Design of a Cognitive Assistant System for Industrial Maintenance Tasks Using Augmented Reality

Ruben Schlagowski
University of Applied Sciences Augsburg
Augsburg, Germany
ruben.schlagowski@hs-augsburg.de

Lukas Merkel
Fraunhofer IGCV
Augsburg, Germnany
lukas.merkel@igcv.fraunhofer.de

***Abstract* — Work tasks in future smart factories will increase in complexity and demand a higher level of expertise from workers. Specialized assistant systems may help them to fulfill the associated requirements. This paper gives insight into a research project examining the potential of augmented reality headsets and a rule-based cognitive architecture as components for such a system, which shall assist machine operators while conducting maintenance tasks in an industrial environment. The prototype is able to connect to a production facility and interpret the situation based on the gathered machine data by using a rule-based cognitive architecture. Based on this data it will plan actions which can be executed either by the system itself, or by the machine operator. A state-of-the-art augmented reality headset is used for the user interactions. It will be able to show where an action needs to be executed by the operator by placing holograms into the user's environment.**

Keywords—Assistant System, Production, Maintenance, Augmented Reality

I. Introduction

One of the major visions of Industrie 4.0 is the increased customization of products [1, p. 5]. As industrial machines and production lines become smarter to produce smaller production lots, workers need to keep up with their expertise. Maintenance workers in particular will need to become more flexible, e.g. by updating their knowledge about machinery that needs to be supervised and maintained at a higher frequency [2, pp. 22-33]. To be able to manage the workload arising from these transitions, maintenance tasks need to be planned and executed smarter. In contrast to ordinary static maintenance intervals that are commonly prescribed by the manufacturer of the machine, a more flexible and intelligent maintenance schedule can reduce downtime of the production line and hours of work. The concept of predictive maintenance suggests that this can be achieved by predicting the future conditions of the machine by performing a real-time analysis of gathered machine data [3, p. 197].

The goal of a research project launched as a cooperation of Fraunhofer IGCV and the Faculty of Electrical Engineering of the University of Applied Sciences Augsburg in 2016, is to develop an assistant system to serve as an appropriate tool to provide information as needed, decrease training time and assist workers with the execution of maintenance tasks. This system should not only provide instructions to the worker, but also gather data from the machinery and interprete its state. It will also be able to manipulate machinery directly if it is required and appropriate.

II. Methodology

The assistant system is developed by following the human-centered design process (cf. Fig. 1), which is described in [4]. A significant part of this process is to understand and specify the context of use. Based on the knowledge gained during this phase, the developer can specify user requirements for the system and design system solutions that aim to fit the specification. Subsequently, the system is evaluated to check whether all user requirements are met. If this is not the case, further iterations of the process are performed.

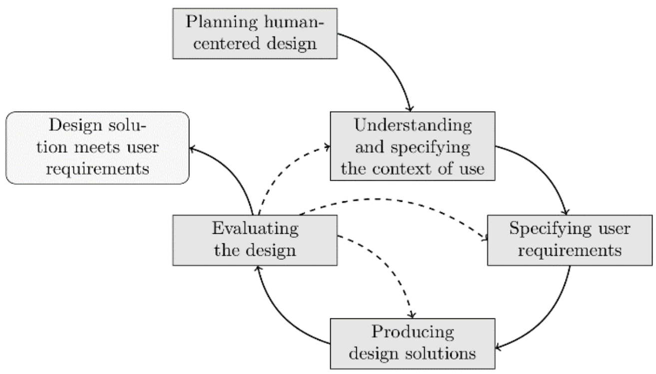

Fig. 1. Human centered design process cf. [4]

For the project described in this paper, the context of use is specified by examining the parts of a particular work system namely the maintenance of a production line, which assembles cubes out of two half-cubes and two pins. The design solution is implemented as a laboratory prototype consisting of hardware and software components, which will be evaluated by conducting user studies. This prototype will be able to assist a worker with the execution of the mentioned maintenance tasks The worker will not need any specific know-how with respect to the handling of this machinery as all the information that he needs to maintain the facility will be communicated by using a state-of-the-art head mounted display (HMD). Moreover, this prototype is able to influence the production facility autonomously.

III. RESULTS

A. Analysis of the Context of Use

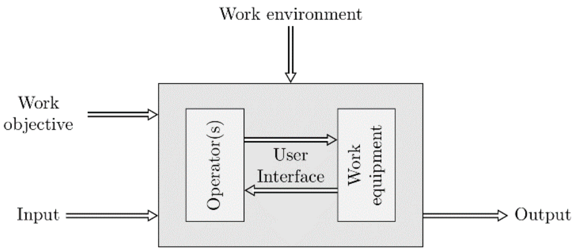

Fig. 2. Illustration of a work system cf. [5]

The context of use is analyzed by examining the components of the work system. It consists of one or multiple *operator(s)* using *work equipment* in a specific *work environment* to transform a given *input* into an *output* which is defined by the *work objective* [5, p. 6].

For the use case analyzed in this study, the operator's work objective is to reach and maintain the operational state of a mechatronic training system (cf. Fig. 3) consisting of three stations. Station 1 provides half cubes from two magazines and checks a variety of their properties such as material or orientation by using sensors on a conveyor belt. Station 2 combines two of these half-cubes together by using two metal pins. Finally, Station 3 places these complete cubes into stock. The operator is a maintenance worker without any procedural knowledge about the system that needs to be maintained. He has basic knowledge about handling tools and conducting typical maintenance work. The environment is a learning factory for cyber-physical production (Abbreviation: LVP, operated by Fraunhofer IGCV in Augsburg, Germany).

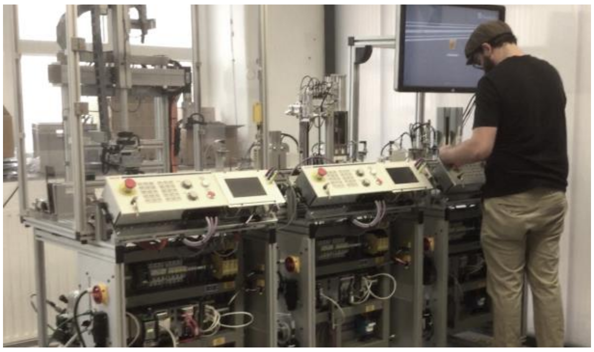

Fig. 3. Operator maintaining the production line in the learning factory

The *target state* of the work system, which is defined to be the operational state of the production line by the work objective, can be reached from a variety of other states that can be defined through combinations of machine properties. These states can be determined by gathering in-depth knowledge about the work objective (e.g. through self-observation while operating the machine) and the work equipment, which in this case mainly consists of the production line itself. Figure 4 displays this state (TS for target state) together with a variety of states that can occur as well, such as sub-goals (SG), an error-state (ES) and multiple transitions between these states.

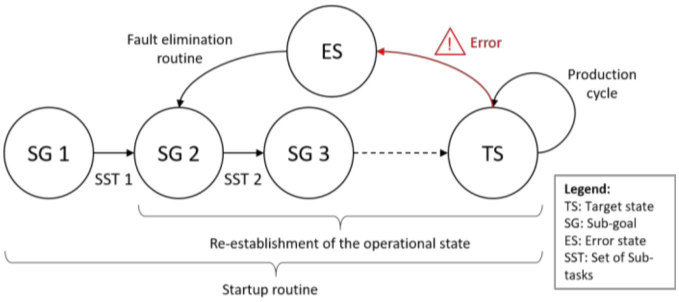

Fig. 4. Startup routine and elimination of fault as transitions between states

If the initial state is a completely turned off production line, a *startup routine* (cf. Fig. 4) needs to be executed to reach the operational state of the production line. The startup routine can be divided into distinct sub-states or sub-goals. A transition between these states can be accomplished by executing distinct action patterns, which may repeat within a variety of typical maintenance procedures, so called *sub-tasks* [6]. The execution of a distinct set of sub-tasks (cf. Fig. 4: SSTs) causes a transition to a new state of the work system, a *sub-goal*. All sub-goals need to be technically identifiable to facilitate an interpretation of the prevailing situation by the assistant system. Table I shows an excerpt of described sub-goals for the specific use-case together with the sub-tasks that are needed to reach these states and characteristics which can be used to verify that the state has been reached.

TABLE I: SUB-GOALS AND LINKED SUB-TASKS OF THE USE-CASE (EXCERPT)

Sub-goal	Set of sub-tasks	Characteristics
SG 1: PLCs are online	3x press main switch, wait until PLCs have booted	Connection to PLCs ocer TCP is possible
SG 2: Air pressure is up	turn on compressor, wait until air pressure is up	Air pressure relays sensors are "true"
SG 3: All stations homed	3x rotate key, 3x press switch	Homed variables in PLCs are "true"
SG 4: Magazines refilled	2x empty stock, 4x refill magazine	Sensors in magazines are "true", operator's acknowledgement

In case an error occurs, the production cycle may come to an hold and a new state is reached, an *error state* (cf. Figure 4). Error states differ according to the various errors that might occur during production. To get the system back to the target state again, usually two routines need to be executed:

1. *Fault elimination routine:* A sequence of sub-tasks that eliminate the fault(s) that were caused by the error. The final state is a sub-goal of the startup routine.
2. *Re-establishment of the operational state:* A partial startup routine. The purpose of this routine is to reach the target state again by conducting a subset of the sub-tasks of the startup routine.

Similar to the sub-goals, all possible error states need to be determined and described along with their fault elimination routines, appearances, characteristics and follow-up states. Table II displays this information for two error states of the example use-case.

TABLE II: ERROR STATES AND FAULT ELIMINATION ROUTINES (EXCERPT)

Error State	**Appearance**	**Characteristics of Error State**	**Fault elimination routine**	**Follow-up sub-goal**
Lack of air pressure	Supply of half-cubes stops	At least one pressure relay sensor "false"	Increase pressure on compressor	SG2
Emptying of stock is not registered	Stock empty, cubes accumulate on station 3	Stock capacity PLC variable is maxed out	Reset of the stock capacity variable	SG3

Besides this state-analysis of the work objective and work equipment, additional knowledge needs to be gathered about the operator(s) and the work environment to achieve a complete understanding of the given work system. Both might differ significantly from the scenario in which the system is developed, e.g. in qualifications and knowledge of the operator(s) or in environmental conditions such as noise levels, illumination and atmospheric influences.

B. Specification of User Requirements

According to [4, pp. 17-18], the specification of the user requirements must contain

A) "the intended context of use",

B) "requirements that can be derived from the user and the context of use",

C) "requirements, that are based on relevant Knowledge about ergonomics and user interfaces, such as norms and guidelines",

D) "requirements and goals for usability, including measurable criteria for performance and user satisfaction in certain contexts of use" and

E) "requirements that can be derived from organizational requirements that influence the user directly."

The first iterations of the human centred design process (cf. chapter II) of this study as described in [6] primarily focused on user requirements that address the operator and work objective and are related to the context of use and usability and performance goals. These are:

- The intuitiveness of interaction, quality of the operator's perception of information and attention should be considered.
- The user's mobility should not be constrained by the system. Furthermore, his hands should be kept free.
- The operator's learning efficiency shall increase without decreasing his level of expertise and reliability.

In further iterations of the design process, the design shall be refined to fulfill requirements regarding ergonomics and user interfaces such as:

- The assistant system shall consider the user's expertise and procedural knowledge, mental and physical workload.
- The operator shall be able to perceive information and interact with the system while working simultaneously by stressing the appropriate human resources according to the situation. [7, p. 163]

For all the user requirements that are chosen for the system, measurable target values or verifiable standards need to be added, such as "The average learning time must decrease by 10%" or "In terms of ergonomics, the standard ISO 9241-110 should not be violated".

C. Design Solution: Concept and Design Principles

To be able to cover the user requirements as specified in chapter B, we design a *cognitive assistant system*. A system of this kind assists the operator to achieve the given work objective by

1. drawing the operator's attention towards the most important task,
2. ensuring, that the mental workload of the operator is neither too high nor too low and
3. possibly executing actions for the operator by itself. [8]

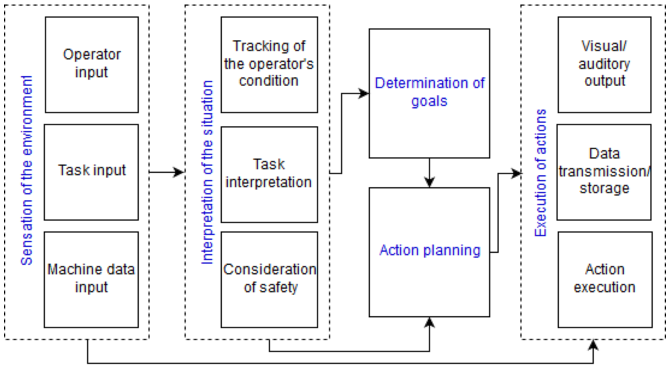

Fig. 5. Architecture of the assistant system [6]

An architecture as shown in Fig. 4 is used to enable the system to achieve the necessary situational awareness, which is required to make appropriate decisions based on the operator's current situation. The components are based on the typical states of human information processing as described in [8] and [9]. These are

- *Sensation of the Environment*: Gathers data about the work system and its current state. This includes the operator, the machinery and the work environment.
- *Interpretation of the Situation*: This component compares the data about the environment with known states of the work system such as the target state, sub-goals and error states to interpret the current situation.
- *Determination of Goals*: Based on the situation, the goals are determined in this section. For the example use-case, the default goal will be to get the production line into the target/operational state. This goal temporarily be replaced by a more important one, e.g. if

the operator's safety needs to be ensured or a higher prioritized maintenance task becomes due.

- *Action Planning*: In this component, the actions are chosen that should be executed to fulfill the work objective. It is also planned whether the system should execute actions by itself or if the operator should be advised to take action. The used codes and modalities for user interaction are also planned in this section.
- *Execution of Actions*: This part realizes the planned actions. Possible actions are the communication of instructions to the operator or an action of the system itself which influences the environment directly.

D. Design Solution: Prototype

The prototype consists of multiple hardware and software components (cf. Fig. 6). For all user interaction, the Augmented Reality headset *HoloLens* is used. It runs an application, which was developed in the *Unity 3D* engine. The user can choose to interact with the device by using speech or gesture inputs. Information can be presented to the user by displaying text and holograms on the integrated optical see-through displays (cf. Fig. 8).

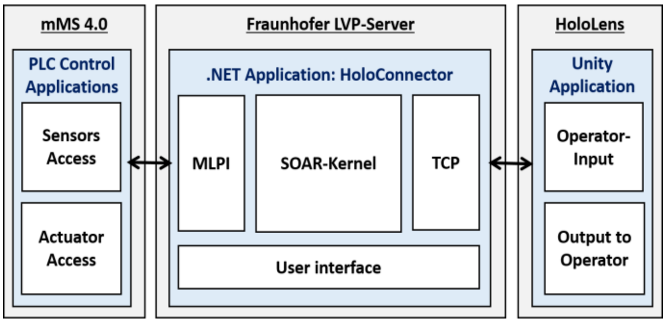

Fig. 6. Hard- and software components of the prototype

The .NET application *HoloConnector* is responsible for the realisation of assistant functions that deal with the interpretation of the situation, goal determination and the planning of actions. Moreover, the application realizes all communications with the environment such as the production line (mechatronical training system by Bosch Rexroth, model mMS 4.0) and the operator using the head-mounted device. It uses TCP/IP to connect to the HoloLens and a *motion logic programming interface* (MLPI) to connect to the three PLCs of the production line. These interfaces enable the system to perceive the environment (cf. Figure 5: *Sensation of the Environment*) and influence the environment by executing actions (cf. Fig. 5: *Execution of Actions*) e.g. by reading and writing on symbolic variables on the PLC control applications and by presenting information to the user wearing the HoloLens headset.

The architecture components *Interpretation of the Situation*, *Determination of Goals* and *Action Planning* (cf. Fig. 5) are implemented by using a kernel of the cognitive architecture Soar [10]. It represents the current state as a working memory graph. This working memory can be modified by the application of rules that contain knowledge. The kernel is connected to the TCP and MLPI interfaces through a special set of working memory elements (WMEs) called input-link and output-link.

The input-link of the kernel is updated as soon as a change in the environment is detected. Subsequently, the Soar kernel is run (cf. Fig. 7). It compares the gathered data about the environment with known states such as sub-goals and error states (cf. Chapter *A. Analysis of the Context of Use)* to interprete the situation. Based on this knowledge it plans actions, including details on how they should be executed. If new instructions for actions have been generated by the kernel, they are registered on the ouput-link and transmitted to the HoloLens application in case the user needs to take action. If they can be executed by writing onto a symbolic variable over the MLPI interface, they are directly executed. After executing these actions, the production line is furtherly monitored by reading available data from the machinery over the MLPI interface cyclically until another change is detected.

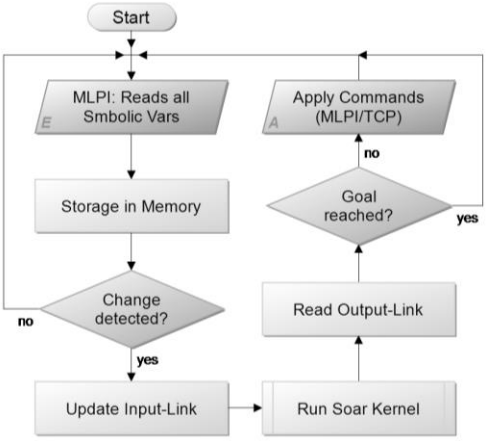

Fig. 7. Flowchart of the HoloConnector application

In an example scenario, a production error might occur due to false orientations of half-cubes in one of the magazines of station 1. This error causes the production line to sort out all of the wrongly oriented half-cubes into an ejection stock with limited capacity. As all sensor changes are detected by the system, the Soar-kernel recognizes the negative sonsor values that represent the half-cubes' orientation on station 1 and assigns them to an error state. It then plans the actions necessary to get the system back into operational state. As the fault elimination routine includes sub-tasks that cannot be executed autonomously, like exchanging half-cubes in the magazines and emptying the ejection stock, instructions are generated and passed to the operator. Figure 8 shows one of these instructions ("Bitte Auswurflage leeren!" (German), English: "Please empty ejection stock!") that is generated by the system. It is displayed statically in the field of view of the operator via the HoloLens headset. Additionally, a hologram (arrow) is placed seemingly above the hardware component to give the operator additional spatial information on the task he needs to execute. In this situation, the system waits until it registeres the successful execution of the fault elimination routine (cf. Chapter A, Fig. 4). Subsequently it would advise the operator to put the production line back into the operational state (cf. Fig.4: *re-establishment of the operational state*).

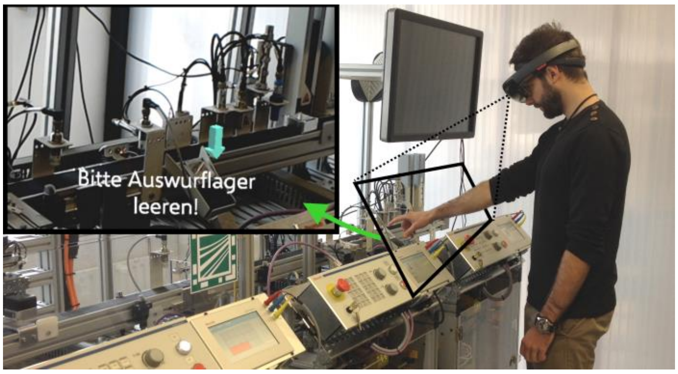

Fig. 8. View of the operator while conducting assisted maintenance work

IV. Discussion

As the rule-based cognitive architecture that is responsible for a large part of the system architecture is not fully implemented yet, more efforts need to be put into the development of the protoype. At this point of time, the basic functionality of the overall system can be confirmed, as all necessary interfaces and structural components of the system have been tested successfully. As soon as we have implemented the rule-based Soar-kernel, user studies need to be conducted to evaluate the overall system. Regarding the used augmented reality headset (HoloLens), there are ergonomic concerns, as it can only be used for short periods of time due to issues with wearing comfort and limited energy capacity. A solution to this problem could be the additional integration of different hardware such as smart watches into the system. These could then be worn over longer periods of time and be used to notify the operator in case maintenance work needs to done. Subsequently the headset can be used to assist the operator throughout the maintenance procedure itself while maintaining the production facility.

V. Conclusion and Future Work

This paper gives insight into the current state of the ongoing research project that aims to develop a cognitive assistant system for industrial maintenance tasks using augmented reality. It shows an approach on how to analyze the context of use and specify user requirements for a system that is meant to assist a user while conducting maintenance work on a production line. Subsequently, the design solution was described, including both the system architecture and the implementation in form of a prototype using a state-of-the-art augmented reality head-mounted display to interact with the operator. In the future, additional interfaces can be implemented into the system to extend the knowledge base of the assistant system. For instance, an augmented reality content authoring tool could provide a user-friendly way of teaching the system additional knowledge e.g. about other machines and production lines. Furthermore, a user recognition interface shall be implemented to enable the system to consider the user's expertise.

Acknowledgment

We thank the Bavarian Ministry of Economic Affairs and Media, Energy and Technology for funding the project 'Digitalisierte Produktion - Demonstrations- und Trainings-zentrum für Cyber-Physische Produktionssysteme' within the program 'Bayern Digital'.

References

[1] A. Roth, Ed., "Einführung und Umsetzung von Industrie 4.0," Berlin, Heidelberg: Springer Berlin Heidelberg, 2016.

[2] W. Mülder, "Arbeit 4.0: Smarte Maschinen für Smart Workers?," Wissensmanagement, no. 4, pp. 22–23, 2016.

[3] U. Kumar, A. Ahmadi, A. K. Verma, and P. Varde, "Current Trends in Reliability, Availability, Maintainability and Safety," Cham: Springer International Publishing, 2016.

[4] International Organization for Standardization (ISO), "Ergonomics of human-system interaction - Part 210: Human-centred design for interactive systems (ISO 9241-210:2010)," 2010.

[5] International Organization for Standardization (ISO), "Ergonomics principles in the design of work systems (ISO 6385:2016)," 2016.

[6] R. Schlagowski, L. Merkel, and C. Meitinger, "Design of an assistant system for industrial maintenance tasks and implementation of a prototype using augmented reality," in 2017 IEEE International Conference on Industrial Engineering & Engineering Management: IEEE IEEM2017 : 10-13 Dec, Singapore, Singapore, 2017, pp. 294–298.

[7] C. D. Wickens, "Multiple resources and performance prediction," Theoretical Issues in Ergonomics Science, vol. 3, no. 2, pp. 159–177, 2002.

[8] R. Onken and A. Schulte, "System-Ergonomic Design of Cognitive Automation: Dual-Mode Cognitive Design of Vehicle Guidance and Control Work Systems," Berlin, Heidelberg: Springer-Verlag Berlin Heidelberg, 2010.

[9] H. Putzer and R. Onken, "COSA: A generic cognitive system architecture based on a cognitive model of human behavior," Cognition, Technology & Work, vol. 5, no. 2, pp. 140–151, 2003.

[10] J. Laird, "The Soar cognitive architecture," Cambridge, Mass, London, England: MIT Press, 2012.