Model-Driven Framework for Human Machine Interaction Design in Industry 4.0

Patrícia Leal[1,2], Rui Neves Madeira[1,2(✉)], and Teresa Romão[1]

[1] NOVA LINCS, DI, FCT, NOVA University of Lisboa, Lisbon, Portugal
p.leal@campus.fct.unl.pt, tir@fct.unl.pt
[2] Sustain.RD Center and ESTSetúbal, Polytechnic Institute of Setúbal,
Setúbal, Portugal
rui.madeira@estsetubal.ips.pt

Abstract. Industry 4.0 (I4.0) has brought many changes in the way workers operate on and interact with smarter industrial machinery and spaces. New Human-Machine Interaction (HMI) solutions are required as the workers' demands in the factory are still evolving. This paper introduces the development of a generic model-driven HMI solution that consists of a framework providing an API and widgets that can be (re)used to create the final interfaces for HMI across different I4.0 scenarios. The generic model takes into account the most important aspects of the shop floor context. As a first phase of evaluation, this solution is being tested in a prototype for a real use case, responding to the needs of an industrial partner. The paper ends by presenting this prototype to show the potential of the presented solution.

Keywords: Industry 4.0 · Human-Machine Interaction · Model-driven · OPC UA

1 Introduction

The fourth industrial revolution, known as Industry 4.0 (I4.0), marks the evolution to Cyber-Physical Systems (CPS), being strongly driven by the advent of the Internet of Things (IoT), towards a smarter automation and data exchange in manufacturing technologies. I4.0 has brought many changes in the way shop floor users should interact with smarter industrial machinery. HMI proposals need to allow a greater symbiosis between humans and machines, providing a smooth user interaction with a pervasive computing-based environment, leading to a safer, more productive and more engaging work-life [1, 2]. In general, new HMI solutions need to be tailored to specific scenarios, which can vary widely. It means designing, developing and testing HMI solutions from scratch make data reuse difficult due to the changes in data models and protocols used, which is costly and time-consuming [3]. The lack of a global framework serving as a common basis to guide in the HMI design for different scenarios may a key obstacle to the full realization of I4.0.

This paper introduces the development of a generic model-driven HMI solution (smartHMI4I4) that consists of a framework providing an API and widgets that can be (re)used to create the final HMI for different devices across many I4.0 scenarios. It aims

Published by Springer Nature Switzerland AG 2019
D. Lamas et al. (Eds.): INTERACT 2019, LNCS 11749, pp. 644–648, 2019.
https://doi.org/10.1007/978-3-030-29390-1_54

to make the process of mapping all the industry data more streamlined and agile, while trying to support a standard way of developing HMI solutions for I4.0. As a first phase of evaluation, this solution is being used in a prototype for a real use case to assess the potential of the developed solution, responding to the needs of an industrial partner.

2 Background and Related Work

It is common to find different types of users in a factory environment, such as, team leaders, machine operators, or even directors, who may interact with each other, directly with machines or with the shop floor as a whole, needing to access and manipulate different information that can still present overlaps [2, 4]. I4.0 aims to provide the relevant data and services in context of use to the shop floor's users, according to their needs and preferences. Appropriate HMI designs should present consistent, functional and easy-to-maintain interfaces with multi-platform and multi-device support toward a pervasive user experience that do not distract users from their primary functions. There are some interesting commercial solutions, such as iFix with their HMI/SCADA software [5] providing rapid application development for system integrators, and myHMI [6] an HMI architecture oriented towards the supervision of industrial plants, which comprises of a server that handles the communication with the field by using an OPC XML-DA based module to retrieve data. UX-FAB [7] seems to be an interesting project with points in common to our solution and focused on a Web-based platform for the development of interfaces oriented to industry, multi-platform and devices of different sizes. However, these solutions do not meet every necessity, or they are closed solutions that offer no possibilities for further customization by third party developers.

3 Generic HMI Solution for Industry 4.0

The architecture of smartHMI4I4 can be seen in Fig. 1, where the bottom layer represents a specific shop floor zone (can be a whole factory), feeding the Smart Object (SO) Model, which models the computing-augmented machines of the shop floor's CPS, and the HMI Model. This one uses the SO model and describes everything else in the smart factory scenario for HMI purposes. The API and widgets are based on the guiding HMI Model to provide a standard and agile development of applications (apps).

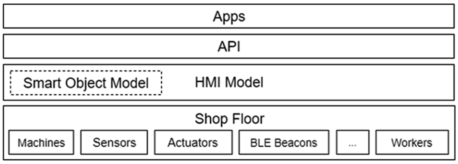

Fig. 1. Architecture of smartHMI4I4

3.1 The Generic HMI Model

The HMI Model characterizes the elements of the whole industry's scenario, considering important entities of the shop floor, such as the production numbers, the smart machines and their layout, the user profile, or the physical context (environment measures). Figure 2 shows the HMI model diagram, where the workers are represented by the User element, while the Scenario element represents the characteristics of the shop floor, such as noise, danger and temperature levels, which are important to define interaction features (e.g., screen type, mobile vs stationary, by voice). CPS is associated with the scenario and represents the smart objects (machines) composition (taken from the SO model). The ArrangedData element models the CPS data that have been transformed and structured to be shown in the HMI devices.

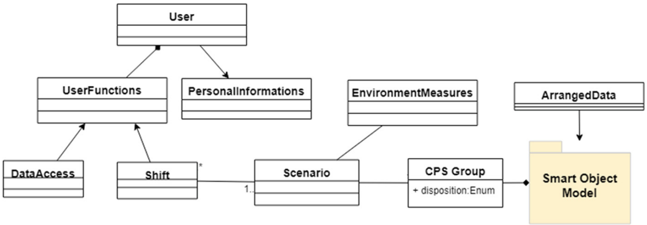

Fig. 2. A partial and simplified view of the HMI Model.

3.2 The SmartHMI4I4 Implementation

The implementation was divided into two components: HMI server; API/widgets for clients. The HMI Server Layer was created to support the HMI framework and it was implemented using Node.js. It manages the model and it is where data is stored, processed, aggregated and transformed, being prepared to be consumed. Figure 3a shows a diagram of the modules of the implemented server. The Data Retrieval module deals with data coming from the IIoT Platform that gathers data from the environment ("things" in the scenario) and transforms it in order to fit the HMI model. This component also contains both an OPC UA Client, which communicates with the Smart Objects (the machines with external sensors, actuators and OPC UA capabilities) of the shop floor, and an OPC UA Server, which connects to the HMI devices to track data coming from the Smart Objects. OPC UA is a platform-independent, scalable and high-performing communication protocol that provides a unified and standardized communication infrastructure for a greater degree of interoperability between different industrial machinery, factory devices and the architecture's higher-level tiers.

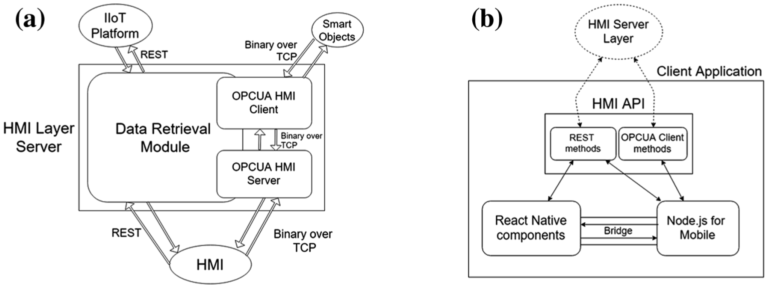

Fig. 3. (a) The HMI layer on server side; (b) the client side using the HMI API.

The client-side API was developed using JavaScript and Node.js. Figure 3b presents the diagram of the general implementation and illustrates the use of the API by an HMI app. The API's methods make requests to the server tier, which returns a Promise that can be used by the app to process the data that asynchronously arrives from the server. The API contains an OPC UA client module implemented in Node.js, offering methods that can be used to monitor and exchange data from the Smart Objects of the factory.

3.3 First SmartHMI4I4-Based Prototype - Preliminary Evaluation

As part of our evaluation plan, we created a first mobile prototype for a real scenario applying smartHMI4I4. MCG - Mind for Metal [8] presented us with the need to access information about a certain shop floor's area comprising three production machines (jigs) positioned in line. Three BLE (Bluetooth Low-Energy) beacons were placed near each jig and another one at the area's entrance, as it was required to differentiate the HMI according to the user profile (e.g. machine operator or team leader) and the user's proximity to machines. When a beacon is found (strongest signal → nearest) by the HMI mobile device of a user, it means s/he is facing the area/machine (a smart object) to which the beacon "belongs" and then the app automatically displays an adapted interface with data related to the object and tailored to the user profile.

The Node.js for Mobile module was used in order to integrate Node.js scripts with React Native (used in this app). As can be seen in Fig. 3(b), the information flow between the React Native components and the Node.js scripts in the application using the bridge built by the Node.js for Mobile module. This is the path used by the React Native client to transmit/receive information from the OPC UA client given by the API.

4 Conclusions and Future Work

Our solution is being developed to fulfil the need for a standard and agile solution that guides and supports the implementation of smart HMI across different I4.0 scenarios. smartHMI4I4 was created for application developers, being based on a generic, extendable and industry domain-independent model for HMI in I4.0.

We will develop new HMI applications applied to real case scenarios to test if smartHMI4I4-based applications work well in factory context. We also intend to conduct tests to study the usability of the framework from a developers point-of-view.

Acknowledgments. This research is supported by Polytechnic Inst. of Setúbal under the project "Smart Human Machine Interaction for Industry 4.0", FCT/MCTES NOVA LINCS PEst UID/CEC/04516/2019, and the European Regional Development Fund (FEDER) through a grant of the Operational Programme for Competitivity and Internationalization of Portugal 2020 Partnership Agreement (PRODUTECH-SIF, POCI-01-0247-FEDER-024541).

References

1. Cotteleer, M., Sniderman, B.: Forces of change: Industry 4.0. https://www2.deloitte.com/insights/us/en/focus/industry-4-0/overview.html. Accessed 28 Mar 2019
2. Johannsen, G.: Human-machine interaction. Control Syst. Robot. Autom. **21**, 132–162 (2009)
3. Fleischmann, H., Brossog, M., Beck, M., Franke, J.: Automated generation of human-machine interfaces in electric drives manufacturing. In: Proceedings of 7th International Electric Drives Production Conference (EDPC 2017), pp. 1–8. IEEE (2017)
4. Relatório Técnico-Científico Intercalar Consolidado: Programa Mobilizador PRODUTECH SIF - Soluções para a Indústria de Futuro. Technical report. PRODUTECH SIF (2018)
5. GE Digital: iFIX. https://www.ge.com/digital/applications/hmi-scada/ifix. Accessed 07 Apr 2019
6. Bozzon, A., Brambilla, M., Fraternali, P., Speroni, P.: Bringing internet architectures into the plant: the case of HMI. In: European Control Conference (ECC 2007), pp. 5530–5537. IEEE (2007)
7. Critical Manufacturing. UX-FAB: Universal Experience for advanced Fabs. http://www.criticalmanufacturing.com/en/r-d/ux-fab. Accessed 03 Apr 2019
8. MCG - Mind for Metal. http://www.mcg.pt/pt/. Accessed 03 June 2019