

Chapter 15

Worker Assistance Suite for Efficient Human–Robot Collaboration

Gergely Horváth¹, Csaba Kardos¹, András Kovács¹, Balázs Pataki¹ and Sándor Tóth²

¹ Institute for Computer Science and Control (SZTAKI), Kende utca 13–17, 1111, Budapest, Hungary

² SANXO Systems Ltd., Arany János u. 87/B, 1221, Budapest, Hungary

Email: {gergely.horvath, csaba.kardos, andras.kovacs, balazs.pataki}@sztaki.hu, sandor.toth@sanxo.eu

Abstract

Providing efficient worker assistance is of key importance in successful human–robot collaboration as it can reduce the cognitive load of the workers. The human and the robot together form an environment where bi-directional information exchange and context awareness play a crucial role. Using multiple channels of communication allows to fit this context to the worker’s preferences, which helps increase efficiency and the quality of production. This chapter introduces a novel worker assistance suite, which supports multimodal communication. The structure of the system and its interfaces are discussed and the implementation is detailed.

15.1 Introduction

This chapter introduces the aspects of worker assistance related to Human–Robot Collaboration (HRC) [15.1]. The increased flexibility, which is demanded from modern production systems, especially in the case of HRC, requires advancement on the worker assistance side as well. This chapter gives an overview of related work and formulates the problem as providing multimodal context-dependent assistance for human workers. This is followed by a detailed walk-through of the definition, development and implementation of an assistance system. The concepts and results presented here are based on papers by Kardos *et al.* [15.2], [15.3].

15.2 Related work

Proper work instructions are of key importance in production systems as they contribute to the quality of the resulting products. Work instructions describe the

operations, parts, equipment, resources, safety regulations and commands applicable during the execution of the production tasks. Paper-based documentation is still the most widely applied form of delivering work instructions, however, its static structure makes it difficult to maintain and update over time. With shortening product life cycles and growing product diversity, there is a need to replace paper-based instructions with enhanced (digital) Worker Instruction Systems (WIS) [15.4], [15.5].

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 Lifecycle 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 mostly focus 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 customised products, while, on the other hand, adapting to this variety puts additional cognitive load on skilled workforce [15.4], [15.6]. More information to be handled by the operator means more decisions to be made, which can easily lead to errors in these decisions [15.7]. 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 focuses on ways of utilising them in worker assistance systems.

In [15.4] authors discussed how digital models can be used to generate virtual assembly for simulation, planning and training purposes. The various modalities available in modern production systems and WISs are also introduced (see Figure 15.1).

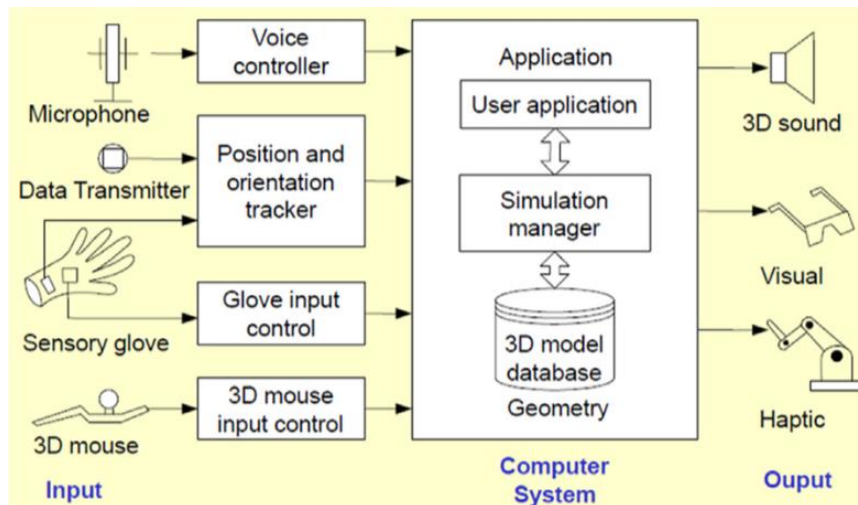


Figure 15.1. Input and output interfaces in a VR-based WIS [15.4]

In [15.8], as an early contribution to the field, three important questions (aspects) were proposed regarding worker information systems: what (content), how (carrier) and when (timing) to present. Based on the work of Alexopoulos *et al.* [15.9], Claeys *et al.* [15.10] focused on the content and the timing of information in order to define a generic model for context-aware information systems (Figure 15.2). Context awareness and tracking of assembly process execution are also the focus of research by Bader and Aehnelt [15.11], and Bannat *et al.* [15.12].

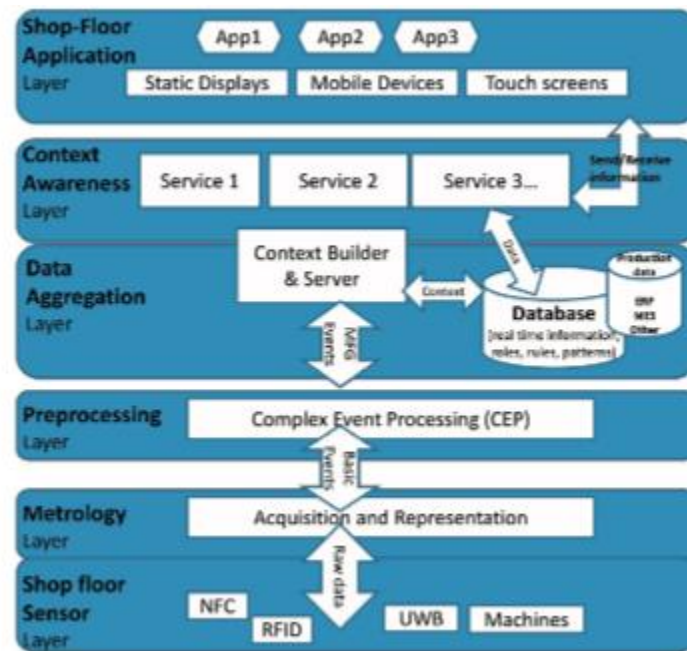


Figure 15.2. A generic model for context-aware information systems [15.10]

A special field of interest can be quality check or maintenance operations, with an even higher variety of tasks—Fiorentino *et al.* [15.13] suggested addressing this challenge using augmented reality instead of paper-based instructions to increase productivity, while Erkoyuncu *et al.* [15.14] proposed improving efficiency by context-aware adaptive authoring in augmented reality.

Lušić *et al.* [15.6] aimed to provide a methodology for selecting WIS for assembly, and they identify the following key characteristics:

- The applied modalities of the communication,
- The degree of mobility, i.e., the application of stationary or mobile devices,
- Dynamic or static provision of the information, where dynamic provision requires a suitable representation of the sequence of operations and their tracking,
- Flexibility of information delivered: context-aware or predefined information,
- Source of information delivered: local or distributed (web-based) data flow.

The impact of WIS on work performance is analysed in [15.15], and a causal model and KPIs are proposed to discover the relationship between work instructions and production (see Figure 15.3.). They propose the following classification of the functions of a WIS:

- Functions of perception, such as: documentation, monitoring and context recognition,

- Functions of instructions for known processes, such as: initiation of processes, information output, learning and practicing,
- Functions of instructions for unknown processes, such as: decision-making support, communication.

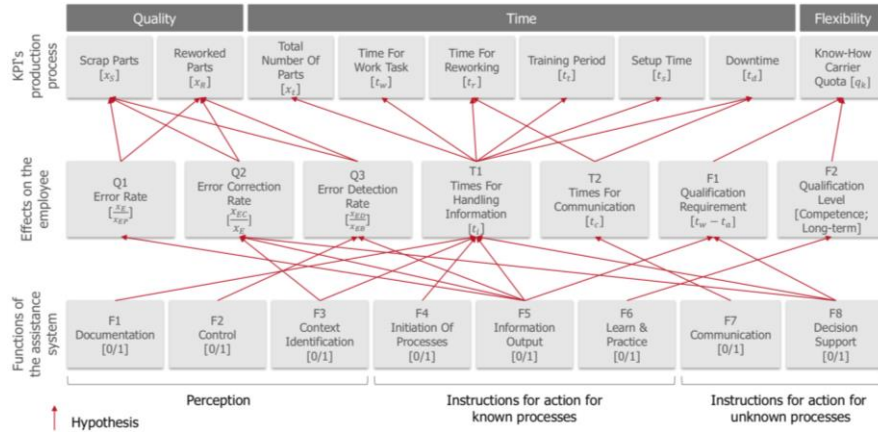


Figure 15.3. A causal model for analysing the effects of worker assistance on production processes [15.15]

Morioka and Sakakibara [15.16] identified that WISs (by providing skill-dependent and multimedia content) have key impact on shortening the learning and training phase for workers. They proposed a system for extracting expert knowledge and transferring it as multimodal learning materials to novice workers (see Figure 15.4.). Vernim and Reinhart [15.17] also confirmed that better performance can be achieved by carefully selecting the media to deliver the content.

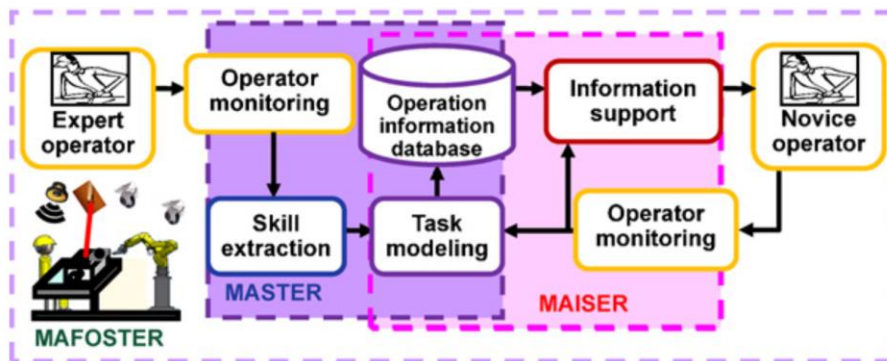


Figure 15.4. Extraction-storage-delivery model for multimodal instruction [15.16]

This review of currently available software solutions and works on 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 [15.17]. The overview also shows that WIS for assembly can provide the necessary support to human workers in flexible production systems. The emerging field of collaborative robotics aims to even further increase the flexibility offered [15.18]. However, as the availability of (skilled) human workforce is often the bottleneck in production systems, successful collaborative applications require efficiency from both human and robot actors [15.16].

Nevertheless, 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 multimodal interfaces.

15.3 Problem statement

Assuming that human–robot collaboration (HRC) takes 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 WIS 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, and
- Providing multimodal bidirectional communication to and from the human worker, with the main goal of delivering content according to the specifications of process engineering.

It is also assumed 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, and
- A Mobile Worker Identification (MWI) system for locating human workers across the shop-floor and within the workcell, also transmitting this information towards the cockpit.

The operation of a collaborative WIS is (at least) twofold: content is created and the interfaces of the system are defined *offline* (i.e., not during production), while content delivery and the interactions with system take place during production, when the system is *online*. These two basic use cases define two basic user types (roles) in

a collaborative WIS: the *process engineer* (PE) who authors the content offline, and the *operator* who interacts with the system, typically when it is online.

15.4 Data model

The general concept of providing the worker with a wearable mobile assistance suite is guided by the recognition that wearable components do not necessarily need to implement autonomous or smart behaviour. The most important function of the worker assistance system is to provide multimodal 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 business 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 15.5 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 Cockpit and the UC. (2) The *frontend* provides the necessary interfaces towards the devices of the wearable assistance suite.

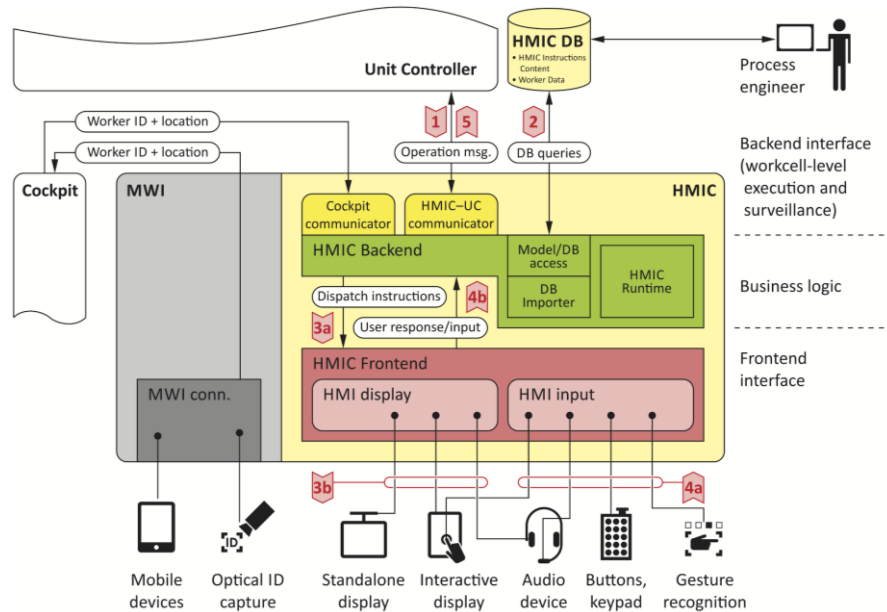


Figure 15.5. Overview of the HMIC system

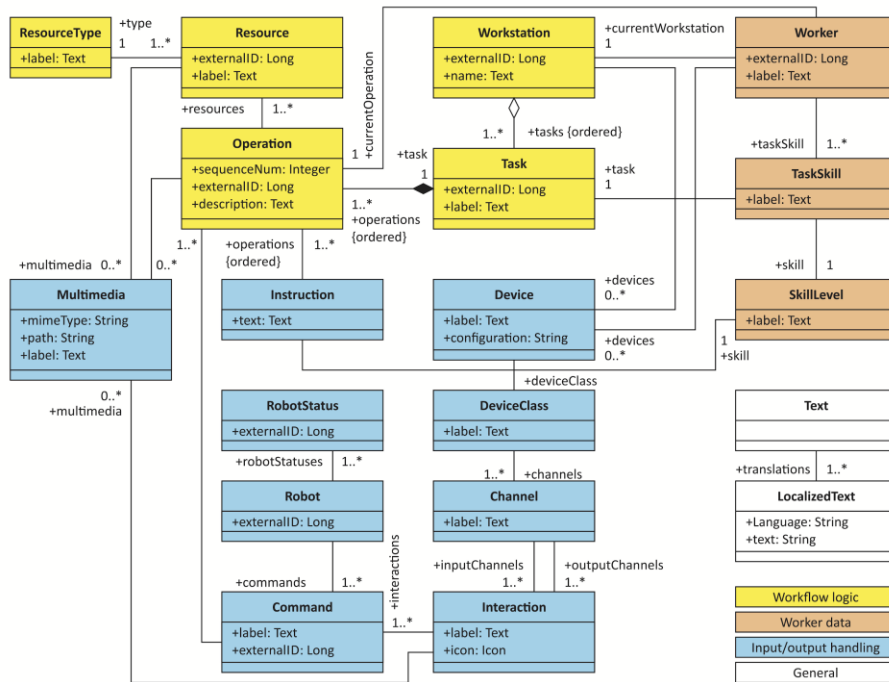


Figure 15.6. Data model of the HMIC WIS

The HMIC backend is coupled with a database (HMIC DB, see Figure 15.6.) that stores the context data and all the relevant information delivered to the worker through the assistance suite. These data are queried during execution-time content generation by the backend before delivery, in order to have the delivered information fit (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 cover 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. 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.

15.5 Context definitions and communication with the production ecosystem

15.5.1 Contexts of process execution

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:

- *Process context*: tasks and operations are originally specified externally, then managed and distributed to the relevant workcells by the Cockpit. Upon execution of a predefined task, the UC of the workcell notifies HMIC about the task to be executed and the assigned worker.
- *Worker properties context*: skills, abilities, and preferences of the worker. This information, stored in the worker database, has to be taken into account during instruction delivery in order to ensure that the instructions displayed match the requirements 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 HMIC instance belonging to the affected workcells. The location within the workcell also enables delivering location-aware instructions (e.g., increasing font size or volume when the worker is further away from the devices).
- *Worker activity context*: actions performed by the workers 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 HMIC, which also interprets the commands (e.g., acknowledgement of a task, failure, etc.), and forwards the commands to the UC.

15.5.2 Information flow

When an operation is to be performed in the workcell, the UC sends a request for displaying the adequate instructions to the worker. The content, the delivery mode, and the formatting of the instruction is selected in accordance with the current process execution context, including the operation and worker identities (process context), worker skill level and individual preferences (worker properties context), and the precise location of the worker within the workcell (worker location context). Then, the instructions are delivered to the selected HMI display device. Upon termination of the operation, the worker responds with one of the predefined status responses for the task, typically, a success or failure response. This action is matched with the messages valid in the given context, and the selected message is passed on to the UC. The UC then determines the next action to perform, in view of the outcome of the preceding operation.

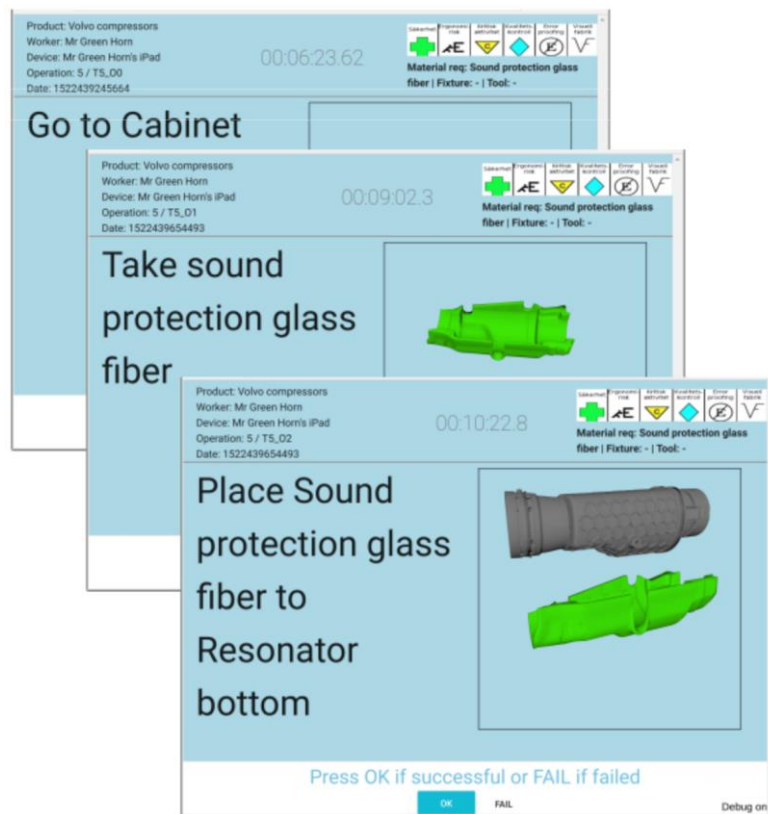


Figure 15.7. Examples of instruction generated to a “novice” worker

Figure 15.7 shows a series of detailed work instructions displayed to a novice worker. The status responses available for the operation are displayed at the bottom of the screen.

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. A confirmation is sent to the UC instead, however, this only means that the message was successfully displayed.

Similarly, 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 operation, 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 recognised, the command is sent to the UC.

15.6 Multimodal interaction

In order to present work instruction to the human worker with the modality (or multiple modalities) that suits the given situation the best, visual (text, image, video, or 3D animation) and audio modalities are used. In addition, multimodal human-to-machine commands allow the human worker to submit commands to the robot and to the assembly system in various novel contactless (e.g., hand or body gestures, or audio commands) and traditional contact-based modalities, for which the implementation is discussed below.

15.6.1 Hand gestures

Hand gesture recognition is an effective and easy way for human workers to interact with a WIS. Hand gesture communication is fast, natural and easy to learn.

The wearable gesture recognition device is a glove with sensors that is capable of measuring the momentary state of the worker's hand, recognises previously recorded gestures and sends the corresponding command that is paired with it. The Gesture Glove interface is mobile: while the worker wears it, there is no need to be close to a button or touch screen. It is ideal to operate in noisy environments where voice communication would not work. It also works in occlusion or disadvantageous lighting conditions which are not ideal for visual communication.

The gestures produced by the glove and interpreted by the gesture interpreter can be used to confirm the end of tasks (e.g., when a task is successfully finished by the

worker), or can be used to control robots either to start/stop the robot or instruct the robot's movements.

The Gesture Recognition device has three main parts: a glove with sensors, a microcontroller unit and a client application. The Gesture Glove uses 12 inertial measurement units (IMU): two on each finger, one on the hand and one on the wrist. These IMUs contain three 3-axis microelectromechanical sensors (MEMS): a 3-axis gyroscope, a 3-axis accelerometer and a 3-axis magnetometer. The gyroscope measures rotation, but the signal integration causes a drift which must be corrected. For drift correction, the accelerometer and magnetometer are used. The accelerometer measures gravity, and the magnetometer measures the Earth's magnetic field. Because the accelerometer and magnetometer are slow and noisy, we cannot rely on them only, so that all three sensors are needed for correct spatial orientation measurement.

15.6.2 Body gestures

Gesture recognition using body gestures is a convenient way of communication in situations when buttons or other more sophisticated interfaces are not available. It has the advantage that the operator needs no separate equipment to wear or operate, as the necessary sensors are all part of the working environment.

To have an effective body gesture recognition system in place, there must be a way to constantly observe the whole body of the operator. It may not be possible all the time because of occlusion, be it workbenches, other workers, robot movements or just the self-hiding nature of some movements. It is, therefore, better to define redundant and unambiguous gestures. Although recognition of gestures is one crucial aspect, it is also essential that these gestures should not look akin to gestures that are regularly used by the worker during assembly operation. Failure to fulfill the latter requirement may lead to misdetection of gestures, which is unacceptable.

Example gestures used during the implementation are standing in an upright position, facing a sensor and holding both hands straight to the side. This way the operator signals that the operation is finished, basically sending the command OK to the system. Standing upright and keeping one arm to the side and another high up in the air may signal that the operator had some trouble during the current operation, and thus sends the FAIL command.

In the example presented here, a Microsoft Kinect v2 was selected as a depth camera. For depth measurement, the Kinect v2 camera utilises time-of-flight sensors, but the camera also features RGB sensors. During implementation, the Microsoft Kinect SDK, version 1409 was used. This SDK is the reference implementation used in association with the Kinect sensor, however, it is only available on Microsoft Windows platform. The recognition software had a dictionary-like configuration file, containing gesture names and the associated commands that are to be sent upon gesture recognition. For gesture definition, the official Microsoft Visual Gesture Builder was used. Although there were other gesture proposals beside the ones mentioned above, many had to be dismissed because either the SDK could not handle them firmly, and therefore recognition quality was poor, or they were easily confused with regular movements of the operator.

15.6.3 Voice commands

Voice based interaction can be useful in cases when workers operate in a moderately noisy environment and it is easier for them to say commands out loud rather than use hand gestures or touch screens.

Voice recognition works using a hot word (e.g., “Symbiotic”). When the voice-to-text engine recognises the hot word, it starts to interpret the subsequent words as commands to the system.

For each work instruction, one can define voice commands, i.e., the words expected to be said out loud by the worker for the purpose of communication with the robot. These voice commands, once interpreted, are converted to actual commands that are also used in other interaction modes to send information back from the worker to the system.

For example, when the worker is finished with the operation, he/she can finalise the work by responding with an OK command or signal; whereas a FAILED command can signalise a failure. In case of voice interaction, the worker can say the hot word + OK (e.g., “Symbiotic OK”) to generate an OK command, or the hot word + FAIL (e.g., “Symbiotic Fail”) to generate a FAIL command.

In the same way, when the worker controls the robot in a specific situation, saying the hot word + START ROBOT (e.g., “Symbiotic Start robot”) lets the robot do its operations and send a ROBOT_START command, or it can be stopped with a hot word + STOP ROBOT (e.g., “Symbiotic Stop robot”) to send a ROBOT_STOP command.

For text recognition, a possible solution is to use Google’s text-to-speech engine available in the Chrome browser, which makes voice recognition available on any device capable of running a recent Chrome browser. Google’s text to speech is of high quality and allows voice recognition without any training by the speaker and allows voice commands in practically any language.

It was found that voice recognition works best if the worker uses a headset with a good quality microphone. This can also enhance voice recognition when working in a noisy or crowded environment.

15.7 Mobile worker identification and tracking

An important component of the proposed worker assistance suite, which provides the worker location context for the HMIC, is Mobile Worker Identification (MWI). Worker identification, localisation, and tracking methods based on sensor information are crucial for efficient HRC. Nevertheless, the adequate technology and the required functionalities strongly depend on the given application environment, such as the area to cover (complete shop-floor or single workcell) or the desired accuracy.

In the current application, the goal was to cover a limited area on the shop-floor close to the robot and the conveyor system, with an accuracy requirement below 0.5 metres. Additional orientation information (the direction the human worker faces) was also desired. Finally, the MWI system developed had to be easy-to-integrate and easy-to-set up, cost effective, and scalable, with a possibility to add, remove or rearrange the sensors in case of modifying the area covered. These requirements could be satisfied with an area-scan camera system that tracks workers’ motion with the help of QR codes attached to the workers’ clothing. Since working with an industrial robot

requires wearing a protective helmet, attaching the QR codes to the helmet is proposed, see Figure 15.8. QR codes ensure fast readability and greater storage capacity compared to standard barcodes. The reading of QR codes can be robust, also when codes move. A part of the robustness of QR codes in the physical environment is their possibility of error correction—if this option is utilised, they continue to be correctly readable even when a part of the QR code image is obscured, defaced or removed.

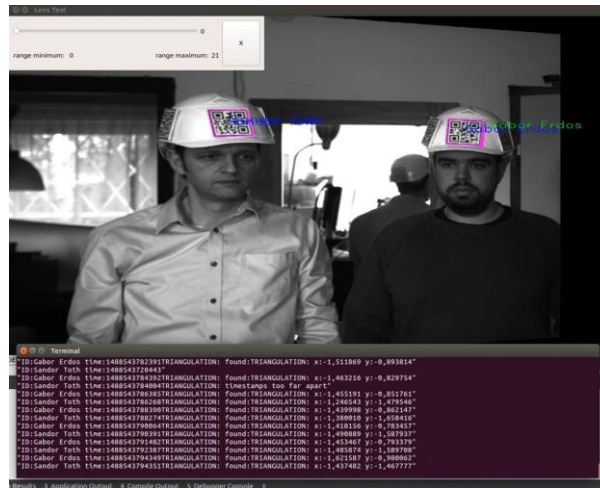


Figure 15.8. Workers identified by MWI based on QR codes attached to their helmets



Figure 15.9. Smart camera node with a Raspberry Pi 3 computer and an IDS XS2 camera

The requirement of covering a large area implies that a multi-camera system, consisting of low-cost, easy-to-integrate camera nodes is needed, where each node is composed of local computing devices and cameras with suitable short-range optics. For the particular application, Raspberry Pi 3 single-board computers with IDS XS2 cameras (range: 0.5–2.2 m) as shown in Figure 15.9, and INTEL NUC i3 computers with DS UI3540LE cameras and TAMRON M12VM412 optics (range: 1.3–3 m) were used. Positions, orientations, and view angles of the camera nodes are established at the time of calibrating the system. Figure 15.10 illustrates one experimental setup with 7 smart camera nodes for MWI in a robotic work cell.

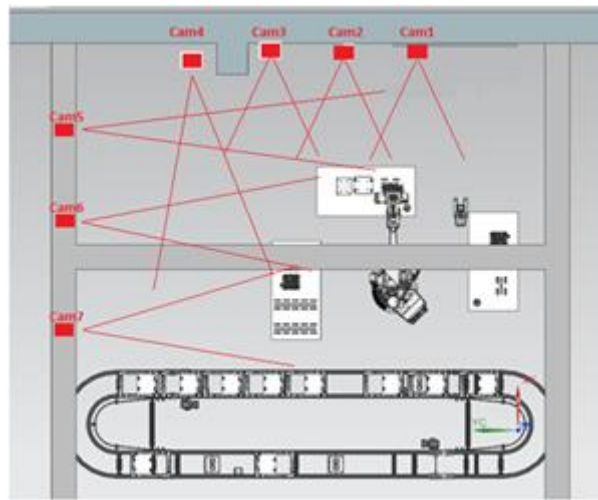


Fig. 15.10. Experimental setup with 7 smart camera nodes for MWI

The captured images are scanned for QR codes (where multiple QR codes in a single image are allowed), and the code positions in the image plane are determined. Possible perspective distortions of the QR code are corrected, which is required for the robust reading of QR codes. The exposure time of the cameras is continuously adjusted according to the contrast and intensity parameters of the QR codes. The identified QR codes and their positions are subsequently submitted to a server application via Wi-Fi and TCP/IP communication, which performs the localisation of the human operators using one of the two possible calculation methods: when only one camera sees a given QR code, the position of the code is calculated by the size of the QR code and its position in the image plane. More accurate positioning can be achieved by triangulation when multiple cameras capture the same QR code within a given time window.

With the above physical architecture and computation methods, MWI enables delivering dedicated content to the authenticated workers corresponding to their skill levels, as well as selecting the appropriate display device depending on their position and head orientation in the HMIC.

15.8 Implementation

The HMIC backend was implemented as a *Spring Framework*-based Java web application providing 3 communication interfaces:

- *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 *SocketJS* to handle delivery of instructions and 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 devices including monitors, tablets, mobile phones and AR (Augmented Reality) glasses. It also handles user interactions in the form of button based inputs. The HMIC frontend is implemented as a *NodeJS* application with *React*-based user interface.

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 [15.19]). The content was prepared and delivered to the system via XML files structured according to the HMIC DB schema. In order to test the multimodal interfaces, the following devices were connected to the system: a large-screen tablet device, a smart phone, a gesture glove, a Kinect device and a pair of AR glasses. The devices with HTML5 web browsers were registered directly in the frontend of the system, while those without web browsers were connected via their client applications developed, which 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 simultaneous communication channels. The assembly tasks and the worker identification signals were issued by using mock interfaces of a UC and a Cockpit. The system was able to handle inputs from, and to deliver content to the multiple devices with an adequate response time.

15.9 Conclusions

Human–robot collaboration in industrial production relies heavily on efficient and dependable communication between human and robotic resources. This is not only due to the fact that the interlinking of robots and humans with variable cognitive capabilities and skill levels needs appropriate information exchange, but also due to

the high operation diversity in production areas where collaborative settings have the highest expected benefits (i.e., cases with large product variability and operations that often need a “human touch”). Trends and expectations outline the need for a worker assistance system that goes beyond today’s prevailing practice of push-buttons and paper-based work instruction sheets by (1) enabling dynamic changes to the instruction content, (2) extending static content with more expressive multimedia elements (video, audio, virtual or augmented reality), (3) integrating instruction delivery and machine-to-human signalling with human-to-machine communication, and (4) integrating the context-awareness of production control (including worker identification) with adaptation to individual user characteristics such as skill level, current mental/physical state, or preferences in mode of communication.

The chapter explained a possible solution for a worker assistance system by presenting a pilot implementation integrated into a proposed multi-level production control architecture, and offering possibilities of adaptive content generation, selection of multiple communication modes, and recognition of current working context. The chapter presented selected technologies supporting communication over a variety of channels establishing a multi-modal character of the suite, and discussed advantages and drawbacks of each principle. In addition, a possible architecture of a worker assistance suite was presented, which was validated in a pilot implementation representing an assembly scenario from the automotive industry.

The development of underlying technologies and changes in industrial production practice make it probable that the use of integrated worker assistance suites experiences considerable growth, especially in cases where current instruction and feedback practices are known to be a hindrance to work efficiency. Worker assistance systems have close dependencies to both process planning (preparing content offline), and process execution (deploying content with production being online). Therefore, the maturing of the multi-modal communication paradigm is anticipated to bring about supporting technologies enabling routine coupling with existing solutions and technologies in production planning and control.

References

- [15.1] L. Wang, R.X. Gao, J. Váncza, J. Krüger, X.V. Wang, S. Makris and G. Chryssolouris, “Symbiotic Human–Robot Collaborative Assembly,” *CIRP Annals—Manufacturing Technology*, 68(2), pp. 701–726, 2019.
- [15.2] Cs. Kardos, Zs. Kemény, A. Kovács, B.E. Pataki, and J. Váncza, “Context-dependent multimodal communication in human–robot collaboration,” *Procedia CIRP*, vol. 72, pp. 15–20, Jan. 2018.
- [15.3] Cs. Kardos, A. Kovács, B.E. Pataki, and J. Váncza, “Generating Human Work Instructions from Assembly Plans,” in *UIISP 2018: Proceedings of the 2nd Workshop on User Interfaces and Scheduling and Planning*, Delft, the Netherlands, 2018, pp. 31–40.
- [15.4] M.C. Leu, H.A. ElMaraghy, A.Y. Nee, S.K. Ong, M. Lanzetta, M. Putz, W. Zhu, A. Bernard., “CAD model based virtual assembly simulation, planning and training,” *CIRP Annals*, vol. 62, no. 2, pp. 799–822, 2013.
- [15.5] M. Lušić, K. Schmutzer Braz, S. Wittmann, C. Fischer, R. Hornfeck, and J. Franke, “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*, vol. 1018, pp. 23–30, Sep. 2014.
- [15.6] M. Lušić, C. Fischer, J. Bönig, R. Hornfeck, and J. Franke, “Worker Information Systems: State of the Art and Guideline for Selection under Consideration of Company Specific Boundary Conditions,” *Procedia CIRP*, vol. 41, no. Supplement C, pp. 1113–1118, Jan. 2016.
- [15.7] Å. Fast-Berglund, T. Fässberg, F. Hellman, A. Davidsson, and J. Stahre, “Relations between complexity, quality and cognitive automation in mixed-model assembly,” *Journal of Manufacturing Systems*, vol. 32, no. 3, pp. 449–455, Jul. 2013.
- [15.8] E. Hollnagel, “Information and reasoning in intelligent decision support systems,” *International Journal of Man–Machine Studies*, vol. 27, no. 5, pp. 665–678, Nov. 1987.
- [15.9] K. Alexopoulos, S. Makris, V. Xanthakis, K. Sipsas, A. Liapis, and G. Chryssolouris, “Towards a Role-centric and Context-aware Information Distribution System for Manufacturing,” *Procedia CIRP*, vol. 25, no. Supplement C, pp. 377–384, Jan. 2014.
- [15.10] A. Claeys, S. Hoedt, H. V. Landeghem, and J. Cottyn, “Generic Model for Managing Context-Aware Assembly Instructions,” *IFAC-PapersOnLine*, vol. 49, no. 12, pp. 1181–1186, Jan. 2016.
- [15.11] S. Bader and M. Aehnelt, “Tracking Assembly Processes and Providing Assistance in Smart Factories,” in *Proceedings of the 6th International Conference on Agents and Artificial Intelligence (ICAART)*, Angers, France, 2014, pp. 161–168.
- [15.12] A. Bannat, F. Wallhoff, G. Rigoll, F. Friesdorf, H. Bubb, S. Stork, H.J. Müller, A. Schubö, M. Wiesbeck, M.F. Zäh, “Towards optimal worker assistance: a framework for adaptive selection and presentation of assembly instructions,” in *Proceedings of the 1st international workshop on cognition for technical systems, Cotesys*, 2008.
- [15.13] M. Fiorentino, A.E. Uva, M. Gattullo, S. Debernardis, and G. Monno, “Augmented reality on large screen for interactive maintenance instructions,” *Computers in Industry*, vol. 65, no. 2, pp. 270–278, Feb. 2014.
- [15.14] J.A. Erkoyuncu, I.F. del Amo, M. Dalle Mura, R. Roy, and G. Dini, “Improving efficiency of industrial maintenance with context aware adaptive authoring in augmented reality,” *CIRP Annals*, vol. 66, no. 1, pp. 465–468, 2017.
- [15.15] T. Keller, C. Bayer, P. Bausch, and J. Metternich, “Benefit evaluation of digital assistance systems for assembly workstations,” *Procedia CIRP*, vol. 81, pp. 441–446, 2019.
- [15.16] M. Morioka and S. Sakakibara, “A new cell production assembly system with human–robot cooperation,” *CIRP Annals*, vol. 59, no. 1, pp. 9–12, Jan. 2010.
- [15.17] S. Vernim and G. Reinhart, “Usage Frequency and User-Friendliness of Mobile Devices in Assembly,” *Procedia CIRP*, vol. 57, no. Supplement C, pp. 510–515, Jan. 2016.
- [15.18] G. Michalos, S. Makris, J. Spiliotopoulos, I. Misios, P. Tsarouchi, and G. Chryssolouris, “ROBO-PARTNER: Seamless Human–Robot Cooperation for Intelligent, Flexible and Safe Operations in the Assembly Factories of the Future,” *Procedia CIRP*, vol. 23, no. Supplement C, pp. 71–76, Jan. 2014.
- [15.19] Cs. Kardos, A. Kovács, and J. Váncza, “Decomposition approach to optimal feature-based assembly planning,” *CIRP Annals*, vol. 66, no. 1, pp. 417–420, 2017.