Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 72 (2018) 15–20

www.elsevier.com/locate/procedia

51st CIRP Conference on Manufacturing Systems

Context-dependent multimodal communication in human–robot collaboration

Csaba Kardos[a,b,*], Zsolt Kemény[a], András Kovács[a], Balázs E. Pataki[c], József Váncza[a,b]

[a] *EPIC Center of Excellence in Production Informatics and Control, Institute for Computer Science and Control, Hungarian Academy of Sciences*
[b] *Department of Manufacturing Science and Technology, Budapest University of Technology and Economics*
[c] *Department of Distributed Systems, Institute for Computer Science and Control, Hungarian Academy of Sciences*

[*] Corresponding author. Tel.: +36-1-279-6189; *E-mail address:* csaba.kardos@sztaki.mta.hu

Abstract

In the dynamic environment of human–robot collaboration, a key for boosting the efficiency of human workers is supporting them with context-dependent work instructions, delivered via communication modalities that suit the actual context. Workers, in turn, should be supported in controlling the robot or other components of the production system by using the most convenient modality, thus lifting the limitations of traditional interfaces as push buttons installed at fixed locations. We introduce a workflow for context-dependent multimodal communication in a collaborative work environment and present its implementation in a Human–Machine Interface Controller system.

Peer-review under responsibility of the scientific committee of the 51st CIRP Conference on Manufacturing Systems.

Keywords: human-robot collaboration;worker information systems;multi-modal communication

1. Introduction

In recent years, human–robot collaboration (HRC) has gradually gained foothold in several branches of industry, reflecting both advances in underlying technologies and evolution of application demands. Typically, collaborative working environments (i.e., where humans and machines share tasks, occupied space and resources in mutual awareness of each other's current and foreseen behavior) find use where the advantages of humans and machines are needed to complement each other in executing production tasks, necessitated e.g., by the complexity of the task in assembly, by the diversity of incoming assignments resulting from a rich product mix, or by the need of combining strength and precision with versatile cognitive and problem solving skills as in construction of large, often individual, products. While modes of collaboration span a wide spectrum [1,2], mutual awareness and timely information of human workforce are of key importance in most, if not all, cases.

Communication with the worker in a collaborative environment has two main facets: (1) the worker needs to know in advance what operations are to be performed, what actions are to be expected from the robot, and whether execution deviates from a nominal track/conditions; and (2) feedback from the worker must be captured regarding success/failure of actions, possible deviations of the worker from the nominal operating track/conditions, and the worker's current situation within the working environment. In conventional human working environments, these were distinctly separated in the form of—typically static, paper-based—work instructions and signaling/feedback instruments. Recent trends point towards a seamless fusion of these components, moreover, technological development is increasing the possibilities of dynamic configuration of communication modes, channels, and content. The latter is especially important in environments where the context of human–machine relations changes at runtime, which frequently occurs in the case of dynamic resource assignment and high task diversity (i.e., where collaborative environments perform best).

The paper recapitulates the requirements of communication with human workforce in collaborative production environments, and presents an implemented example of a context-dependent human–machine interface which exploits several modes of communication. While the previous paragraph outlined a wide spectrum of roles and functions regarding communication with human workforce, the particular context of the paper focuses on introducing a system for delivering work instructions and providing multi-modal communication interfaces, in an industrial assembly setting.

In further parts, the paper is organized as follows. Section 2 gives an overview of preliminaries found in literature, followed by a problem statement; Section 3 defines the contextual aspects of the HRC situations examined; Section 4 explains modes of communication in HRC, and Section 5 presents a pilot implementation case of a multi-modal, context-dependent human–machine interface.

2212-8271
Peer-review under responsibility of the scientific committee of the 51st CIRP Conference on Manufacturing Systems.
10.1016/j.procir.2018.03.027

2. Literature review and problem statement

2.1. Review of previous work

As maintaining process and product quality is a key issue in production systems, having proper documentation of processes and work instruction (i.e. description of operations, parts, equipment, resources, safety regulation and commands regarding the production tasks) is mandatory. The traditional—and still widely applied—form of work instructions is paper-based documentation which offers simple implementation, however, its static structure makes it difficult to maintain and update over time. Thus, in today's manufacturing—where the product lifecycle is becoming shorter and product diversity is growing—paper-based work instructions are being replaced with enhanced (digital) Worker Instruction Systems (WIS) [3,4].

The main advantage of using WIS over paper-based solutions is their ability to be linked to multimedia instruction databases or other information sources, allowing dynamically changeable content. This suits the requirements of quickly changing manufacturing environments. Many companies have developed their own WIS solutions, but also, several commercial frameworks are available, with various options to link to instruction databases. Most of them offer interfaces to import product and process data from Enterprise Resource Planning (ERP), Product Management (PLM) and Manufacturing Execution Systems (MES), supporting the definition of instruction templates, which the imported information can be linked to. Application of WIS allows maintaining consistent and up-to-date work instructions.

The modalities offered by commercial WISs are mostly visual—e.g., text, figure, video, 3D content—in some cases extended with audio instructions or commentary. The required user interaction is, in most cases, provided by contact-based means, e.g., a button or a touchscreen, however, more advanced cases can also rely on Near Field Communication (NFC) or Radio Frequency Identification (RFID) technologies.

Commercial WIS applications are mostly focusing on maintaining the consistency of work instructions, however, with the increasing complexity of production systems, there is need for enhanced support to the human operator as well. On one hand, assembly systems offer an efficient answer to handling the growing number of product variants and the need for customized products, while, on the other hand, adapting to this variety puts additional cognitive load on skilled workforce [3,5]. More information to be handled by the operator means more decisions to be made, which can easily lead to errors in these decisions [6]. In order to handle this increased cognitive burden, improvements in WISs have to be implemented. With the prevailing technologies offered by cyber-physical devices, new research and numerous new papers are focusing on ways of utilizing them in worker assistance systems.

Hollnagel [7] proposes three important questions (aspects) regarding worker information systems: *what* (content), *how* (carrier) and *when* (timing) to present. Based on the work of Alexopoulos *et al.* [8], Claeys *et al.* [9] focus on the content and the timing of information in order to define a generic model for context-aware information systems. Context awareness and tracking of assembly process execution are also in the focus of research by Bader *et al.* [10] and Bannat *et al.* [11]. A special field of interest can be quality check or maintenance operations, with an even higher variety of tasks—Fiorentino *et al.* [12] suggest to address this challenge using augmented reality instead of paper-based instructions to increase productivity, while Erkoyuncu *et al.* [13] propose to improve efficiency by context-aware adaptive authoring in augmented reality.

This review of currently available software solutions and works in WIS solutions shows that—besides recent development of commercial systems (connectivity, handling various contents, etc.)—technologies delivered by cyber-physical devices offer new ways to further improve the capabilities of WIS solutions [14]. Nevertheless, the recent trends in assembly systems (e.g., large product diversity and, especially, the prevalence of collaborative applications) make it necessary to increase the efficiency of bidirectional communication by providing processing context related information and multi-modal interfaces.

2.2. Problem statement

Assuming that human–robot collaboration (HRC) is taking place in a workcell equipped with a single robot, and a sequence of tasks to be executed is given, with each step assigned to either human workers or to the robot, a worker assistance system is required to provide bidirectional interfaces for the human workers. Based on the literature review, the expectations and requirements towards such an assistance system can be formulated, and the problem can be stated as the definition of a worker assistance system capable of:

- Processing and storing input data coming from connected systems which defines the context of the worker's activity and the process subject to execution.
- Providing multi-modal bidirectional communication to and from the human worker, with the main goal of delivering content according to the specifications of process engineering.

The problem statement also implies that HRC assistance is provided not by a standalone system but as a part of an HRC ecosystem, the latter being composed of the following:

- A *Unit Controller* (UC) controlling and tracking the task execution within the workcell and also implementing robot controller functions.
- A *Cockpit* responsible for scheduling and rescheduling the tasks between the available resources (e.g., robot or human worker) and monitoring the shop-floor.
- A *Mobile Worker Identification* (MWI) system for locating human workers across the shopfloor and within the workcell, also transmitting this information towards the cockpit.

3. Context definition and information flow

3.1. Context of process execution and worker

In order to deliver instructions in a way that suits the worker and the situation the best, the worker assistance system must obtain information about the following aspects of the working context:

- *Worker properties context*: skills, abilities, and preferences regarding assistance of the given worker.
- *Worker location context*: the current location of each individual worker within the plant detected by the MWI systems. MWI notifies the Cockpit about significant changes in worker location, which in turn forwards this information to all relevant parties, including the worker assistance system instance belonging to the affected workcells.
- *Worker activity context*: worker actions that encode commands to the robot (or other assembly system components) are captured by sensors of various types placed in a wearable suit or deployed at the workstation. These devices are managed directly by the worker assistance system.
- *Process context*: task and operation definitions are originally specified externally, then managed and distributed to the relevant resources by the Cockpit and the UC. Upon execution of a pre-defined task, the worker assistance system is notified of the process context by the UC.

Note that imminent safety hazards may also have their own—possibly changing—context, however, their priority places them into a separate subsystem which is required to work reliably even in the presence of faults in other subsystems. Due to their special requirements, safety surveillance and warning functions are exempt from the considerations in this paper.

3.2. Information flow

When an *operation is to be performed* in the workcell, the UC sends a request for displaying the adequate instructions for the worker (in Figure 1 the information flow is laid over the implemented system's architecture, marked "1", arrow denotes direction of information flow). The content and delivery mode of the instruction is selected in accordance with the current execution context (operation and worker identities, worker skill level and individual preferences, obtained from the context related input (2) or determined by the state of internal execution logic of instruction delivery). Next, the instructions are delivered to the selected HMI display device (3a, b). Upon termination of the operation, the worker responds with a success or failure response (4a, b). This action is matched with messages valid in the given context, and the selected message is passed on to the UC (5). The UC then determines the next action to perform, in view of the outcome of the preceding operation.

Means of *ad-hoc communication* towards the worker are also provided in order to be able to communicate outside the above described normal process flow of displaying operation related instructions. This is typically needed when exceptions occur in the execution, *requiring human intervention* for correction (removal of obstacles, repositioning of workpiece to resume normal process flow), or recovery (removal of faulty workpiece or taking the cell offline for tool replacement, repair, etc.). In such cases, this information is displayed as a message to the worker, and a confirmation message is returned to the UC. This essentially corresponds to the information flow of normal process execution, however, the context and nature of messages and actions are set by the detected exception.

Similarly to the ad-hoc messaging, the worker needs to be notified about *changes in robot/cell status* (e.g., when the robot starts moving). Unlike the previous cases, this information does not need to be confirmed by the worker, i.e., steps 4a and 4b of the normal information flow are omitted. A confirmation is sent to the UC (step 5), however, this only means that the message was successfully displayed.

Ad-hoc commands can be issued by the worker to intervene with the normal process flow when needed (e.g., slow down, halt robot, retract robot, resume operation). The UC can specify the set of valid commands or messages for a given context, which are then displayed along with the robot status as a palette of command options, using the same information flow as explained in the previous paragraph. The worker can issue one of the displayed ad-hoc commands through the input devices available in the given context. Once recognized, the command is sent to the UC, passing through flow steps 4a–5. The 5th step is, in this case, a *command*, as opposed to a confirmation in previously described cases.

4. Multi-modal communication

4.1. Visual instruction delivery

Providing visual instructions (e.g., text, images, videos) is in line with traditional and generally applied methods of instruction delivery. However, displaying visual instructions can, nowadays, involve multiple formats and channels, each having advantages and disadvantages in the given application context. While creating the instruction material, the process engineer may have several delivery channels at hand, and various preferences may be taken in consideration for a particular choice. There are, nevertheless, binding requirements to be fulfilled with the channel chosen, e.g., context-aware responsiveness of instruction delivery towards the human operator is needed during task execution. In order to support the above requirements of a flexible, responsive and portable user interface, a web-based visual interface was chosen. This has the advantage that most mobile and desktop devices are able to render web pages without any additional component or software. Web pages are also perfectly suited for providing templates for displaying content dynamically—compliance with the HTML5 standard also enables interactions and visualization of 3D objects.

In addition to text and images, video contents are used widely in operators training, as they are able to show the processes dynamically and display the real production environment without any simplification. The offered interactivity is, however, limited as viewpoints or focus cannot be adjusted once the video is recorded. Embedding and displaying videos are seamlessly supported by modern web browsers.

To overcome the limitations of video content, 3D models can provide interactive views, enabling the user to highlight and manipulate single or multiple elements, and change the viewpoint. Embedding 3D models into a web-based visual interface is, therefore, mandatory. Using the X3D technology, complex 3D scenes with hierarchical model structure can be displayed. Moreover, X3D's support of animations enables the combination of moving sequences (as in a video) with the interactivity of a 3D model. X3D is now a part of the HTML5 standard, and thus supported by most of the currently available web browsers, which allows its easy integration. Through DOM (Document Object Modeling), the embedded X3D scenes can be accessed and manipulated via a web browser. The triangle mesh representation used by X3D offers a generic and scalable represen-

tation, as the resolution of the mesh models can be set using mesh decimation technologies. This increases flexibility and portability as the resolution of the model can be tailored to the available resources of the display device and the overall complexity of the scene displayed.

4.2. Audio instructions and commands

Audio work instructions are used less frequently compared to visual channels. However, HRC can be one of their key application areas, due to process complexity and intensity of communication calling for non-visual media as well. Audio channels can be used to inform the worker about the state of the robot and also to deliver textual work instructions without the need of the worker's eye contact. The application of multilingual text-to-speech (TTS) technology can offer a solution to both. Also, it is possible to embed such a solution into a web-based interface using an API (application programming interface). Speech-to-text solutions are also available for providing an audio input interface for the worker.

4.3. Contact-less commands

Using contact-less modalities (i.e., not requiring mechanical contact) as input have importance in manufacturing environments where contact-based devices (e.g., buttons, touchscreens) may be difficult to use in certain situations, and audio commands may be error-prone due to ambient noise. While these technologies might currently have limited robustness in industrial environment, they show potentially new use-cases where the worker is not required to use hand-held devices for transmitting commands.

Hand and arm gesture recognition is an efficient and easy way for human worker interaction. Gesture communication is fast, natural and easy to learn. Hand gesture can be captured by a sensor-equipped glove that is capable of measuring the momentary state of the worker's hand, matching it to previously recorded samples, and issuing the corresponding command paired with the gesture recognized. Arm gestures can be extracted from a 3D point cloud image of the workstation containing the worker [15], acquired either by a single camera or by merging data from several imaging devices [16].

5. Implementation and use-case

5.1. System architecture

The general concept of providing the worker with a wearable mobile assistance suite is guided by the recognition that the wearable components do not necessarily need to implement autonomous or smart behavior. The most important function of the worker assistance system is to provide multi-modal and bidirectional communication through multiple devices between the worker and the Unit Controller (UC). As the UC is a single, dedicated instance in every workcell, this mode of operation is best supported by a server–client architecture where the majority of the logic is implemented on the server side. Placing the logic into a central component is advantageous as it increases the compatibility offered towards the connected devices which, therefore, are able to connect and communicate via standard interfaces and protocols (e.g., HTML5). The server–client-based human–machine interface system implemented in accordance with the aforementioned requirements is named Human Machine Interface Controller (HMIC), Figure 1 shows its major structure and connections to other workcell components. The HMIC consists of two main parts. (1) The *backend* is responsible for implementing the business logic (i.e., the instruction data model and database) required for the content handling, and the interfaces for communication towards the the Cockpit and the UC. (2) The *frontend* provides the necessary interfaces towards the devices of the wearable assistance suite.

5.2. Data model

The HMIC backend is coupled with a database (HMIC DB, see Figure 2) that stores the context data and all the relevant information that is delivered to the worker through the assistance suite. These data are queried during the execution time content generation by the backend before delivery, in order to have the delivered information suit (1) the context defined by the process execution, (2) the devices available at the worker, and (3) the skills of the worker using these devices. The delivery of the instructions is triggered by the HMIC input interfaces. The most important fields of the database are detailed below.

Similarly to the representation applied in the Cockpit and the UC, the *task* is a fundamental element of the HMIC representation, as it links operations to workstations. It is the smallest element tracked by the Cockpit during execution.

An *operation* is the smallest element of task execution in the HMIC DB. A task is built up by a sequence of operations.

Instructions are possessing one or more operations, which are delivered to the worker in the format defined by the instruction. For each instruction, there is a skill level specified, which allows having different, skill-level dependent instructions for the same operation.

The *worker* table contains the defined task skills, information regarding the devices available for the worker and the identifier of the workstation, where the worker is located in. This field is updated as a result of the worker identification.

The *task skills* assigned to each worker contain records that describe the skill level of the worker for a given task. Therefore, instead of having one skill level for every activity, it is possible for a worker to have different skill levels for different tasks, which is useful, for example, when the worker is transferred into a new role. (By default there are three levels available: beginner, trained and expert.)

Input and output devices available and assigned to each worker are stored in *device* tables.

Each device is an instance of a *device class* object, which specifies the input and output channels available for a certain device. As the device class can have multiple channels, it is possible to define multimedia devices, e.g., a smartphone device can be defined to provide channels for audio input/output, screen and buttons.

Using *interactions*, it is possible to define how specific input channels are allowed to receive commands from the worker.

5.3. Implementation and use-case

The HMIC Backend has been implemented as a Spring Framework based Java web application providing 3 communication interfaces:

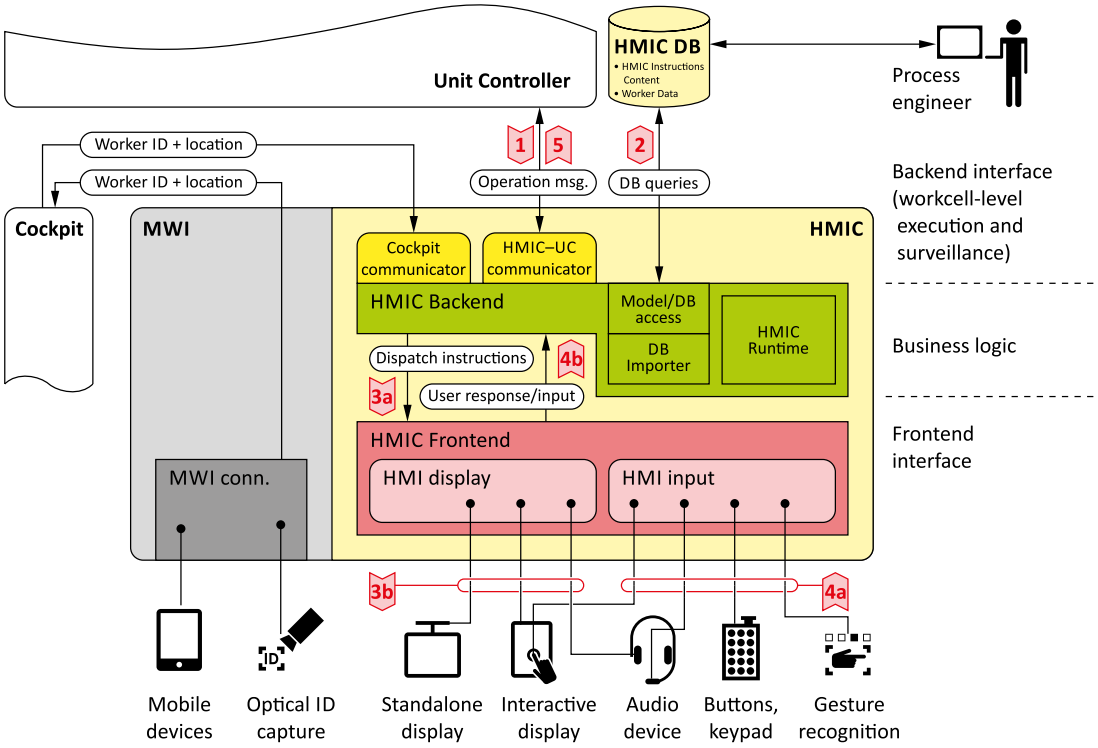

Fig. 1. Schematic architecture of the HMIC implementation and its immediate environment in the production system

- ZeroMQ messaging between the HMIC and UC. For this, the HMIC provides a server port to receive messages from the UC and connect back to a server port provided by the UC to send messages. This way, messages between HMIC and UC can be sent asynchronously.
- REST interface for accessing data managed internally by the HMIC, such as the descriptive data of the Workstations, Workers, Operations, Instructions, etc.
- WebSocket interface using STOMP messages via SockетJS to handle delivery of instructions and handling user input. The HMIC frontend and other device handlers, such as the gesture glove and Kinect-based gesture controllers, connect to the HMIC via this interface, receive instructions, and send back commands as necessary

The HMIC frontend is the main client to the HMIC system providing display of instructions on various displays including monitors, tablets, mobile phones and AR (Augmented Reality) glasses. It also provides handling user interactions in the form of button based inputs. The HMIC frontend is implemented as a NodeJS application with React based user interface. The gesture glove controller, Kinect based gesture controllers only provide ways to interpret gesture based user inputs either via a special smart glove developed by SZTAKI, or via a Microsoft Kinect device.

The implemented HMIC system was demonstrated in the laboratory simulation of an automotive assembly use-case, where 29 parts were assembled in 19 tasks (for details see [17]). The content was prepared and delivered to the system via XML-files structured according to the HMIC DB schema. In order to test the multi-modal interfaces, the following devices were connected to the system: a large-screen tablet device, a smartphone, a gesture glove, a Kinect device and an AR glass. The devices with HTML5 web browsers were registered directly in the frontend of the system, while those without web-browsers were connected via their developed client applications that provide the necessary interfaces according to the specifications. During the tests the system was able to handle all devices in parallel, thus providing a wide variety of communication channels concurrently. The assembly tasks and the worker identification signals were issued by using mock interfaces of a UC and a Cockpit and the system was able to handle inputs from and to deliver the content to the multiple devices with an adequate response time.

6. Summary, future work

As human–robot collaboration is gaining acceptance in the industry, growing demands for adequate human–machine communication interfaces are perceived. The paper presented a systematic description of aspects and requirements of a human–machine interface serving the needs of human–robot collaboration where (1) the interaction context may be subject to change from operation to operation (i.e., it is *context-dependent*), and (2) multiple modes of communication are used, depending on the characteristics of the environment, the task, and the type of collaboration (i.e., the communication is *multi-modal*). The paper also presented a possible solution of a context-dependent, multi-modal human–machine interface which was built up and tested in a pilot implementation.

As part of a research project, an industrial use-case is planned to be implemented as the next step. Further research and development is planned to follow several tracks in connection with the human–machine interface presented here. Recently, Kardos *et al.* [17] have developed a feature-based assembly planner which is extended to generate detailed work instructions for the introduced HMIC system. Also, communication to hierarchical layers above workcell level, and repre-

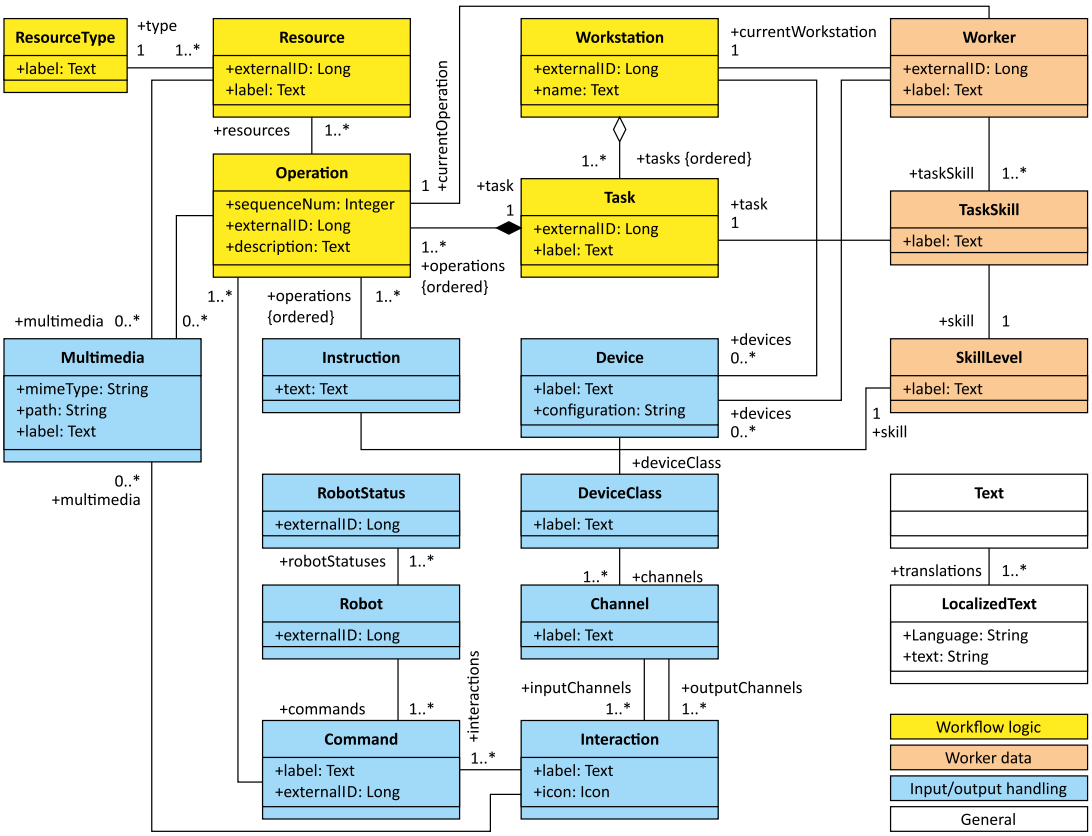

Fig. 2. Structure of the HMIC DB showing elements for storing instructions and the context related information (note that relations to text and localizations are not shown for simplicity)

sentation of HRC characteristics in higher-level planning and execution control are to be covered in future research, enabling a seamless gradual introduction of new collaborative workstations in existing production systems.

Acknowledgment

This research has been supported by the GINOP-2.3.2-15-2016-00002 grant on an “Industry 4.0 research and innovation center of excellence” and by the EU H2020 Grant SYMBIOTIC No. 637107.

References

[1] Yanco, H.A., Drury, J.. Classifying human–robot interaction: an updated taxonomy. In: Systems, Man and Cybernetics, 2004 IEEE International Conference on; vol. 3. IEEE; 2004, p. 2841–2846.

[2] Wang, X.V., Kemény, Z., Váncza, J., Wang, L.. Human–robot collaborative assembly in cyber-physical production: Classification framework and implementation. CIRP Annals—Manufacturing Technology 2017;66(1):5–8.

[3] Leu, M.C., ElMaraghy, H.A., Nee, A.Y., Ong, S.K., Lanzetta, M., Putz, M., et al. CAD model based virtual assembly simulation, planning and training. CIRP Annals 2013;62(2):799–822.

[4] Lušić, M., Schmutzer Braz, K., Wittmann, S., Fischer, C., Hornfeck, R., Franke, J.. Worker information systems including dynamic visualisation: A perspective for minimising the conflict of objectives between a resource-efficient use of inspection equipment and the cognitive load of the worker. Advanced Materials Research 2014;1018:23–30.

[5] Lušić, M., Fischer, C., Bönig, J., Hornfeck, R., Franke, J.. Worker information systems: State of the art and guideline for selection under consideration of company specific boundary conditions. Procedia CIRP 2016;41:1113–1118.

[6] Fast-Berglund, A., Fässberg, T., Hellman, F., Davidsson, A., Stahre, J.. Relations between complexity, quality and cognitive automation in mixed-model assembly. Journal of Manufacturing Systems 2013;32(3):449–455.

[7] Hollnagel, E.. Information and reasoning in intelligent decision support systems. International Journal of Man–Machine Studies 1987;27(5):665–678.

[8] Alexopoulos, K., Makris, S., Xanthakis, V., Sipsas, K., Liapis, A., Chryssolouris, G.. Towards a role-centric and context-aware information distribution system for manufacturing. Procedia CIRP 2014;25:377–384.

[9] Claeys, A., Hoedt, S., Landeghem, H.V., Cottyn, J.. Generic model for managing context-aware assembly instructions. IFAC-PapersOnLine 2016;49(12):1181–1186.

[10] Bader, S., Aehnelt, M.. Tracking assembly processes and providing assistance in smart factories:. SCITEPRESS—Science and and Technology Publications. ISBN 978-989-758-015-4 978-989-758-016-1; 2014, p. 161–168.

[11] Bannat, A., Wallhoff, F., Rigoll, G., Friesdorf, F., Bubb, H., Stork, S., et al. Towards Optimal Worker Assistance: A Framework for Adaptive Selection and Presentation of Assembly Instructions. 2008.

[12] Fiorentino, M., Uva, A.E., Gattullo, M., Debernardis, S., Monno, G.. Augmented reality on large screen for interactive maintenance instructions. Computers in Industry 2014;65(2):270–278.

[13] Erkoyuncu, J.A., nigo Fernández del Amo, I., Mura, M.D., Roy, R., Dini, G.. Improving efficiency of industrial maintenance with context awarea daptive authoring in augmented reality. CIRP Annals 2017;66(1):465–468.

[14] Vernim, S., Reinhart, G.. Usage frequency and user-friendliness of mobile devices in assembly. Procedia CIRP 2016;57:510–515.

[15] Horváth, G., Erdős, G.. Gesture control of cyber physical systems. Procedia CIRP 2017;63:184–188.

[16] Horváth, G., Erdős, G.. Point cloud based robot cell calibration. CIRP Annals—Manufacturing Technology 2017;66(1):145–148.

[17] Kardos, C., Kovács, A., Váncza, J.. Decomposition approach to optimal feature-based assembly planning. CIRP Annals—Manufacturing Technology 2017;66(1):417–420.