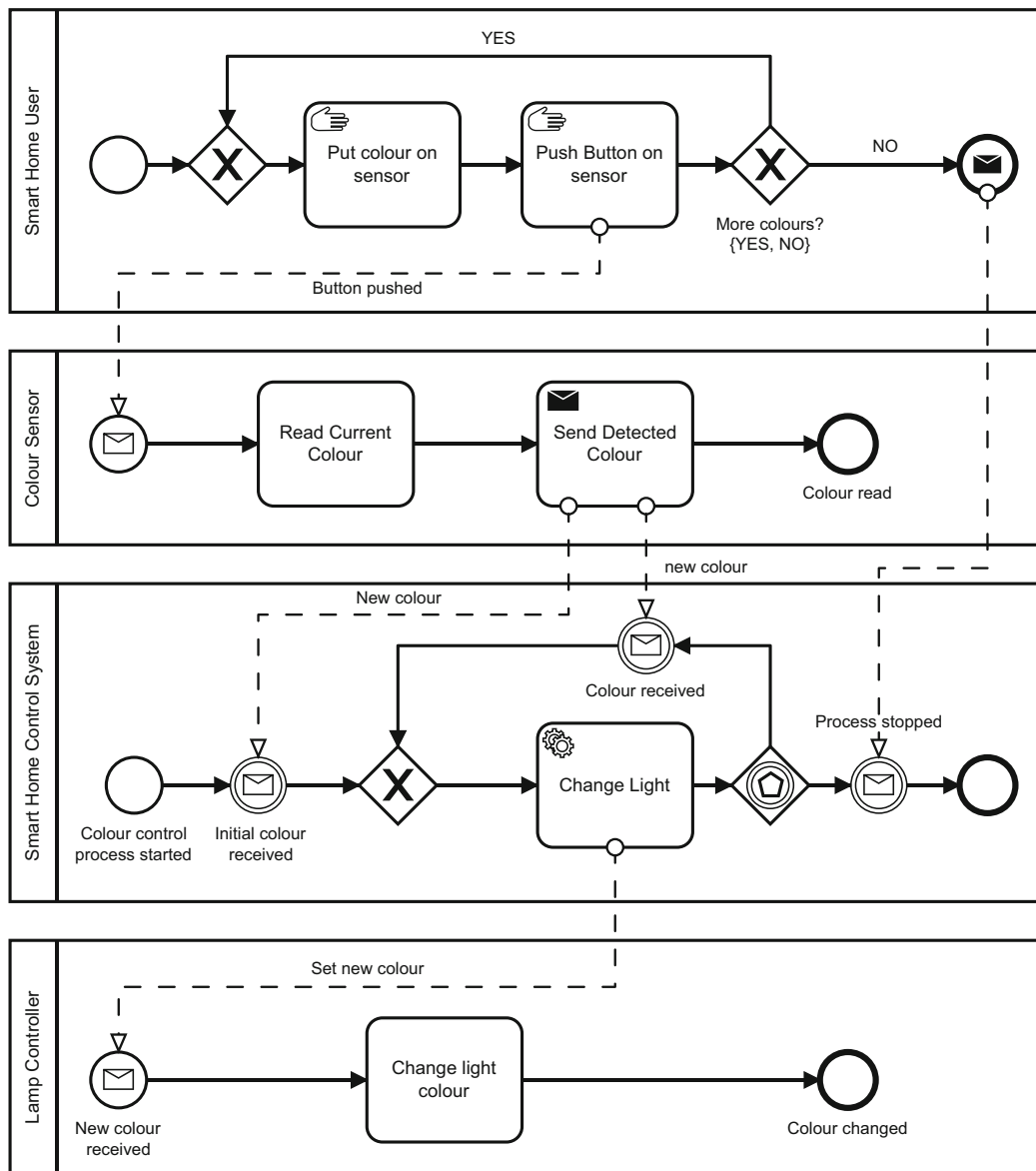


Fig. 18 Example IoT process 3 modelled in BPMN 2.0

Appendix: IoT process examples

In addition to the IoT process (IoT Process 1) described in Sect. 2.1, the following two IoT processes were used as part of modelling tasks in the user study described in Sect. 5.

IoT Process 2 In the process, data from a temperature sensor are continuously collected and analysed. Once the temperature is over 25°C, a fan is switched on and the process ends afterwards.

IoT Process 3 In the process, data from a colour sensor are continuously collected and transferred to a lamp to set it as its current colour. The user has to put a specific colour on the sensor and push a button for the sensor to read the value.

Business process-oriented modelling

Figure 17 shows the IoT Process 2 in BPMN 2.0. Participants of this process are the temperature sensor, the smart home control system and the fan controller. The temperature sensor continuously sends new temperature events via a *Send Task* to the smart home control system. Here these events are received via a *Receive Task*, and based on the temperature values either a *Service Task* is executed to switch on the fan controller (temperature value > 25), or the process goes back and waits to receive new temperature events (temperature value ≤ 25). An *Exclusive Gateway* is used to model the split and join at this point. The fan controller receives the service call in

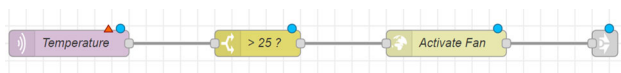


Fig. 19 Example IoT process 2 specified in node-RED



Fig. 20 Example IoT process 3 specified in node-RED

a *Message Start Event* and starts the process to activate the light switch as a *Task*.

Figure 18 shows the IoT Process 3 in BPMN 2.0. Participants of this process are the smart home user, the colour sensor, the smart home control system and the lamp controller. In two subsequent *Manual Tasks* the user first puts a colour on the sensor and then pushes a button on the sensor, which sends a message to the colour sensor. The user decides whether to repeat this process with more colours or to stop the process (modelled by an exclusive gateway), which is signalled to all participants as a *Message End Event*. The process on the colour sensor starts with receiving a *Message Start Event* after the button has been pushed. The sensor then reads the current colour in a *Task* and sends the detected colour afterwards in a *Send Task*. The process already running in the smart home control system waits to receive the initial colour from the colour sensor's send task in a *Message Intermediate Catch Event*. After receiving this event, a *Service Task* is used to call the lamp controller to change the light colour. The process on the control system then waits within an *Event-based Gateway* to either receive a new colour from the sensor or a stop event from the user, both via a *Message Intermediate Catch Event*. The lamp controller starts with a *Message Start Event* received as a consequence of executing the control system's service task. Afterwards a *Task* is used to model the light colour change.

Flow-based modelling

Figure 19 shows the IoT Process 2 in Node-RED. An *MQTT In* node is used to subscribe to the data produced by the temperature sensor. Its output port is directly connected to the input port of a *Switch* node, which contains the condition that if the payload of the input message is above 25°C then its first output port should be activated. This port is wired to an *HTTP Request* to activate the IoT service controlling the fan actuator. If the condition is not fulfilled, the process waits for new events from the sensor. A *Link Out* node is used to indicate the end of the process after the HTTP request.

Figure 20 shows the IoT Process 3 in Node-RED. An *MQTT In* node is again used and parameterized to subscribe to the data produced by the colour sensor. Its output port is directly wired to the input port of an *HTTP Request* node configured

to call a RESTful IoT service, which controls the corresponding lamp actuator to change the light colour.

References

1. Wilson, C., Hargreaves, T., Hauxwell-Baldwin, R.: Smart homes and their users: a systematic analysis and key challenges. *Pers. Ubiquit. Comput.* **19**(2), 463–476 (2015)
2. Weiser, M.: The computer for the 21st century. *Sci. Am.* **265**(3), 94–104 (1991)
3. Janisch, C., Koschmider, A., Mecella, M., Weber, B., Burattin, A., Ciccio, C.D., Fortino, G., Gal, A., Kannengiesser, U., Leotta, F., Mannhardt, F., Marrella, A., Mendling, J., Oberweis, A., Reichert, M., Rinderle-Ma, S., Serral, E., Song, W., Su, J., Torres, V., Weidlich, M., Weske, M., Zhang, L.: The internet-of-things meets business process management: a manifesto. *IEEE SMC Mag.* (2020)
4. Hasic, F., Serral Asensio, E.: Executing iot processes in bpmn 2.0: Current support and remaining challenges. In: *IEEE RCIS 2019 Proceedings* (2019)
5. Henriksen, P., Kölling, M.: Greenfoot: combining object visualisation with interaction. In: *Companion to the 19th Annual ACM SIGPLAN Conference on Object-Oriented Programming Systems, Languages, and Applications*, pp. 73–82 (2004)
6. Fryling, M.: Low code app development. *J. Comput. Sci. Coll.* **34**(6), 119–119 (2019)
7. Torres, V., Serral, E., Valderas, P., Pelechano, V., Grefen, P.: Modeling of iot devices in business processes: A systematic mapping study. In: *2020 IEEE 22nd Conference on Business Informatics (CBI)*, vol. 1, pp. 221–230. *IEEE* (2020)
8. Morrison, J.P.: Flow-based programming: a new approach to application development. CreateSpace (2010)
9. OMG: Business Process Model and Notation (BPMN), Version 2.0 (2011)
10. Van Dam, A.: Post-wimp user interfaces. *Commun. ACM* **40**(2), 63–67 (1997)
11. Milgram, P., Kishino, F.: A taxonomy of mixed reality visual displays. *IEICE Trans. Inf. Syst.* **77**(12), 1321–1329 (1994)
12. Peffers, K., Tuunanen, T., Rothenberger, M.A., Chatterjee, S.: A design science research methodology for information systems research. *J. Manag. Inf. Syst.* **24**(3), 45–77 (2007)
13. Seiger, R., Gohlke, M., Abmann, U.: Augmented reality-based process modelling for the internet of things with holoflows. In: Reinhartz-Berger, I., Zdravkovic, J., Gulden, J., Schmidt, R. (eds.) *Enterprise, Business-Process and Information Systems Modeling*, pp. 115–129. Springer, Cham (2019)
14. Barricelli, B.R., Valtolina, S.: Designing for end-user development in the internet of things. In: *International Symposium on End User Development*, pp. 9–24. Springer (2015)
15. Balakrishna, S., Thirumaran, M.: Programming paradigms for IoT applications: an exploratory study. *Handbook of IoT and Big Data*, pp. 23–57 (2019)
16. Gubbi, J., Buyya, R., Marusic, S., Palaniswami, M.: Internet of things (IoT): a vision, architectural elements, and future directions. *Fut. Gen. Comput. Syst.* **29**(7), 1645–1660 (2013)
17. Bauer, M., Bui, N., De Loof, J., Magerkurth, C., Nettsträter, A., Stefa, J., Walewski, J.W.: Iot reference model. In: *Enabling Things to Talk*, pp. 113–162. Springer (2013)
18. Fielding, R.T., Taylor, R.N.: Principled design of the modern web architecture. *ACM Trans. Internet Technol.* **2**(2), 115–150 (2002)
19. Dumas, M., La Rosa, M., Mendling, J., Reijers, H.A., et al.: *Fundamentals of Business Process Management*, vol. 1. Springer, Berlin (2013)

20. Biggs, G., MacDonald, B.: A survey of robot programming systems. In: *Proceedings of the Australasian Conference on Robotics and Automation*, pp. 1–3 (2003)
21. Thomas, U., Hirzinger, G., Rumpe, B., Schulze, C., Wortmann, A.: A new skill based robot programming language using uml/p statecharts. In: *2013 IEEE International Conference on Robotics and Automation*, pp. 461–466. IEEE (2013)
22. Montano, L., García, F.J., Villarroel, J.L.: Using the time petri net formalism for specification, validation, and code generation in robot-control applications. *Int. J. Robot. Res.* **19**(1), 59–76 (2000)
23. Peterson, J., Hager, G.D., Hudak, P.: A language for declarative robotic programming. In: *Proceedings 1999 IEEE International Conference on Robotics and Automation* (Cat. No. 99CH36288C). vol. 2, pp. 1144–1151. IEEE (1999)
24. for Standardization/International Electrotechnical Commission, I.O., et al.: *Iso/iec 19510: 2013. Information Technology—Object Management Group Business Process Model and Notation* (2013)
25. Chang, C., Srirama, S.N., Buyya, R.: Mobile cloud business process management system for the internet of things: a survey. *ACM Comput. Surv.* **49**(4), 70 (2016)
26. Recker, J.C.: Bpmn modeling—who, where, how and why. *BPTrends* **5**(3), 1–8 (2008)
27. Genon, N., Heymans, P., Amyot, D.: Analysing the cognitive effectiveness of the bpmn 2.0 visual notation. In: *International Conference on Software Language Engineering*, pp. 377–396. Springer (2010)
28. Krishnamurthi, R.: Teaching methodology for iot workshop course using node-red. In: *11th International Conference on Contemporary Computing (IC3)*, pp. 1–3. IEEE (2018)
29. Lekić, M., Gardašević, G.: Iot sensor integration to node-red platform. In: *17th International Symposium Infoteh-Jahorina (Infoteh)*, pp. 1–5. IEEE (2018)
30. Chaczko, Z., Braun, R.: Learning data engineering: Creating iot apps using the node-red and the rpi technologies. In: *16th International Conference on Information Technology Based Higher Education and Training (ITHET)*, pp. 1–8. IEEE (2017)
31. Repenning, A., Ioannidou, A.: What makes end-user development tick? 13 design guidelines. In: *End User Development*, pp. 51–85. Springer (2006)
32. Jacob, R.J., Girouard, A., Hirshfield, L.M., Horn, M.S., Shaer, O., Solovey, E.T., Zigelbaum, J.: Reality-based interaction: a framework for post-wimp interfaces. In: *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 201–210 (2008)
33. Farshid, M., Paschen, J., Eriksson, T., Kietzmann, J.: Go boldly!: Explore augmented reality (ar), virtual reality (vr), and mixed reality (mr) for business. *Bus. Horiz.* **61**(5), 657–663 (2018)
34. Azuma, R., Bailiot, Y., Behringer, R., Feiner, S., Julier, S., MacIntyre, B.: Recent advances in augmented reality. *IEEE Comput. Gr. Appl.* **21**(6), 34–47 (2001)
35. Avila, L., Bailey, M.: Augment your reality. *IEEE Comput. Gr. Appl.* **36**(1), 6–7 (2016)
36. Poole, E.S., Le Dantec, C.A., Eagan, J.R., Edwards, W.K.: Reflecting on the invisible: understanding end-user perceptions of ubiquitous computing. In: *Proceedings of the 10th International Conference on Ubiquitous Computing*, pp. 192–201 (2008)
37. Mendling, J., Baesens, B., Bernstein, A., Fellmann, M.: Challenges of smart business process management: An introduction to the special issue. *Decision Support Systems* (2017)
38. Janiesch, C., Koschmider, A., Mecella, M., Weber, B., Burattin, A., Di Ciccio, C., Gal, A., Kannengiesser, U., Mannhardt, F., Mendling, J., et al.: The internet-of-things meets business process management: mutual benefits and challenges. *arXiv preprint arXiv:1709.03628* (2017)
39. Leotta, F., Mecella, M., Mendling, J.: Applying process mining to smart spaces: Perspectives and research challenges. In: *Advanced Information Systems Engineering Workshops*, pp. 298–304. Springer (2015)
40. Leotta, F., Marrella, A., Mecella, M.: Iot for bpmers. challenges, case studies and successful applications. In: *International Conference on Business Process Management*, pp. 16–22. Springer (2019)
41. Fischer, G., Giaccardi, E., Ye, Y., Sutcliffe, A.G., Mehndjiev, N.: Meta-design: a manifesto for end-user development. *Commun. ACM* **47**(9), 33–37 (2004)
42. Brouns, N., Tata, S., Ludwig, H., Asensio, E.S., Grefen, P.: Modeling iot-aware business processes—a state of the art report. *arXiv preprint arXiv:1811.00652* (2018)
43. Domingos, D., Martins, F., Cândido, C.: Internet of things aware WS-BPEL business processes—context variables and expected exceptions. *J. UCS* **20**(8), 1109–1129 (2014)
44. Sungur, C.T., Spiess, P., Oertel, N., Kopp, O.: Extending BPMN for Wireless Sensor Networks. In: *IEEE 15th Conference on Business Informatics*, pp. 109–116 (2013)
45. Meyer, S., Ruppen, A., Magerkurth, C.: Internet of things-aware process modeling: integrating IoT devices as business process resources. *Lecture Notes in Computer Science 7908 LNCS*, pp. 84–98 (2013)
46. Meyer, S., Ruppen, A., Hilty, L.: The things of the internet of things in bpmn. In: *Advanced Information Systems Engineering Workshops*, pp. 285–297 (2015)
47. Graja, I., Kallel, S., Guermouche, N., Kacem, A.H.: Bpmn4cps: A bpmn extension for modeling cyber-physical systems. In: *IEEE 25th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE)*, pp. 152–157 (2016)
48. Martins, F., Domingos, D.: Modelling iot behaviour within BPMN business processes. *Procedia Comput. Sci.* **121**, 1014–1022 (2017)
49. Suri, K., Gaaloul, W., Cuccuru, A., Gerard, S.: Semantic framework for internet of things-aware business process development. In: *IEEE 26th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE)*, pp. 214–219. IEEE (2017)
50. Friedow, C., Völker, M., Hewelt, M.: Integrating iot devices into business processes. In: *International Conference on Advanced Information Systems Engineering*, pp. 265–277. Springer (2018)
51. Petrasch, R., Hentschke, R.: Process modeling for industry 4.0 applications: towards an industry 4.0 process modeling language and method. In: *2016 13th International Joint Conference on Computer Science and Software Engineering (JCSSE)*, pp. 1–5. IEEE (2016)
52. Bocciarelli, P., D’Ambrogio, A., Giglio, A., Paglia, E.: A bpmn extension for modeling cyber-physical-production-systems in the context of industry 4.0. In: *2017 IEEE 14th International Conference on Networking, Sensing and Control (ICNSC)*, pp. 599–604. IEEE (2017)
53. Vanderfeesten, I., Erasmus, J., Traganos, K., Bouklis, P., Garbi, A., Bouladakis, G., Dijkman, R., Grefen, P.: Developing process execution support for high-tech manufacturing processes. In: *Empirical Studies on the Development of Executable Business Processes*, pp. 113–142. Springer (2019)
54. Seiger, R., Zerbato, F., Burattin, A., García-Bañuelos, L., Weber, B.: Towards iot-driven process event log generation for conformance checking in smart factories. In: *IEEE 24th International Enterprise Distributed Object Computing Workshop (EDOCW)*, pp. 20–26. IEEE (2020)
55. Tomlein, M., Grønbaek, K.: A visual programming approach based on domain ontologies for configuring industrial iot installations. In: *Proceedings of the 7th International Conference on the Internet of Things*, p. 9. ACM (2017)

56. Tomlein, M., Boovaraghavan, S., Agarwal, Y., Dey, A.K.: Chariot: An end-user programming environment for the iot. In: *Proceedings of the 7th International Conference on the Internet of Things*, p. 25. ACM (2017)
57. Schönig, S., Ackermann, L., Jablonski, S., Ermer, A.: An integrated architecture for iot-aware business process execution. In: *Enterprise, Business-Process and Information Systems Modeling*, pp. 19–34. Springer (2018)
58. Jacobs, T.: A reference architecture for IoT-enhanced business processes (2019)
59. Meroni, G., Baresi, L., Montali, M., Plebani, P.: Multi-party business process compliance monitoring through IoT-enabled artifacts. *Inf. Syst.* **73**, 61–78 (2018)
60. Baumgraß, A., Botezatu, M., Di Ciccio, C., Dijkman, R., Grefen, P., Hewelt, M., Mendling, J., Meyer, A., Pourmirza, S., Völzer, H.: Towards a methodology for the engineering of event-driven process applications. In: *International Conference on Business Process Management*, pp. 501–514. Springer (2016)
61. Marrella, A., Mecella, M., Sardina, S.: Supporting adaptiveness of cyber-physical processes through action-based formalisms. *AI Commun.* **31**(1), 47–74 (2018)
62. Seiger, R., Huber, S., Heisig, P., Aßmann, U.: Toward a framework for self-adaptive workflows in cyber-physical systems. *Softw. Syst. Model.* **18**(2), 1117–1134 (2019)
63. Appel, S., Kleber, P., Frischbier, S., Freudenreich, T., Buchmann, A.: Modeling and execution of event stream processing in business processes. *Inf. Syst.* **46**, 140–156 (2014)
64. Walch, M., Karagiannis, D.: Service-driven enrichment for kbr in the omilab environment. In: *International Conference on Service-ology*, pp. 164–177. Springer (2017)
65. Peng, T., Ronchetti, M., Stevovic, J., Chiasera, A., Armellini, G.: Business process assignment and execution from cloud to mobile. In: *International Conference on Business Process Management*, pp. 264–276. Springer (2013)
66. Seiger, R., Huber, S., Schlegel, T.: Toward an execution system for self-healing workflows in cyber-physical systems. In: *Software and Systems Modeling*, pp. 1–22 (2016)
67. Kammerer, K., Pryss, R., Sommer, K., Reichert, M.: Towards context-aware process guidance in cyber-physical systems with augmented reality. In: *4th International Workshop on Requirements Engineering for Self-Adaptive, Collaborative, and Cyber Physical Systems (RESACS)*, pp. 44–51. IEEE (2018)
68. Wang, X., Ong, S.K., Nee, A.Y.: A comprehensive survey of augmented reality assembly research. *Adv. Manuf.* **4**(1), 1–22 (2016)
69. Korzetz, M., Kühn, R., Gohlke, M., Aßmann, U.: Holofacility: get in touch with machines at trade fairs using holograms. In: *Proceedings of the 2017 ACM International Conference on Interactive Surfaces and Spaces*, pp. 336–341. ACM (2017)
70. Tepper, O.M., Rudy, H.L., Lefkowitz, A., Weimer, K.A., Marks, S.M., Stern, C.S., Garfein, E.S.: Mixed reality with hololens: where virtual reality meets augmented reality in the operating room. *Plast. Reconstr. Surg.* **140**(5), 1066–1070 (2017)
71. Ríos, A.P., Callaghan, V., Gardner, M., Alhaddad, M.J.: Using mixed-reality to develop smart environments. In: *International Conference on Intelligent Environments*, pp. 182–189. IEEE (2014)
72. Andrade, T., Bastos, D.: Extended reality in iot scenarios: Concepts, applications and future trends. In: *5th Experiment International Conference (exp. at'19)*, pp. 107–112. IEEE (2019)
73. Mayer, S., Inhelder, N., Verborgh, R., Van de Walle, R.: User-friendly configuration of smart environments. In: *12th IEEE International Conference on Pervasive Computing and Communication (PERCOM)*, pp. 163–165. IEEE (2014)
74. Poppe, E., Brown, R.A., Recker, J.C., Johnson, D.M.: A prototype augmented reality collaborative process modelling tool (2011)
75. Oppl, S., Stary, C.: Facilitating shared understanding of work situations using a tangible tabletop interface. *Behav. Inf. Technol.* **33**(6), 619–635 (2014)
76. Nolte, A., Brown, R., Poppe, E., Anslow, C.: Towards collaborative modelling of business processes on large interactive touch display walls. In: *Proceedings of the International Conference on Interactive Tabletops & Surfaces*, pp. 379–384. ACM (2015)
77. Poppe, E., Brown, R., Recker, J., Johnson, D., Vanderfeesten, I.: Design and evaluation of virtual environments mechanisms to support remote collaboration on complex process diagrams. *Inf. Syst.* **66**, 59–81 (2017)
78. Pryss, R., Reichert, M., Bachmeier, A., Albach, J.: Bpm to go: supporting business processes in a mobile and sensing world (2015)
79. Seiger, R., Kühnert, A., Aßmann, U.: Workflow-based setup of smart devices in mixed reality. In: *Proceedings of the 9th International Conference on the Internet of Things*, pp. 35:1–35:4. IoT 2019, ACM, New York, NY (2019)
80. Kawsar, F., Nakajima, T., Fujinami, K.: Deploy spontaneously: supporting end-users in building and enhancing a smart home. In: *Proceedings of the 10th International Conference on Ubiquitous Computing*, pp. 282–291. ACM (2008)
81. Giner, P., Cetina, C., Fons, J., Pelechano, V.: Developing mobile workflow support in the internet of things. *IEEE Pervas. Comput.* **9**(2), 18–26 (2010)
82. Tüysüz, G., Avenoglu, B., Eren, P.E.: A workflow-based mobile guidance framework for managing personal activities. In: *7th International Conference on Next Generation Mobile Apps, Services and Technologies*, pp. 13–18. IEEE (2013)
83. Seiger, R., Lemme, D., Struwe, S., Schlegel, T.: An interactive mobile control center for cyber-physical systems. In: *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct*, pp. 193–196 (2016)
84. Yousfi, A., Bauer, C., Saidi, R., Dey, A.K.: UBPMN: A BPMN extension for modeling ubiquitous business processes. *Inf. Softw. Technol.* **74**, 55–68 (2016)
85. Yousfi, A., Hewelt, M., Bauer, C., Weske, M.: Towards UBPMN-based patterns for modeling ubiquitous business processes. *IEEE Trans. Ind. Inf.* (2017)
86. Tomlein, M., Grønbæk, K.: Augmented reality supported modeling of industrial systems to infer software configuration. In: *Proceedings of the ACM on Human-Computer Interaction 2(EICS)*, 5 (2018)
87. White, G., Cabrera, C., Palade, A., Clarke, S.: Augmented reality in iot. In: *International Conference on Service-Oriented Computing*, pp. 149–160. Springer (2018)
88. Jo, D., Kim, G.J.: Ariot: scalable augmented reality framework for interacting with internet of things appliances everywhere. *IEEE Trans. Consum. Electron.* **62**(3), 334–340 (2016)
89. Seiger, R., Korzetz, M., Gohlke, M., Aßmann, U.: Mixed reality cyber-physical systems control and workflow composition. In: *Proceedings of the 16th International Conference on Mobile and Ubiquitous Multimedia*, pp. 495–500. ACM (2017)
90. Seiger, R., Keller, C., Niebling, F., Schlegel, T.: Modelling complex and flexible processes for smart cyber-physical environments. *J. Comput. Sci.* **10**, 137–148 (2015)
91. Widom, J., Ceri, S.: Active database systems: Triggers and rules for advanced database processing. Morgan Kaufmann, Burlington (1996)
92. Martinez, W.L.: Graphical user interfaces. *Wiley Interdiscip. Rev. Comput. Stat.* **3**(2), 119–133 (2011)
93. Craft, B., Cairns, P.: Beyond guidelines: what can we learn from the visual information seeking mantra? In: *9th International Conference on Information Visualisation (IV'05)*, pp. 110–118. IEEE (2005)
94. Lakoff, G.P., Johnson, M.: *Metaphors We Live By*. University of Chicago Press, Chicago (2008)

95. Norman, D.: The design of everyday things: Revised and expanded edition. Basic books (2013)
96. Huber, S., Seiger, R., Kühnert, A., Theodorou, V., Schlegel, T.: Goal-based semantic queries for dynamic processes in the internet of things. *Int. J. Semant. Comput.* **10**(02), 269–293 (2016)
97. Hart, S.G., Staveland, L.E.: Development of nasa-tlx (task load index): Results of empirical and theoretical research. In: *Advances in psychology*, vol. 52, pp. 139–183. Elsevier (1988)
98. Davis, F.D.: A technology acceptance model for empirically testing new end-user information systems: Theory and results. Ph.D. thesis, Massachusetts Institute of Technology (1985)
99. Lewis, J.R.: The system usability scale: past, present, and future. *Int. J. Hum.-Comput. Interact.* **34**(7), 577–590 (2018)
100. Nielsen, J.: Why you only need to test with 5 users (2000)
101. Sauter, T.: Integration aspects in automation-a technology survey. In: *IEEE Conference on Emerging Technologies and Factory Automation*, vol. 2, pp. 9. IEEE (2005)
102. Milgram, P., Takemura, H., Utsumi, A., Kishino, F.: Augmented reality: A class of displays on the reality-virtuality continuum. In: *Telemanipulator and Telepresence Technologies*, vol. 2351, pp. 282–293. International Society for Optics & Photonics (1995)
103. Shneiderman, B.: The eyes have it: A task by data type taxonomy for information visualizations. In: *Proceedings 1996 IEEE Symposium on Visual Languages*, pp. 336–343. IEEE (1996)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Ronny Seiger received his diploma in computer science from Technische Universität Dresden, Germany, in 2011 and his PhD in 2018. He has been a member of the Software Engineering of Ubiquitous Systems group from 2012 to 2015. From 2015 to 2019, he has been a research assistant at the Software Technology group at TU Dresden. Since 2019, Ronny Seiger is working as a postdoc at the Chair for Software Systems Programming and Development at the University of St. Gallen, Switzerland.

His PhD thesis was about self-managed workflows for cyber-physical systems. His research interests include workflows and processes, cyber-physical systems, Internet of Things, robotics, distributed systems, complex event processing and software engineering. His main focus is on applying workflow and process technologies for automating processes in cyber-physical systems.



Romina Kühn received her diploma in media computer science from Technische Universität Dresden, Germany. From 2011 to 2016, she was research associate at the Software Engineering of Ubiquitous Systems group at TU Dresden. Currently, she is a research associate at the of Software Technology group at TU Dresden. In her research, she focuses on CSCW, human-computer interaction and user experience specifically in the context of mobile and ubiquitous devices in collaborative work.



Mandy Korzetz is a research associate at the Software Technology group at Technische Universität Dresden, Germany. She received her diploma in media computer science in 2013 for investigating full-body gestures for elderly users. From 2013 to 2016, she was a research associate at the Software Engineering of Ubiquitous Systems group at TU Dresden and was involved in research on interactive mobile solutions for the public transport of the future. Her research interests include human-centred user experience engineering with a main focus on mobile and ubiquitous interaction and interface design.



Uwe Aßmann is head of the Software Technology Group of the Technische Universität Dresden. Uwe Aßmann got his PhD in 1995 at the Universität Karlsruhe on the generation of program optimizers from graph rewrite systems. Besides invasive model- and software composition, Uwe Aßmann works in the areas of self-adaptive architectures and software systems. He is a member of the IFIP Working Group 2.4 (System Implementation Languages) and member of the Gesellschaft für Informatik (GI).

He is member of the scientific advisor board for Silicon Saxony in the area of “Software”, one of the founding members of the “Cyber-Physical Systems Working Group” of Silicon Saxony, member of the working group “Microelectronics” of the state of Saxony. Uwe Aßmann is also reviewer of the European Union for “Complex Systems”.