End-User Development in Industrial Contexts: The Paper Mill Case Study

Marco Manca[0000-0003-1029-9934], Fabio Paternò[0000-0001-8355-6909], and Carmen Santoro[0000-0002-0556-7538]

CNR-ISTI, HIIS Laboratory, Pisa, Italy
{marco.manca,fabio.paterno,carmen.santoro}@isti.cnr.it

Abstract. This work aims to explore the potentialities of a personalisation platform in industrial settings. In such a context, stakeholders with different roles and competencies collaborate to manage and control an environment where legacy machines coexist and interact with newer ones. Our goal is to provide a rule-based tool that allows end-users to build personalised solutions to respond quickly to the dynamic needs of factories. We report on a case study in the paper factory domain, in which the industrial aspects identified with expert stakeholders through interviews have been simulated and addressed through an extension of a personalisation platform. A first user test of the resulting environment has been carried out with a representative set of users, and has provided useful and encouraging feedback in terms of the potentialities of the proposed approach in industrial contexts.

Keywords: End User Development, Internet of Things, Industry 4.0.

1 Introduction

Today's industrial environments are becoming highly dynamic, with shorter product life cycles and delivery times, requiring increased levels of innovation and customisation. Such requirements call for rapidly responding systems that can adjust to required changes in processing functions and production, and thereby meet customisation demands on a timely basis. Industry 4.0 (Yong et al., 2018) is the current response to these complex scenarios: by connecting and combining different technologies and software, it aims to enable seamless and flexible production, thus realising the power of digitalisation in industrial plants. In particular, thanks to the Internet of Things (IoT), a key enabling technology of Industry 4.0, the way in which operations and processes are carried out is changing radically. What was 'closed' inside factories, stored in different local 'data silos' (i.e. one for each machinery producer), and managed using devices based on proprietary/non-standard communication protocols (which kept them rather isolated and inflexible), now is increasingly handled through more standard approaches, promoting easy connectivity and interoperability between the devices, sensors, and actuators available in firms. This will offer unprecedented access to real-time data on products and processes, and enable more informed decisions across the whole enterprise (from technicians to front-line operators and top managers), potentially leading to continuous factory optimisation.

In the manufacturing sector, while the availability of up-to-date information at all levels (i.e. from technical processes, to individual equipment components, to associated production and business processes) for better factory control is becoming paramount (Wieland et al., 2017), turning this vision into reality is extremely challenging. This is not only because in these contexts there is a plethora of processes, IoT assets, information sources and up-to-date as well as legacy machines to manage, but also because the integration, maintenance, and control of software is usually the responsibility of Information Technology (IT) experts. Therefore, when manufacturing workers identify that a change is needed to the software controlling some processes (e.g. because in a specific situation a different behaviour is needed), they strongly depend on the IT department to implement it. However, current software development cycles are not always able to respond quickly to the dynamic needs of factories, a situation that could introduce significant delays and increase costs. Thus, it is becoming clear that applications whose behaviour depends on context cannot be completely "hard-coded" at design time by professional developers, since they cannot predict all the possible situations of use, or whether the results produced will actually be meaningful, as they often lack the knowledge that usually only domain experts have. This scenario seems a suitable application area for End-User Development (EUD), which aims to provide domain experts with effective tools to build solutions to the problems they face every day, by empowering them to develop and iterate autonomously needed customisations without including IT experts at each stage.

In the context of IoT-based applications, EUD approaches that exploit the trigger-action paradigm have demonstrated particularly promising potential (Manca et al., 2021a; Bellucci et al., 2019), thanks to their compact and intuitive structure which directly links dynamic events or conditions of the current context to actions to be executed when the rule is triggered. In this regard, several pieces of work have been presented to support the creation of trigger-action rules, even using different composition paradigms (Gallo et al., 2021; Asunis et al., 2021). Several applications from the academic and industrial fields have shown that the trigger-action paradigm could be easily understood also by people without specific programming skills (Ur et al., 2014), since its use does not require specific algorithmic knowledge, or abilities in the use of complex programming structures: users have just to specify the rules that indicate the desired effects (i.e. in terms of changes to the state of devices, appliances, and user interfaces) when specific situations occur. Such approaches have been applied to different domains ranging from rehabilitation (Tetteroo et al., 2015), robotics (Weintrop et al., 2017), smart homes (Caivano et al., 2018), Ambient Assisted Living (Jaschinski et al., 2019), and finance (Elsden et al., 2019). However, to the best of our knowledge, so far industrial contexts have been considered only in a limited manner with regard to EUD themes. This is especially interesting taking into account that this domain involves radically different factors for EUD research. For instance, in industrial scenarios the personalisation goals would be typically geared towards achieving specific objectives of efficiency, optimisation, cost-reduction and safety (while in other domains the goals can be different, or even less well-defined).

Moreover, while end users in industrial scenarios could still be unskilled in programming, their motivation to use personalisation tools could be higher than in other domains since workers would need them to optimise their own day-by-day work or to increase the workplace safety. In addition, in industrial settings a variety of different stakeholders, with precise roles and competencies collaborate and can intervene in managing (and personalising) the behaviour of the devices in the factory according to well-defined, strict protocols and procedures. Preliminary ideas in this regard have been provided in previous work (Manca et al., 2021b), even though to a limited extent.

In this paper, considering the increasingly emergent trend of Industry 4.0, we focus our attention on applying an EUD trigger-action approach to an industrial scenario in the paper sector, to investigate to what extent the concepts associated with this approach can be suitable for addressing current issues in such Industry 4.0 scenarios, and easily exploited by domain experts for personalising the behaviour of factory equipment according to events and situations occurring in it. The contribution of this work is to show how a solution based on trigger-action rules can be used to make such personalisation easier for people who are not professional software developers. In order to show this, we extended an EUD platform in order to support triggers and actions relevant in an industrial context, and then we gauged the solution through a remote usability study in which real experts in the considered sector had to specify pertinent rules. We also provided participants with the possibility to see the effects of the interactions with the EUD tool, by executing some rules using simulations.

The structure of the paper is as follows. In the next section we discuss related research, then in Section 3 we describe the case study considered, and in Section 4 we report on some interviews with stakeholders, which have been carried out to identify the end-users needs, and then we present the collected requirements. In Section 5 we detail the solution to support experts of this domain to personalise their applications. In Section 6 we describe a user study that we carried out involving relevant stakeholders in the paper sector, also providing a discussion of the main results gathered. Then we conclude, and describe our future plans in this area.

2 Related Work

According to (Barricelli et al., 2019), the application domain of business and data management is one of the most frequent in which End User Development or End User Programming techniques have been applied (24% of total). This is also because it was the historical domain where the idea of tailoring digital artefacts by end users at use time was born, by exploiting spreadsheet programming (Bricklin et al., 1979). Well before the advent of the Industry 4.0 initiative, component-based *tailorability* (Stevens et al., 2006) was proposed within the context of an industrial case study to enable users to match computer systems to the specific application context considered, more specifically that technical flexibility can be achieved by allowing end users to recompose components at runtime. For this purpose, the authors emphasise that the system should have already been appropriately broken down into modules at design time, in a way that it provides sufficient flexibility with respect to the application context and is understandable to end users.

A recent contribution (Modesto et al., 2021) describes a systematic literature mapping study analysing the main EUD strategies used by organisations, as well as the benefits of and barriers to their adoption. The benefits they identify can be classified into human and organisational factors, whereas the barriers are related to people, processes and technologies. In particular, on the one hand, support for decision-making, reduced dependence on IT, increased end-user productivity and increased end-user satisfaction are the most mentioned benefits of EUD adoption. On the other hand, lack of training and support for end users, and the need for technological support were the most cited barriers.

In the business domain, a common approach is to use workflow-based technologies to define and execute business processes, with established standards being BPEL and BPMN. However, such approaches focus just on business processes, whereas Industry 4.0 settings are typically more complex, as they involve a variety of heterogeneous physical IoT devices, digital resources, services, and activities, which can change based on events occurring on them or in the operator's context. As an attempt to bridge the gap between physical IoT devices and business processes, Friedow et al. (2018) suggested employing process models to define the process layer of IoT applications, and enact them through a process engine. However, while workflow-based approaches facilitate the integration of different systems, they require quite strong programming skills, therefore being unsuitable for unprofessional developers.

A key component of Industry 4.0 is its human-centricity, which Romero et al. (2016) concretised in the *Operator 4.0* concept. It refers to smart and skilled operators of the future, who will be assisted by automated systems providing sustainable relief to their physical and mental stress, and enabling them to better leverage their creative skills without compromising production objectives. The authors propose an Operator 4.0 categorisation, arguing that one operator could incorporate one or several others, differentiated between: Super-Strength Operator (e.g. using exoskeletons), Augmented Operator (e.g. using augmented reality tools), Virtual Operator (using a virtual factory), Healthy Operator (e.g. using wearable devices to track well-being), Smarter Operator (e.g. using agent or artificial intelligence for planning activities), Collaborative Operator (e.g. interacting with cobots), Social Operator (sharing knowledge using a social network) and Analytical Operator (using Big Data analytics).

Fogli and Piccinno (2019) highlight that there is a gap between what Industry 4.0 promises, and how Operators 4.0 will be called on to change their work practices, suggesting that the integration of EUD with Industry 4.0 enabling technologies might help workers to evolve more smoothly into the various types of Operator 4.0. For instance, the Super-Strength Operator can be included in the Augmented Operator by assuming exoskeletons as a form of augmentation that must be (physically) personalised to the user, while Smarter, Healthy and Social Operators can be embraced in the IoT Operator, who, through EUD, should be able to manage the entire IoT ecosystem. Thus, the enabling technologies of Industry 4.0 should be tailored to the work context and the type of operator *by users themselves*, supported by suitable EUD tools developed according to meta-design (Fischer and Giaccardi, 2006), as it not only focuses on technologies, but can also sustain the cultural transformation needed to address the future complexity of workplaces.

Other examples that focused on EUD applied to industrial environments involve robotics, which in recent years inspired several contributions in the EUD area (see e.g. (Ajaykumar et al., 2022) and (Coronado et al., 2020) for two literature reviews). While robots have not yet become commonplace in homes, collaborative robots work with humans in factories with increasing frequency. Robot Blockly (Weintrop et al., 2017) is a block-based programming environment for a single-armed industrial robot, and first user tests indicate that novices with no prior programming experience can use it to successfully write programs to accomplish basic robotics tasks. However, as acknowledged by the authors themselves, it was just a preliminary investigation into the potential of block-based programming for industrial robots. Fogli et al. (2022) propose an approach to collaborative robots for non-technical users, through the development of a prototype programming environment called CAPIRCI, which can be tailored to different application domains through the definition of objects, locations, and actions. Two experimental tests have been carried out on CAPIRCI using the COBOTTA industrial robot: one to see whether the integration of natural language chat and block-based interaction make programming easier for non-technical users than relying on block-based interaction only; another one to understand how usable this environment is. The results obtained show that this approach exploiting both natural language dialogue and block-based interaction can help make the programming task easy and efficient for non-technical users. Even though the robot was a collaborative, non-humanoid robot aimed at fulfilling the typical repetitive tasks that can be found in industrial settings, the test did not involve the real target users, but a variety of non-technical participants with diverse knowledge/backgrounds (i.e. school teachers, managers, housewives, clerks, nurses, farmers, factory workers, unemployed people).

Ong et al. (2020) discuss an Augmented Reality -assisted Robot Programming System (ARRPS) designed to allow users with little robot programming knowledge to program tasks for an industrial robot, by transforming the work cell of a serial industrial robot into an AR environment. A prototype of AARPS has been implemented and applied to two applications, namely, welding and pick-and-place operations. For each application, a user study was conducted with ten participants who were tasked with programming the robot to perform certain operations. None of the participants had prior experience in robot programming. The results show that the system could significantly speed up the programming of robotic tasks, and reduce the need for user expertise in robot programming. However, also in this case real workers were not involved in the evaluation.

Senft et al. (2021) present “situated live programming” for Human-Robot Collaboration (HRC), an approach that enables users with limited programming experience to program collaborative applications for human-robot interaction. Allowing end users, such as shop floor workers, to program collaborative robots themselves would make it easy to “retask” robots from one process to another, facilitating their adoption by small and medium enterprises. The approach builds on the trigger-action programming paradigm in order to empower end users to create rich and personalised automations. It enables end users to iteratively create, edit, and refine a reactive robot program while executing partial programs: users can create trigger-action programs by annotating an augmented video feed from the robot’s perspective and assign robot actions to trigger conditions. The system was evaluated in a study where ten participants developed robot programs for solving collaborative light-manufacturing tasks, and results showed that users with little programming experience were able to program HRC tasks in an interactive manner. However, once again, the study participants were not drawn from domain experts such as factory workers, but from a student population.

Recent commercial automation platforms such as IFTTT or Zapier also allow users to integrate different IT systems in an easy and flexible manner, without having programming skills. However, they typically allow users to define rather simple rules, and have so far only been considered for integration in business scenarios (e.g. Zapier integrated with Customer Relationship Management systems).

To date, in sectors such as the manufacturing industry, tailoring issues have been addressed only limitedly as well. In this regard, Wieland et al. (2016) propose MIALinx, a lightweight and easy-to-use integration solution for SMEs using *if-then* rules that connect situations occurring in manufacturing environments (e.g. machine breakdowns) with corresponding actions (e.g. an automatic maintenance call generation). To this goal, MIALinx connects sensors and actuators according to rules defined in a domain-specific and easy manner, to enable rule modelling by domain experts. In their approach, rules involve available sensors and actuators in the current production environment, and they are then transformed to be managed and executed using existing rules engine (e.g. Jess or Drools). In a more recent paper (Lucke et al., 2019) the user interface of MIALinx has been presented. It was installed and tested in an industrial plant and in a lab dedicated to research on future working places. The first test results show that it usually takes less than 30 seconds to create a rule after a short introduction (less than 5 minutes). However, no further details on these tests are provided to fully appreciate the validity of their solution.

Another recent trend involves approaches that leverage AI-based techniques to support different stakeholders in achieving the personalisation needed. For instance, relevant work has been proposed, using different approaches, by (Mattioli and Paternò, 2021), (Corno et al., 2020/2021) and (Zhang et al., 2020) to the aim of helping democratise IoT device programming for non-technical end users by providing them with suitable *recommendations* to meet their individual contextual needs. In particular, Zhang et al. (2020) introduce and evaluate Trace2TAP, a method for automatically synthesising TAP rules from traces (time-stamped logs of sensor readings and manual actuations of devices). An additional line of research is the one that leverages AI-based techniques to enable users to better *understand* and *predict* the future behaviour of a system (Coppers et al., 2020). As acknowledged by their authors, this approach can be particularly useful for debugging unintended behaviour (by providing suitable explanations about what the system will do in the future and why), and also for managing possible conflicts, which are two challenges that can be particularly relevant in Industry

4.0 scenarios. However, none of these works has been applied to the Industry 4.0 domain.

To sum up, by analysing the state of the art there is a lack of solutions that apply EUD approaches in an Industry 4.0 context (such as the manufacturing sector of a paper mill in our case), also gathering feedback from real stakeholders. This work aims to contribute to filling this gap, also aiming at understanding how this approach was received by real stakeholders and to what extent the rule-based metaphor exploited can be concretely used by them.

3 The Case Study Considered

The case study focuses on a paper mill. Paper production is basically a process in which a fibrous raw material is first converted into pulp, which is then converted into paper. To this aim, wood chips are first processed so that the unusable part of wood (i.e. lignin) is separated from useful fibres (i.e. cellulose), which are broken up using water within one of the machineries in such factories, which are 'pulpers' to produce pulp, the main ingredient of paper. The characteristics of the various pieces of equipment available in a paper mill should be properly monitored, to understand whether they are working properly and efficiently. The 'pulp' produced by the pulpers then feeds a continuous "paper machine", together with the other ingredients that define the "recipe" used to deliver a specific product (e.g. 'paper' is distinguished from 'carton board' since it has a lower basis weight or 'grammage'). Paper machines represent the core of the papermaking production process: they are endlessly moving belts that receive a mixture of pulp and water and drain excess water off (by suction, pressure, or heat desiccation). The continuous paper sheet (called 'web') coming out of the paper machine is wound onto an individual spool, to become a 'parent reel' (or 'jumbo roll', see Fig. 1). Since the reel width is fixed for each paper machine, next, another machine (a 'winder') cuts the reel into rolls of smaller diameter, minimizing as much as possible trim losses. For cut-sheet paper products, rolls are loaded onto a 'sheeter', which unwinds them and slices the paper into sheets of the desired size, which are then wrapped and loaded onto vehicles for shipment to customers/distribution centres. While jumbo rolls are the main output of paper mills (and whose quality is characterised by various parameters), they, in turn, represent the input of paper converting companies, which transform them into e.g. napkins, envelopes, tissue. In this sector, on the one hand, factory managers have the strategic objective of always having available the needed materials and the capability to produce without interruption, while ensuring/maintaining a proper cost/revenue ratio. In this regard, it is worth noting that materials, parts, as well as products delivered are stored in warehouses that need to be properly managed as well (i.e. avoid shortage of inventory items, avoid stocking too much raw material to reduce holding costs, maintain warehouses sufficiently empty to store finished products).

At the same time, managers should also reduce to a minimum the costs (i.e. those associated to the waste of the material used), also ensuring that the work in the factory is done according to specific protocols and procedures (also considering safety), and providing support when emergency situations occur, such as the well-known "man-down" situations. On the other hand, in order to properly customise the behaviour of such factories, the responsible persons in charge must be able to issue suitable actions on the available equipment (i.e. the production lines) as well as to send alarms, notifications, and warnings to concerned people via proper audio/visual channels in specific situations.

Figure 1: A paper machine with jumbo rolls in the foreground (from VOITH)

4 Domain Analysis and Requirements Elicitation through Interviews

To elicit the requirements for an EUD approach to supporting better control of a paper mill by the people working in it, we first performed a domain analysis. The EUD platform targets operators and department managers who need to supervise the activities that are carried out within the factory, and the complex systems that these activities control.

We interviewed stakeholders of the paper sector preferably having a managerial view (i.e. responsible for departments) to identify relevant requirements, to better understand current practices and challenges, and also to uncover events and actions for customisation rules relevant in this domain. Such stakeholders were recruited from the network of the members of a regional project. Initially contacted by phone, they received via email a brief introduction about the personalisation approach that follows the trigger-action paradigm and a document on personal data processing and informed consent to fill in and sign. Interviews were remotely conducted to gather information on: *i) The stakeholders* (age, gender, familiarity with technology, experience in the sector, current role they play in the company, and associated tasks) and their *companies* (goals, size);

ii) *Adoption of IoT/Industry 4.0 and currently used methods within their company; iii) Relevant events/sensors*; *iv) Relevant actions/actuators*; *v) Challenges* (e.g. aspects that pose problems, situations to improve). We involved 5 subjects (1 woman; AVG age=51.2; SD=3.8; Min=45; Max=55), overall quite familiar with technology, and working in companies all located in the Lucca area, one of the largest districts in this sector.

Stakeholders. One stakeholder (Stakeholder1, M, 52) is responsible of the IT department for a company (300+ workers in the XXX area) that builds undulators (the machines producing undulated cardboard, typically used for packaging). Another one (Stakeholder2, M, 53) works in a paper mill (200+ workers in the XXX area) producing undulated cardboard: he has 37+ years of experience in this sector, currently managing safety. Another stakeholder (Stakeholder3, F, 45) is the IT director of a paper mill (200+ workers only in the XXX area). Another one (Stakeholder4, M, 51) is the administrator of a small transport company (~40 people): his activities range from managing warehouses, customers, to administration and even safety. The last one (Stakeholder5, M, 55) is the General Director of a paper converting company (~65 workers in the XXX area).

Adoption of IoT/Industry 4.0 technologies.

Stakeholder1. The firm producing undulators, where Stakeholder1 works, is quite technologically advanced: for instance, they already use a predictive maintenance system for increasing the lifespan of their equipment and avoiding disruptions to operations. In addition, the IT team (led by Stakeholder1) of this factory developed an application providing the company with real-time data about the equipment they produce, and their customers with various reports about the equipment they use, also allowing them to modify autonomously specific parameters according to their needs (by acting on a database), without the need for the manufacturer's intervention.

Stakeholder2. The paper mill producing undulated cardboard is technologically heterogeneous: modern and legacy equipment coexist, with several costly machines, difficult to replace. Also, they do not strongly leverage Industry 4.0 technologies yet. Even the idea of using web TVs (already available in some plants), to send messages to operators, was hindered by the management, for security reasons. A situation that could be improved regards checking the quality of the cardboard they produce: while workers can manually operate on relevant actuators (i.e. cylinders, pistons and electro valves) to modify some characteristics of the product that is being delivered, some operations could be automatised by considering relevant parameters (i.e. the humidity of the paper or the amount of glue contained in it).

Stakeholder3. The other stakeholder working in a paper mill also reported that Industry 4.0 adoption is still at an early stage in their factory: even though they already use many sensors, they would be eager to have further support such as predictive maintenance or self-correcting equipment. In addition, in the same plant, they can have both 'old' machines (dating to the '80s, on which they use sensors to "retrofit" them), and newer ones measuring parameters such as paper humidity, strength, and grammage (a measure of paper 'thickness', used to define different paper types). Concerning reminders, alarms, warnings, and notifications, currently there are already notifications that are sent to the operator in some situations, for example when the camera detects some issue in paper cutting. The notifications are already made through acoustic and light signals, or through some fixed digital displays placed, however, such notifications are not customisable.

Stakeholder4. In the transport company, they recently purchased a trolley that automatically updates the warehouse's inventories by "firing" barcodes on items. To move goods, they also have elevator carts that automatically register entry/exit via barcodes. To monitor the situation in the factory, the interviewee declared that they use e-mail, or in-person verification, going directly to the warehouse. The stakeholder reported that they would like to have sensors to detect risky situations (e.g. when ground personnel is not properly distanced from forklifts), which currently are not addressed.

Stakeholder5. The paper converting company just started adopting Industry 4.0 recently. Indeed, one of their newer lines has its composing machines (i.e. winders, cutters) connected to the management system, thus, operators can get data in real-time, and they are also installing a predictive maintenance system.

Events. In the company building undulators, relevant events include those related to monitoring, i.e. the quality of the produced equipment (situations in which the final product is not up to the expected quality standard) and the production speed. In one paper mill, situations to detect include the characteristics of the produced board (i.e. cameras, weighing scales, and sensors are already used to control the amount of starch, glue, humidity), anomalies (e.g. unglued sheets), number of produced items, real-time indication of incoming and outgoing materials, detection of truckers' arrivals to provide them with information on how to behave inside the warehouse, and some characteristics of the equipment (i.e. temperature). In the other paper mill, relevant events include the consumption of raw material (in terms of e.g. water, steam, starch) and also its quality (i.e. humidity, amount of ashes or plastic contained in it). The situations to monitor reported by the stakeholder working in the transport company include controlling cost/revenue ratio (e.g. they would like to suitably handle more up-to-date information, whereas now reports are sent every three months), and whether activities are carried out in accordance with safety regulations. In the paper converting machine company, relevant aspects include those related to machinery (i.e. state, production speed, temperature), paper grammage, number of tears in paper rolls, roll length, and diameter. The main challenge in this company is to equip the machines with greater quality control: in fact, quality control is typically done on a sample basis, thus often done long after the issue occurs (because the machines work in a continuous cycle); by quality, they mean the quality of the packaging in terms of the absence (or limited presence) of defects.

Actions. In the company producing undulators, actions include those that operate on components of the equipment (i.e. cylinders, pistons, valves, servomotors). Alarms or notifications are sent in case of anomalies, or when the equipment is working poorly. In paper mills, notifications are sent to users in case of anomaly via sounds or lights, or using monitors on the lines. Also, one of the most serious alarms is issued when a machine stops, while warnings occur e.g. when the "recipe" currently used (i.e. the mixture of ingredients used to produce a particular product) is going to change. Another action needed concerns the possibility of providing general communication to employees in a more pervasive and effective way (for example using some displays). In the paper transport company, audible or visual alarms are already sent in "man-down" situations (by using dedicated devices), or to personnel on moving carts, who may not see their surroundings well; however, the stakeholder points out that it is important to limit such alarms to truly risky situations. This was also confirmed by the stakeholder working in the paper converting machine company, who reported that while acoustic and flashing signals are already used to highlight anomalies, often such alarms do not correspond to truly dangerous situations, thus in such cases it is necessary that human operators check them.

Current Challenges and Personalisation Scenarios. The stakeholder working in the firm producing equipment for papermaking highlights that, with automation increasingly introduced in industrial settings, there is the challenge of improving the satisfaction of operators, whose tasks nowadays are often reduced to rather passive roles (i.e. visual monitoring), as well as improving factory efficiency (i.e. increasing production while decreasing the need for maintenance stops). For paper mills, one challenge is to avoid paper breaking (depending on the contract, there is a maximum number of admitted tears in the same paper roll): when this situation occurs, they have to avoid both customer's 'downgrading' of the product (due to too many 'joints' in the same roll) and wasting material. In particular, in these situations, Stakeholder3 would like to set that e.g. dispensers (feeding the line with the ingredients) automatically stop, and specific warnings reach concerned people with associated reporting of the problem. She also would need more sensors on the lines, to improve checking *in-line* (i.e. in real-time), and not *off-line* (i.e. in laboratories), as it occurs today. In the paper converting machine company, one challenge is enhancing the quality control: currently the data about product defects or equipment efficiency come on a sample basis and at a later stage (since machines work in a continuous cycle), while they would need them continuously to enable suitably prompt reactions. The stakeholder working as a safety officer in a paper mill mentioned several scenarios that can benefit from personalisation: their undulators need to be configured based on dynamic plant factors (e.g. internal temperature, humidity); he also would like to get real-time info on lorry flow at the plant entrance to send tailored navigation info to concerned drivers.

Events-related Requirements	
R1	Monitoring the quality of the final product (for paper, it is in terms of glue, humidity, starch, grammage, number of tears in a paper roll, roll length and diameter)
R2	Monitoring possible anomalies in the final product (e.g. unglued sheets)
R3	Monitoring some parameters of the equipment (e.g. pulpers, elevators) such as temperature, status, anomalies
R4	Monitoring the production line (status, production speed)

R5	Monitoring raw material's quality (i.e. humidity, amount of contained ashes or plastic) and consumption (e.g. in terms of water, steam, starch)
R6	Monitoring the cost/revenue ratio
R7	Monitoring whether the activities are carried out in accordance with safety regulations (e.g. people operating on lifts at a safe distance from staff on the ground)
R8	Monitoring man-down situations
Action-related Requirements	
R9	Supporting actions that operate on parts of the equipment (e.g. production lines)
R10	Supporting audible or visual alarms and notifications (for managing serious situations e.g. when a machine stops)
R11	Supporting notifications via sounds, lights, or using the displays on the lines
R12	Supporting warnings (e.g. when the recipe currently used is going to change).

Table 1: Summary of Requirements revealed by interviews

To sum up, while the companies where the interviewees work are overall at an initial stage in adopting Industry 4.0/IoT for various apparent reasons (i.e. investments needed to replace machines/infrastructure, difficulties in managing IoT-related security issues), their managers seem well aware of the opportunities that these technologies could bring to them in terms of having an integrated, real-time view of the system to enable continuous optimisation. In this scenario, the proposed personalisation approach targeting non-software developers (like them) was judged particularly relevant and indeed concrete personalisation scenarios came up during the interviews, which also provided useful information for the design of our solution in the considered domain. Table 1 summarises the requirements that have been collected during such interviews.

5 The Architecture of the Solution

To address the requirements identified in the paper domain we extended an existing platform (Manca et al., 2021a) previously applied to other sectors (i.e. smart home, Ambient Assisted Living).

5.1 The Platform Architecture

The architecture of the solution is shown in Figure 2. The idea is that the applications used by workers to control and manage the paper factory (for monitoring the production, acting on parts of the production equipment, managing emergency situations within the plant, supporting data analysis and reporting) should be able to adapt their behaviour in a context-dependent manner, reacting to the events occurring in the surrounding context, and applying the actions specified in rules defined by 'end user developers', who, in our case, are mainly experts in the paper sector.

The Rule Editor is the EUD tool they can use to specify such behaviours, following a trigger-action paradigm. Once rules are created via the Rule Editor, those that the user wants to consider for actual execution in the current context are sent to a module called "Rule Manager", which subscribes to another module, the Context Manager, to be informed when relevant events occur in the current context. More specifically, the Context Manager consists of a Context Server receiving the context updates, and several Context Delegates, which are lightweight software applications able to communicate with sensors and appliances to receive info about their state, and consequently forward such information to the Context Server, by exploiting REST-based calls on HTTPS. In our solution, since it was problematic to perform the experimentation in a real industrial setting, two simulators have been developed to simulate the occurrence of events and actions, respectively (further details will be provided later on in this section). The Context Manager receives the data coming from sensors, and stores them in a uniform format used for all the devices, appliances and machines belonging to the considered context. The applications in turn subscribe to the Rule Manager to be informed when an action should be carried out. Whenever an event specified in a rule occurs, the Rule Manager receives a notification from the Context Manager, selects the actions associated with the triggered rule, and sends them to the subscribed applications. The applications have to interpret the received actions, then send via MQTT the associated commands to the devices, appliances, and actuators involved in the actions. This in some cases could also involve additional users, even with different roles (e.g. a message is sent to another factory worker).

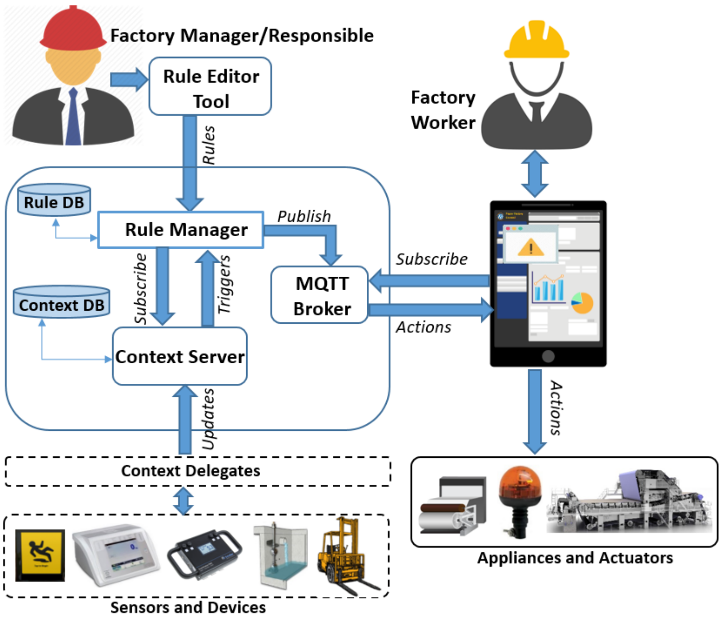

Figure 2: The Platform Architecture

As mentioned before, we also developed two simulator prototypes. The one dedicated to events simulates situations occurring on production lines (by using it, the user can monitor the state of the production lines and also change relevant parameters associated with its composing equipment, e.g. the weight of paper trim losses detected at the end of the production cycle and measured through some weight scales), and also the occurrence of emergency situations (such as the 'man-down' alarm, which in real scenarios is typically issued by dedicated devices that detect worker's falls). As for the actions, the corresponding simulator provides a view of the factory, which includes elements, such as coloured semaphores highlighting specific situations on production lines (e.g. a red semaphore indicates a situation that needs further attention).

As we describe in Section 6, such simulators were also used in the test to provide test participants with a (simulated) representation of the state of some relevant equipment that can typically be found in a paper-based factory, to make them able to see, in real-time (namely: during the test), the effect of the rules they created using the Tailoring Tool on such simulated elements. However, while on the one hand, we need to simulate triggers and actions specific to paper mills, the triggers and the actions that are not specific to paper mills (i.e. the triggers based on weather services, the actions sending alarms via email or SMS) were actually functioning and supported by the platform, and not just simulated, e.g. users actually received an alert email within the test, when the task required it (see Task 5.2).

5.2 How to Configure the Tailoring Platform for a specific Domain and Context

The Tailoring Platform is designed to be a generic, domain-independent environment that can be easily configured depending on the domain considered as well as the specific user's context. When a domain-dependent customisation needs to be done, domain experts (together with developers), have first to identify the domain-specific triggers and actions needed to populate the Rule Editor. This will involve the inclusion of the events, and the conditions sensed from the equipment and/or the services typically exploited in these environments (in the paper domain considered, they are those associated with e.g. production lines, weight scales), and the same will occur also for the actions. After doing this, the Rule Editor is tailored for the particular *sector* considered, as it exploits a specific vocabulary (in terms of triggers and actions) relevant for that domain.

However, since in real scenarios end users need a tool that allows them to control *real* objects in their *real* contexts of use (e.g., a specific paper factory) a further customisation step of the platform is needed to connect the rule editor elements identified in the first step to the specific instances of sensors, equipment, machinery that actually exist in the user's real context. In the presented solution this is done by Context Delegates, which are small softwares associated with each specific object/sensor to communicate with the Context Server. Such Context Deelegates register in the Context Server, so that then the Rule Editor can receive from the Context Server the updated description of the triggers currently available in the considered context.

In the presented platform, the description of the types of the various contextual entities is represented in a context meta-model specified in an XSD file. When the Context Server is compiled, this XSD file is automatically translated into a set of Java classes.

Then, during the context initialisation phase, various instances of Java objects are created accordingly, to define the state of the elements composing the current context (e.g. the various instances of users, environment and technologies). To update such Java objects, the Context Server provides a RESTful service to receive the data from the various Context Delegates which, using this service, will dynamically update the various attributes of the entities composing the current context. Still at initialisation time, the Rule Editor also asks the concerned application(s) about the specific actions to make available in the part of the tool's UI dedicated to actions.

It is worth noting that when a new sensor needs to be dynamically added (i.e. to model an additional property of an equipment), only the XSD file needs to be changed, while the RESTful service mentioned before can still be used to change the value of the newly added attribute. This allows for managing in a flexible manner possible evolutions of the platform in terms of equipment and sensors included in the current context.

In the Rule Editor, there is a specific panel in which some configuration parameters can be specified and are used to customise the platform for the specific domain and context considered (i.e. the URL of the instance of the Context Server, used to identify the triggers to show; the URL of the Rule Manager, which stores the rules created through the Rule Editor).

To summarise, while the customizations mentioned before were done on the basis of the input provided by relevant stakeholders in the considered domain, the Tailoring Tool allows end users to personalise the combination of events, conditions and actions needed to model their intended automations, as well as the values that their attributes can assume. However, in the current version they cannot directly modify neither the structure of the hierarchies of triggers and of the actions currently shown by the Tailoring Tool, nor the types of the attributes associated with the various elements included in its hierarchies.

5.3 The Tailoring Environment for the Paper Industry Domain

The Rule Editor supports trigger and action selection by displaying the available ones organised in logical hierarchies that can be configured according to the needs of the considered domain. In this study the configuration has addressed an exemplary paper mill.

The triggers refer to three contextual dimensions (*User*, *Technology*, *Environment*), while the actions considered state changes of factory appliances, or the generation of reminders and alarms. In particular, the *User* dimension covers aspects associated with workers, who can be of three types: managers, front-line operators (working 'on the floor'), and technicians (i.e. those in charge of equipment maintenance). Their specification is refined into "Physical aspects" and "Position". The first one refers to situations where workers are moving or not (such as the well-known "Man-Down" event). The current position of users can be specified in absolute terms (via GPS) or according to some "points of interest" within the factory (e.g. "Raw Material Warehouse", "Production Line 1", "Pulper").

The *Environments* dimension is refined according to key environments/departments of the factory (e.g. Raw Material warehouse, Finished Product warehouse, Production

Department, Offices). All are characterised by typical environmental properties, such as light level, noise, smoke, pollution, humidity. In addition, warehouses also have 'Entry Speed' and 'Exit Speed', namely the rate at which raw material (resp.: finished product) enters/exits a warehouse, and also the "capacity" currently reached in each warehouse (i.e. empty, almost empty, almost full, full). The warehouses can be internal or external, according to whether they are managed within the company or not.

Regarding the *Technology* dimension, the following elements have been considered: Pulper (the machine that produces pulp from cellulose), Desiccator (which dries excessive water from the paper web), Weight Scale (at the end of the production cycle, it measures paper trim losses), Elevator (the cart moving materials within the plant). Of course, we also considered Production Lines as another key technology. All of them have the following attributes: "Efficiency" (a value in percentage terms, defining the efficiency of the equipment) and "Status" (whether the equipment is working, in pause or is stopped). The Production Lines (see Figure 3) consider additional aspects: Entry Speed (the speed at which raw material is consumed), Exit Speed (the speed at which the final product is delivered), Jumbo Roll Weight (the weight of the reel produced at the end of the production cycle), Paper Grammage (the basis weight of the paper), Paper Waste (the paper trim losses measured by the weight scales at the end of the production line), and Order Type (the type of "job" currently managed by the production line, refined in terms of Type of Customer and Type of Product Requested, thus specifying the customer who commissioned a specific order and the type of product requested).

Actions have been categorised into Alarms, Reminders, and actions on the Production Lines. Alarms and Reminders are refined basically using the same parameters: the text to send, the notification mode (i.e. mail, SMS, push notification), repetition times, and the recipient (i.e. a phone number or a mail address depending on the notification mode). The other actions aim to change the state of a line (stop, start, pause), or change the light of the semaphore associated with the production line (red, green, yellow), and also modify the recipe used for feeding the production line.

Finally, to more properly cover the needs of the considered domain, in the Rule Editor we enabled different "views" of the hierarchies of triggers and actions, depending on the type of user who accesses it and their associated rights/privileges. Indeed, access to the Rule Editor also implies the possibility to have the control of particular equipment/machinery of the company, which of course must be allowed only to specific roles. Thus, beyond the "responsible" role (who can access the whole hierarchies) there is also an "external operator" role, who can access only a portion of triggers and actions, namely those operating on the specific entities this role can manage (i.e. a subset of warehouses). Finally, rules can also be shared with others, using a public rule repository.

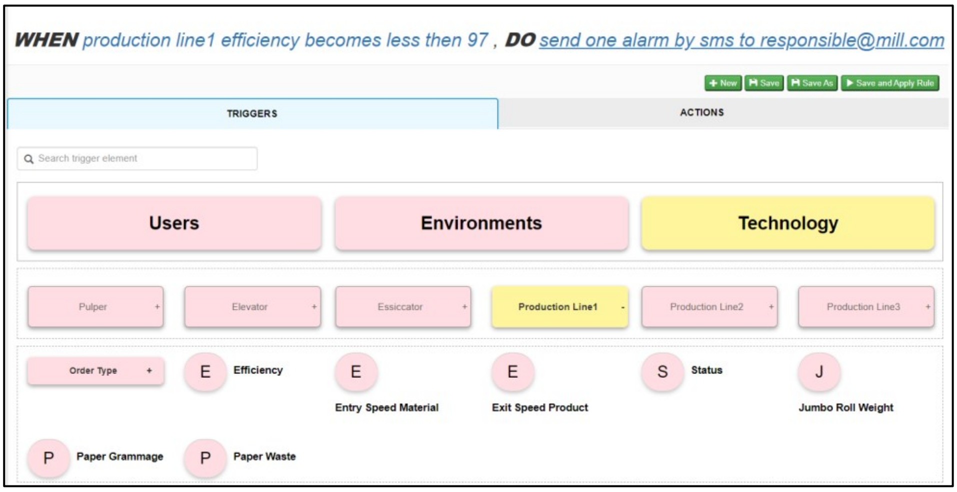

Fig. 3: The Rule Editor for the considered domain

6 User Study

We carried out an empirical test to get feedback from real stakeholders on the potentialities of the platform and the approach. The test was remotely conducted. Potential participants were recruited from the network of the members of a Regional project, trying to involve non-software developers (i.e. heads of departments, managers). They were first contacted by phone/email to ask for their willingness to participate. Then, they received an email detailing the test structure (also including info on the processing of personal data and the request for informed consent), its objectives, and main functionalities of the Rule Editor and the simulators (also with a short video). We also sent them the tasks, the links to the tools and the simulators to use for the test (with associated credentials), and to the online questionnaires to fill in anonymously after the test (it included the SUS Questionnaire, and further ad-hoc questions about the approach and the tool). The metrics considered were errors (how many and of which type), and task success categorised as follows:

- *Success*: the user has not made any mistake;
- *Failure*: the user gave up or did not complete the task;
- *Minor problems*: the user made one or two errors;
- *Major problems*: the user made more than two errors.

We considered an error a difference between the rule defined by the participant and its correct specification. Possible errors on trigger specification are: i) use of an event instead of a condition and vice versa; ii) use of a trigger element other than the one expected (e.g., using a trigger that involves the "dryer" element instead of one involving the "pulper"); iii) selection of an incorrect attribute within a trigger (e.g. instead of specifying "inside" an environment (e.g. a department) specifying "outside" it); iv) inclusion of an additional trigger, not required by the rule; v) a missing trigger. Except for the first type (which deals with the event/condition distinction, peculiar to triggers), similar types of errors were considered for actions.

6.1 Tasks

The tasks were identified to allow users to evaluate different aspects of the approach (trigger/action composition, events vs. conditions), and were proposed according to increasing difficulty levels (progressively asking to do more, and respecting more constraints).

- *Task1*: Write in your own words two rules you consider relevant in the paper sector.
- *Task2*: Using the Rule Editor, build a rule that you consider significant, containing one trigger and one action. Save the rule as "task2".
- *Task3*: Using the Rule Editor, build a rule you consider significant, containing two triggers (combined through AND or OR), and one action. Save the rule as "task3".
- *Task4*: Using the Rule Editor, specify: "As soon as the temperature of the Production Dept. exceeds 30 degrees while the operator is within it, send an alarm SMS to 0011223344". Save the rule as "task4".
- *Task5*: Using the Rule Editor, create two rules:
    - *Task5.1* As soon as paper waste on Production Line1 turns out to be less than 30 kg, a green light is turned on. Save it as "task5_1".
    - *Task5.2* In situations in which the weight of paper waste on Production Line 1 turns out to be equal or beyond 30 kg, a yellow light turns on and an e-mail is sent to your mailbox. Save it as "task5_2".

After creating Task5's rules, users had to activate them in the Rule Editor, use the event simulator to set the context in which the rule is triggered, then check that the executed actions (displayed in the action simulators) were those expected.

6.2 Participants

The test user group was made up of real stakeholders (i.e. experts in the paper sector), familiar with using web applications, but no additional skills were required. Six participants were involved in the test (1 woman), all different from those involved in the interviews. Before the test, the participants had never used the applications to test in the trial. The average age of the participants was 45.3 years (min = 40; max = 53, SD = 4.7). Three participants have a high school Diploma, two users have a Master's Degree (one in Physics, another in Aeronautical Engineering), the latter has a Bachelor's degree in Electrical Engineering. Three participants declared to have no knowledge of programming languages, the others had limited/low knowledge, one user declared to have good knowledge of industrial programming languages. All users have good familiarity with the web. Their companies range from the production of tissue, to production of paper converting machines, to developing services for automation/industrial applications; most of them have 50-250 employees, one company has more than 500 employees. One user is the head of the company's IT department, another is the sales manager for machine components; another participant deals with the sale of spare parts, another with the planning and coordination of maintenance and warehouse management, another deals with solutions for predictive maintenance, the last user is responsible for company's quality and safety. Most users have more than 10 years of experience in the paper industry, with 50% having more than 15 years. 5 users never used any tools for customising applications before the test (one user mentioned the Voith OnCare tool).

6.3 Results

In Task1 users were asked to report two rules in natural language, which they considered significant. Examples of rules created are:

WHEN number of knife cuts = X, DO send to maintenance the following text "number of knife cuts = X, blade change required;

WHEN the traffic light associated with the line signals a reel deviation>10 kg, DO send a warning via email;

WHEN a man-down is detected DO call the safety officer.

WHEN reel diameter = 10m, DO send message to production asking to change the reel.

By analysing the rules, users generally exploited a rather simple structure (one trigger, one action). Three out of the six involved users referred to man-down scenarios in their rules, whereas the actions were generally notifications/alarms/warnings. Task2 required building a one trigger-one action rule that the user considered significant in their domain, while from Task2 onwards users were required to use the Rule Editor. All the rules built by users included sending an alarm as an action. Three rules correctly included an event trigger whereas in the other rules a condition trigger was used: the latter, when combined with an instant action (e.g. sending an alarm), would result in repeatedly sending the notification, a situation that does not always correspond to a desired one. However, all users at most experienced minor problems with Task 2. Some example rules are:

IF operator is laying down, DO send one alarm by mail to sistema@company.com;

WHEN production department temperature becomes more then 40C, DO send 3 alarms by mail to maintenance@mill.com;

WHEN production line1 efficiency becomes less than 80%, DO send one alarm by SMS to 123456789.

Task3 required creating a rule containing two triggers (combined through AND or OR), and one action. Some examples of the rules created are:

WHEN production department smoke becomes more than 100 AND operator is near production_line_1, DO send one alarm by SMS to 123456789;

WHEN production department noise becomes more than 98 OR production department humidity becomes more then 95, DO send three alarms by mail to manager@mill.com;

WHEN production line2 paper grammage becomes 17 AND production line2 paper waste is more than 10, DO send alarm by mail to quality@farm.com.

Most of the times the AND operator was used to combine the triggers, only twice the OR was used. Alarm type notifications were included as an action type, while the most used types of triggers were of the Environment or User type. Most users completed Task3 successfully, in the worst cases with one or two errors, and no failure was reported. In Task 4 the majority of users (66.7%) experienced minor problems or successfully completed the task. For Task 5.1, all the users either experienced minor problems or successfully completed it. Task 5.2 was the most affected by errors: however, it was the most complex one, as it required both the specification of a structured rule (two triggers, two actions) and its actual execution (using the simulators). In all the tasks, the most frequent error was incorrect use of conditions and events (38.2% of the total), followed by using an attribute different from the expected one (23.5%). Figure 4 summarises results concerning task success (left part) and error types (right part).

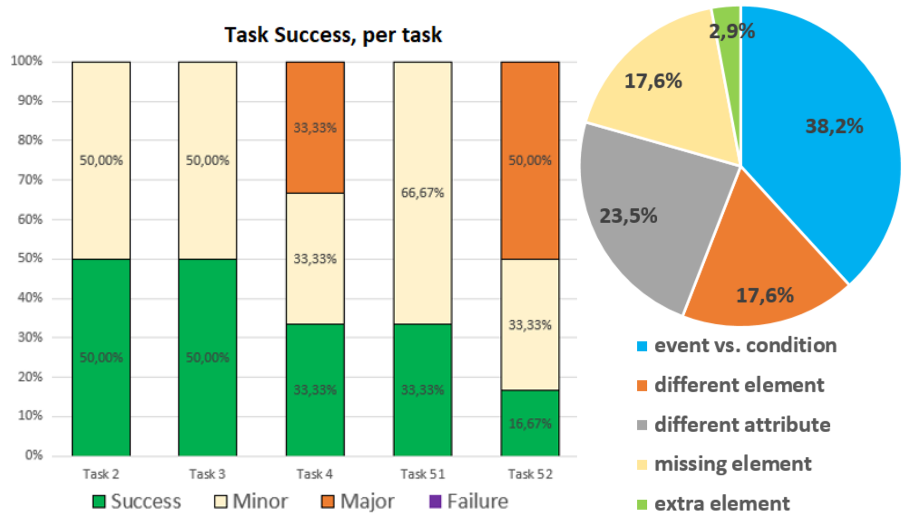

Fig. 4: Charts about Task Success (left) and Error types (right)

The average of the global values obtained by the SUS (i.e. the average level of satisfaction of the sample) was 68.8, thus denoting a more than acceptable usability. Additional questions were included to collect feedback on other aspects of the solution presented. Some questions (Q1-Q5, see Figure 5) involved providing a score using a scale from 1 to 7 (1 = not very useful / appropriate; 7 = very useful / appropriate), and also a motivation for it. The other questions gathered qualitative feedback on the most positive and negative aspects of the approach, and willingness to adopt it. As it can be seen from Figure 5, overall users appreciated the usefulness of the approach. A user stated that since it is not possible to program "a priori" all the events occurring in a complex industrial environment such as a paper mill, a dynamic handling like the one proposed is extremely useful. Two users particularly appreciated its usefulness for managing safety and production: one noted that the control of the variables manipulated by production processes well suits with a trigger-action logic to promptly act on critical situations through corrective actions. Another user highlighted, as one of its main advantages, that the approach can benefit numerous aspects of the management of a paper factory, from handling anomalies and emergencies to quality control and logistics. Both the hierarchy of triggers and of actions were overall well received, although some suggested further expanding the available choices. The description of the rules in natural language was appreciated by the users, one of them stated: "*Those who specify the rule behaviour are often unskilled users, with not high familiarity with logical conditions, then the use of natural language simplifies rule understanding*". One highlighted that this can be useful to make the rule behaviour more easily understandable also to people different from the ones who created them, thereby serving as a useful communication mean. For the event/condition distinction, they judged it “clear and concise” and “simple to use”. However, when it came to actually exploit it within rules, it seems that not all of them completely grasped it, as well as the importance of the impact that a misuse of it could have at rule execution time.

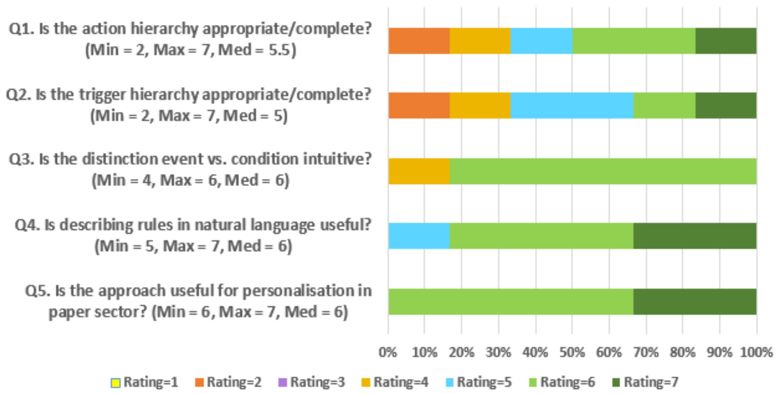

Figure 5: Chart with user ratings on some aspects of the Rule Editor Tool

Among the positive aspects of the tool (Q6), they indicated the simplicity of use and the clarity of its parts. One user reported the good potential of the solution in his company, another user found the possibility to specify alerts through various, different channels very interesting. As for negative aspects (Q7), one user would have preferred more options for triggers/actions, another said that he would have preferred additional mechanism (i.e. flowcharts) for displaying rules. When asked whether in their companies they already faced similar customisation needs (Q8), two users affirmatively replied: one pointed out that they are using a Manufacturing Execution System that integrates functions to send emails or feed SQL tables in a manner suitable also for unskilled users. Another one reported that they are creating a dashboard both at the management level and at the level of the single production plant to handle the underlying processes in a facilitated manner. Another user reported that they are considering this type of issues for situations such as downtime and/or emergencies. Two stated that these issues have not yet been addressed in the company. The last one is not aware of any initiative in this regard.

There was also a question about whether they would recommend the use of a tool like the one proposed in their company (Q9): four users answered positively. A user stated that the proposed tool could be a "plus" to be included in the automation package associated with the machines. Another user found the tool intuitive as it only requires the minimum level of understanding of if-then constructs. Two users stated that the tool has certainly good potential for exploitation in IoT and I4.0 scenarios and would be useful in their companies even though it should be further adapted to consider the multitude and the variety of objects and appliances that can be found in companies working in this domain. One especially found a high potential in making more understandable the policies that are in place in a factory also to not strictly technical people. Further suggestions to improve the tool/approach (Q10) were to include graphics (such as Zabbix[1], one said) to improve the monitoring view offered to users, and to provide a sort of "production line layout" where triggers are also visualised through their actual position on the machines.

7 Discussion

From the data collected it emerges that the tool was generally appreciated by users, even if the limited number of test participants does not allow generalisation of the gathered data, but to consider them only qualitatively as indicative of possible opportunities and promising directions, or problematic areas encountered.

One of the positive aspects –and encouraging for any future development of the platform– is that, despite the participants never had the opportunity to use the tool before the test, they were able to use it with good results, also expressing appreciation on its potentiality in the paper domain. This is especially relevant considering that the participants were real professionals, mostly senior managers operating within paper-related companies, thus having limited time available to devote to activities not strictly connected with their own work.

The proposed approach was found promising to them not only because it supports intuitive personalisation of the functioning of a complex, context-dependent system like the one typically found in companies working in this sector, and without requiring from users specific programming skills. The participants also found that the rules, which are also rendered using natural language, can support communication between different stakeholders, as they allow for externalising the knowledge of a worker to others, which in turn can be easily adapted to fit other scenarios.

Another appreciated aspect was that the considered approach provides a uniform and integrated interface that facilitates dynamic optimisation of factories according to the highly different aspects and scenarios that can emerge at various levels in industrial settings, not only vertically (i.e. within a department), but also horizontally (i.e. between different departments/units across the enterprise, between different plants belonging to the same factory). This allows them to integrate in a homogeneous manner

[1] https://www.zabbix.com/

data coming from different devices, which can be used to inform relevant business decisions. While the goal of this work was more on assessing the opportunities that introducing such approach can provide to workers in this domain in more general terms, some participants highlighted that the presented platform, while providing a promising innovative direction, could be further extended to support *real* industrial scenarios and in this regard, different opportunities can be identified. For instance, end users could be provided with enhanced visualisations able to render in an effective and efficient manner the massive number of sensors, things, appliances and actuators that can be available in real Industry 4.0 scenarios, in a way that remains usable for the workers (e.g. by filtering the hierarchies of triggers and actions, to keep only the elements that are typically of interest for the considered user role). Additionally, stakeholders could be provided with relevant recommendations for enhancing efficiency in the rule creation phase (i.e. if some actuators are often used in combination with specific sensors in particular situations, this could trigger the suggestions of suitable rules in similar situations).

As for the consideration of this platform within Industry 4.0 scenarios: it is a general platform that offers to users who are not skilled in programming intuitive means for personalising the control and the monitoring of IoT-based systems (such as Industry 4.0 smart factories), and which can be easily configured to address the needs of the specific domain and context considered. Also, since it uses open, standard (web-based) technologies, it provides support for dealing with integration and interoperability aspects, additional key issues in industrial scenarios. It also supports a flexible mechanism for dynamic inclusion of new things and services to allow platform's evolution and easy creation of further personalised services in the future (for instance, the output of a predictive maintenance system could be used as a 'virtual' sensor to trigger actions aimed to optimise maintenance tasks). These characteristics seem all important for the adoption of such kind of environments in Industry 4.0 settings.

8 Conclusions and Future Work

This paper presents an approach to supporting personalisation in industrial context by relevant stakeholders, which is obtained through an extension of a trigger-action platform. We report on its application to the paper domain. For this purpose, a set of relevant concepts and requirements have been identified through some interviews carried out with real professionals in the paper domain, which were used to suitably configure the personalisation platform for the considered sector. The approach was assessed through a user test with domain experts, which provided encouraging feedback regarding the potential adoption of the proposed approach and its integration (Mørch, 1997) in industrial settings because it can meet the flexible and dynamic organisational and technological configurations that are adopted in modern industries.

Future work will consider extending the personalisation tool integrating it directly in industrial settings, also considering the possibility of further, more intuitive support for dynamically creating the personalisation rules (e.g. considering Augmented Reality–based techniques), as well as carrying out further empirical studies in such contexts of use.

ACKNOWLEDGMENTS

We thank XXXXX company for collaboration under the XXX Project.

References

Ajaykumar, G., Steele, M., and Huang, C.-M. A Survey on End-User Robot Programming. ACM Comput. Surv. 54, 8, Article 164, 36 pages, November 2022. DOI: https://doi.org/10.1145/3466819

Asunis, L., Frau, V., Macis, R., Pireddu, C., Spano, L.D. (2021). PAC-Bot: Writing Text Messages for Developing Point-and-Click Games. In: Fogli, D., Tetteroo, D., Barricelli, B.R., Borsci, S., Markopoulos, P., Papadopoulos, G.A. (eds) End-User Development. IS-EUD 2021. Lecture Notes in Computer Science, vol 12724. Springer, Cham.

Barricelli, B.R., Cassano, F., Fogli, D., Piccinno, A., End-user development, end-user programming and end-user software engineering: A systematic mapping study, Journal of Systems and Software, Volume 149, 2019, pp. 101-137

Bellucci, A., Vianello, A., Florack, Y., Micallef, L., Jacucci, G., Augmenting objects at home through programmable sensor tokens: A design journey, IJHCS, Vol. 122, 2019, pp. 211-231

Bricklin, D., Frankston, B., and Fylstra, D., "VisiCalc, software arts," 1979. Retrieved at: http://www.bricklin.com/history/intro.htm

Caivano D., Fogli D., Lanzilotti R., Piccinno A., Cassano F. (2018) Supporting End Users to Control Their Smart Home: Design Implications from a Literature Review and an Empirical Investigation. J Syst Softw 144:295-313

Coppers, S., Vanacken, D., and Luyten, K., 2020. FORTNIoT: Intelligible Predictions to Improve User Understanding of Smart Home Behavior. Proc. ACM Interact. Mob. Wearable Ubiquitous Technol. 4, 4, Article 124 (December 2020), 24 pages. DOI:https://doi.org/10.1145/3432225

Corno, F., De Russis, L., Monge Roffarello, A., TAPrec: supporting the composition of trigger-action rules through dynamic recommendations. IUI 2020: 579-588

Corno, F., De Russis, L., Monge Roffarello, A., 2021. From Users' Intentions to IF-THEN Rules in the Internet of Things. ACM Trans. Inf. Syst. 39, 4, Article 53 (October 2021), 33 pages. DOI:https://doi.org/10.1145/3447264

Coronado, E., Mastrogiovanni, F., Indurkhya, B., Venture, G., Visual Programming Environments for End-User Development of intelligent and social robots, a systematic review, Journal of Computer Languages, Volume 58, 2020, 100970, ISSN 2590-1184

Elsden C., Feltwell T., Lawson S.W., Vines J. (2019) Recipes for Programmable Money. In: Proc. of CHI 2019 Systems, ACM, paper 251

Fischer, G., Giaccardi, E. Meta-design: A Framework for the Future of End-User Development. End User Development 2006: 427-457

Fogli, D., Piccinno, A. End-User Development in Industry 4.0: Challenges and Opportunities. IS-EUD 2019: 230-233

Fogli, D., Gargioni, L., Guida, G., Tampalini, F., A hybrid approach to user-oriented programming of collaborative robots, Robotics and Computer-Integrated Manufacturing, Volume 73, 2022, 102234, ISSN 0736-5845, https://doi.org/10.1016/j.rcim.2021.102234.

Friedow, C., Volker, M., and Hewelt, M., Integrating IoT Devices into Business Processes. In book: Advanced Information Systems Engineering Workshops. CAiSE 2018, Lecture Notes in Business Information Processing, vol 316, Springer

Gallo, S., Manca, M., Mattioli, A., Paternò, F., Santoro, C. (2021). Comparative Analysis of Composition Paradigms for Personalization Rules in IoT Settings. In: Fogli, D., Tetteroo, D., Barricelli, B.R., Borsci, S., Markopoulos, P., Papadopoulos, G.A. (eds) End-User Development. IS-EUD 2021. Lecture Notes in Computer Science, vol 12724. Springer, Cham.

Jaschinski C, Ben Allouch S (2019) Listening to the ones who care: exploring the perceptions of informal caregivers towards ambient assisted living applications. J Ambient Intell Humaniz Comput 10(2):761-778

Lucke, D., Steimle, F., Cuk, E., Luckert, M., Schneider, M., and Schel, D. Implementation of the MIALinx User Interface for Future Manufacturing Environments. Procedia CIRP, Volume 81, 2019, Pages 606-611, ISSN 2212-8271

Manca, M., Paternò, F. & Santoro, C. Remote monitoring of end-user created automations in field trials. Journal of Ambient Intelligence and Humanized Computing (2021). https://doi.org/10.1007/s12652-021-03239-0

Manca, M., Paternò, F., Santoro, C., Personalisation in a Paper Factory. IS-EUD 2021: 102-118, Lecture Notes in Computer Science, vol 12724, Springer Verlag.

Mattioli, A., Paternò, F., Recommendations for creating trigger-action rules in a block-based environment. Behav. Inf. Technol. 40(10): 1024-1034 (2021)

Modesto, A.S.C., Figueiredo, R.M. da C., Ramos, C.S., Santos, L. de S., Venson, E., and Pedrosa, G.V. Organisational Strategies for End-User Development—A Systematic Literature Mapping. Informatics 8, 1 (2021), 15.

Mørch, A. Three levels of End-User Tailoring: Customization, Integration, and Extension, Computers and design in context, 51 (1997)

Ong, S.K., Yew, A.W.W., Thanigaivel, N.K., Nee, A.Y.C., Augmented reality-assisted robot programming system for industrial applications, Robotics and Computer-Integrated Manufacturing, Volume 61, 2020, 101820, ISSN 0736-5845, https://doi.org/10.1016/j.rcim.2019.101820

Romero, D., Stahre, J., Wuest, T., Noran, O., Bernus, P., Fast-Berglund, Å., Gorecky, D.: Towards an operator 4.0 typology: a human-centric perspective on the fourth industrial revolution technologies. In: Proceedings CIE46, Tianjin, China (2016)

Senft, E., Hagenow, M., Radwin, R., Zinn, M., Gleicher, M., and Mutlu, B., 2021. Situated Live Programming for Human-Robot Collaboration. In The 34th Annual ACM Symposium on User Interface Software and Technology (UIST '21). Association for Computing Machinery, New York, NY, USA, 613–625. DOI:https://doi.org/10.1145/3472749.3474773

Stevens G., Quaisser G., Klann M. (2006) Breaking It Up: An Industrial Case Study of Component-Based Tailorable Software Design. In: Lieberman H., Paternò F., Wulf V. (eds) End User Development. Human-Computer Interaction Series, vol 9. Springer, Dordrecht. https://doi.org/10.1007/1-4020-5386-X_13

Tetteroo D., Vreugdenhil P., Grisel I., Michielsen M., Kuppens E., Vanmulken D., Markopoulos P. (2015) Lessons Learnt from Deploying an End-User Development Platform for Physical Rehabilitation. In: Proceedings of ACM CHI 2015 Conference, pp. 4133-4142

Ur, B., McManus, E., Yong Ho, M.P. and Littman, M.L. 2014. Practical trigger-action programming in the smart home. In Proc. of CHI 2014, ACM, New York, USA, 803–812.

Weintrop, D., Shepherd, D.C., Francis, P., Franklin, D., Blockly Goes to Work: Block-based Programming for Industrial Robots, IEEE 2017 IEEE Blocks and Beyond Workshop

Wieland, M., Hirmer, P., Steimle, F., Gröger, C., Mitschang, B., Rehder, E., Lucke, D., Rahman, O.A., Bauernhansl, T. Towards a Rule-based Manufacturing Integration Assistant, Procedia CIRP, Volume 57, 2016, Pages 213-218

Wieland, M., Steimle, F., Mitschang, Lucke, B., Einberger, P., Schel, D., Luckert, M., Bauernhansl, T. Rule-based integration of smart services using the manufacturing service bus. SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI 2017, San Francisco, CA, USA, 2017, pp. 1-8

Yong, y., Stecke, K., and Li D.. "The evolution of production systems from Industry 2.0 through Industry 4.0." International Journal of Production Research 56.1-2 (2018): 848-861.

Zhang, L., He, W., Morkved, O., Zhao, V., Littman, M.L., Lu, S., and Ur, B., 2020. Trace2TAP: Synthesizing Trigger-Action Programs from Traces of Behavior. Proc. ACM Interact. Mob. Wearable Ubiquitous Technol. 4, 3, Article 104 (September 2020), 26 pages. DOI:https://doi.org/10.1145/3411838