Orchestration and Situation Awareness in an Assistance System for Assembly Tasks

Wernher Behrendt(✉) and Felix Strohmeier

Salzburg Research, Salzburg, Austria
{wernher.behrendt,felix.strohmeier}@salzburgresearch.at

Abstract. We report on the design, specification and implementation of a situation awareness module used for assistive systems in manufacturing, in the context of Industry 4.0. A recent survey of research done in Germany and Europe, concerning assistive technology in industry shows a very high potential for "intelligent assistance" by combining smart sensors, networking and AI. While the state of the art concerning actual technology in industrial use points more towards user-friendly, speech-based interaction with personal assistants for information retrieval (typically of in-house documentation), the research presented here addresses an enterprise-level assistance system that is supported by a number of specialized *Assistance Units* that can be customized to the end users' specifications and that range from tutoring systems to tele-robotics. Key to the approach is situation awareness, which is achieved through a combination of a-priori, task knowledge modelling and dynamic situation assessment on the basis of observation streams coming from sensors, cameras and microphones. The paper describes a working fragment of the industrial task description language and its extensions to cover also the triggering of assistive interventions when the observation modules have sent data that warrants such interventions.

Keywords: Assembly tasks · Situation awareness · Assistance systems · Process language

1 The Concept of Assistance in Industry 4.0

One of the tenets of Industrie 4.0 is the digitization of all processes, assets and artefacts in order to be able to virtualize and simulate production and to achieve maximum flexibility and productivity. While much of the innovation effort is aimed at outright automation, it is also obvious that for the foreseeable future, human-to-machine interaction will remain an important element of any production. With an aging workforce and products that are becoming less standardized (lot-size 1) the notion of assistance has been getting significant attention. In an Austrian "lighthouse-project" the idea of "Assistance-Units" is put forward. This paper presents a taxonomy and internal structure of these units and a proposed formal language to specify their desired interaction with humans when an assistance need arises. It should be noted that the proposed language acts as a mediation layer between higher level business process descriptions (e.g., an order to manufacture

a product) and machine-specific programming constructs that are still needed to operate e.g. a welding robot. The purpose of our task description language is to orchestrate human workers, robotic assistance as well as informational assistance, in order to keep the factory IT in sync with progress on the shop-floor.

1.1 Motivation and State of the Art

While there is plenty of research work on collaborative robotics, exoskeletons, and informational assistance through e.g. virtual reality headsets, less work has been done on bringing these diverse forms of assistance under a common roof. A recent study [1] also took a broad view that included assistance systems for both services and industry. There, the authors distinguish three kinds of (cognitive) assistance: (1) Helper systems providing digital representations of e.g. manuals, teaching videos, repair guides, or other elements of knowledge management. (2) Adaptive Assistance systems that are able to take into account some situational context, e.g. through sensors, and that can then provide relevant information that is adapted to the specific situation. (3) Tutoring Assistance systems are also adaptive, but address explicitly, the need for learning in the work context.

The study forecasts a very high market potential for the use of AI in assistive technology and this is of relevance to the category of "machines and networked systems" which gives the closest fit with manufacturing. The study also lists a number of German research projects broadly addressing digital assistance, in the years 2013–2017.

Looking at the current state of the art in industrial use of assistive technology, there is a prevalence of personal digital assistants based on speech recognition engines such as Apple Siri, Microsoft Cortana, Amazon Alexa, or Google's Assistant with its cloud speech tool. These are often combined with VR or AR headsets, or with tablets or smart phones depending on the use case. It should be noted that any user-activated assistance (e.g. by speech recognition) is *reactive*, i.e. it has no option of pro-actively helping the user. A pro-active approach requires the system to somehow know where worker and assistive machinery are, in the overall task. The following example comes from one of our industrial use cases for which a demonstrator is being built.

Motivating Example – Assembly. The use case comes from a manufacturer of electrical equipment for heavy duty power supply units as used for welding at construction sites. There are detailed documents available describing the assembly process and the parts involved. The actual assembly process is done by referring to a paper copy of the assembly manual. There are also CAD drawings available that can be used to derive the physical dimensions of the main parts. There are a number of important steps in the assembly process where the worker has to verify e.g. the correct connection of cables, or where he or she has to hold and insert, a part in a certain way. Furthermore, the assembly steps are associated with typical timings in order to ascertain a certain throughput per hour. The purpose of the associated Assistance Unit (we will use the abbreviation A/U) is to help inexperienced workers learn the process quickly and to remind experienced workers to double-check the critical steps. It should also be possible for the worker to ask for assistance at any point, through speech interaction and the A/U should be able to detect delays in a process step, as well as errors in the order of steps when the order of steps is important for the assembly process.

Further use cases not discussed in this paper address maintenance, repair, re-tooling, working on several machines simultaneously, and physical (robotic) assistance with lifting of heavy parts.

1.2   Structure, Usage and Purpose of Assistance Units

The research into assistance units started with a hypothetical structure that was deemed useful to define a methodology for creating and customizing different kinds of assistance units (Table 1):

Table 1.   Initial structure of assistance units.

Characteristic feature	Example
Description of the assistance unit	E.g. Assembly-Assistant
Type of assistance	Cognitive, physical or hybrid
Knowledge resources	E.g. data or media files required
Input format	E.g. speech interface, user interface
Input device(s)	E.g. microphone.
Output format	E.g. video explaining the assembly process
Output device(s)	E.g. tablet and in-built speaker

Looking at the concept from a technical perspective, however, it is difficult to refine this structure sufficiently, in order to use it as a taxonomic discriminator: the *Description* points at a potentially significant set of user requirements that are not covered in the specification. Likewise, distinguishing between *cognitive* and *physical* assistance is a binary decision that can be rewritten as "content management" (for cognitive needs of the user/worker) or "robotics" (addressing physical needs of the end user/worker). Input format and device, and analogously, output format and device are somewhat dependent entities because speech interfaces require microphones and video output requires a screen of some sort. The "*knowledge resources*" are worth a more detailed investigation with two possible technical approaches: firstly, one could think of the resources as media content that is searchable and through appropriate tagging, can be triggered in certain circumstances. The second approach would be to endow the assistance unit with a detailed model of the task and thus, make the unit capable of reasoning over the state of the task in order to assess progress or any deviations. This latter approach involves Artificial Intelligence techniques for knowledge representation and reasoning.

Usage as a Distinguishing Feature?   It is clear that in principle, there may be as many kinds of assistance units as there are kinds of manufacturing tasks. So one approach to classification could be high level distinctions of usage. Initially, we classified our use cases as being either assembly or maintenance or re-tooling, but in each of the cases, the distinguishing features only show themselves in the content of the knowledge resources or in the knowledge based model of the actual activity, without having a structural manifestation in the assistance unit itself.

Purpose (of Interaction) to the Rescue? The main reason for wanting a distinction between different kinds of assistance units is to have clearly separable methodologies for their development and deployment. Ideally, one should be able to offer pre-structured assistance "skeletons" that can be parameterized and customized by experts in the application domain, rather than requiring ICT specialists for bespoke programming. Having been dissatisfied with the previous attempts at structuring, we then looked at the way in which human and machine (aka assistance unit) were supposed to interact and this led to a proposed model that distinguishes eight forms of interaction or purpose, for using an assistance unit, with distinctive capabilities specified for each kind of unit (Table 2).

Table 2. Taxonomy of assistance units according to purpose of interaction

Purpose	Capabilities (A/U = Assistance Unit)
Mediator	A/U enables connection to/with a remote expert
Tutor	A/U explains a certain procedure
Trouble-shooter	A/U helps to localize a specific problem
Guard	A/U observes a situation and reports observations
Savior	A/U is called to clear up a known failure condition
Helper	A/U takes over physical sub-tasks in collaboration
Interpreter	A/U finds situation-relevant information from diverse sources, in an ill-defined problem space
Avatar (for tele-presence)	A/U enables an expert to act in a remote location (requires e.g. tele-robotics and situation awareness)

The above taxonomy has the advantage that it requires an increasing set of capabilities going from mediator to avatar. In other words, the distinguishing features allow one to define capabilities on an ordinal scale (Mediator < Tutor < Trouble-shooter < etc.). It is of course questionable whether the *mediator* role qualifies at all, as an assistance unit in its own right, but we may accept that a combination of on-line manuals, plus the ability to call an expert, all packaged in some smart-phone app, is sufficient functionality for a low-hanging fruit w.r.t. assistance units.

1.3 Assistance Units as Actors on the Shop-Floor

One of the defining features of any assistance unit is the ability to recognize when it is needed. This presupposes some form of situational awareness which can be obtained either by explicitly being called by the worker or by some sensory input triggering the A/U. Situational awareness requires explicit knowledge about the environment, knowledge about desirable sequences of actions (e.g. the steps of an assembly task) and it also requires some planning capability unless we can afford the system to learn by trial and error – an unlikely option in a competitive production environment.

Formal and semi-formal production planning methods are already being used in many manufacturing firms. Hagemann [4] gives the example of Event-Process-Chains for planning assembly lines in the automotive sector, but for assistance, one would need a more granular level of description. At the level of robotic task execution, there are examples such as the Canonical Robot Command Language [6] that provides vendor-independent language primitives to describe robotic tasks.

In the field of production optimization, many firms rely on some variety of measurement-time-method (MTM). There are different variants of MTM offering different granularity [3, 5, 7, 9] but the main problem is that the method is not well suited to formally describing purposeful, planned processes with inputs and outputs, but instead, focusses on the description of isolated actions and their time- and effort-related parameters. The method is also propagated rather exclusively, by a community of consultants belonging to the MTM association which keeps control over the intellectual property of practicing the art of MTM analysis.

So, while there are clearly potential connecting points with process planning at large, and with action-based productivity measurement at the more detailed level, one still requires an orchestration formalism that can distinguish between actors, can name inputs and outputs of processes, and can describe the manufacturing process as well as any assistive measure that one may want to add to the manufacturing process. To achieve this goal, we used – as a starting point – the agent model of Russell and Norvig [2] as shown below (Fig. 1).
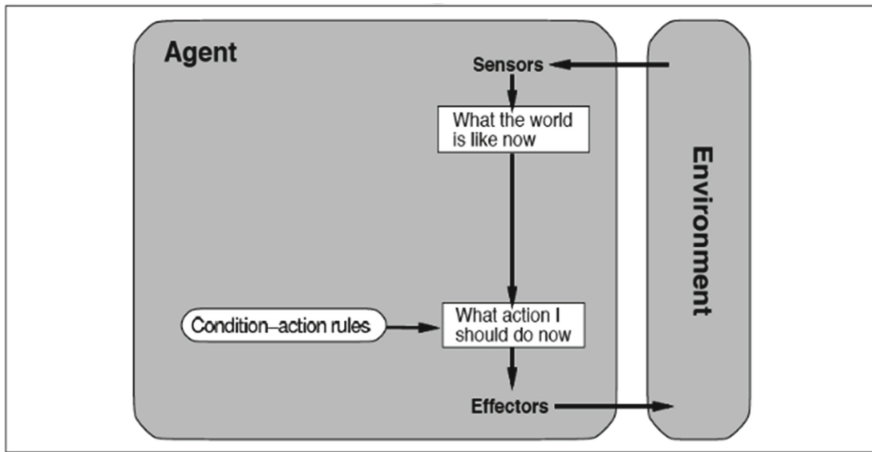

Fig. 1. Simple reflex agent according to Russell and Norvig [2].

The important issue to note is that we distinguish *Environment* and *Agent*, the latter consisting of *sensors* to perceive observations from the environment, *effectors* to manipulate entities in the environment and some form of rationality expressed in a world model and in a rule system that predetermines what actions the agent will take, under certain conditions. For a simple reflex agent as shown above, there is a direct relationship between singular observations and according actions. This is definitely not a sufficient

model for what we would call "intelligent" behavior, and so we should extend the basic model, as shown in the next figure.

As can be seen, the second model introduces a *memory* for the agent to store a state of the perceived world, and a *planning mechanism* to reason about the effects of possible actions. This means that the agent can now make decisions concerning its actions, on the basis of some utility function included in the planning mechanism. This agent-model gives us now a better handle on the original structural interpretation of the assistance unit: the input devices are a subset of all sensory input to the A/U and the associated formats are the data structures of those observations (also known as "percepts"). The output devices are a subset of all effectors of the A/U and quite clearly, for these to work in an intelligent way, a reasoning engine of some sort is required between the input and the output mechanisms of the A/U – as illustrated in Fig. 2.
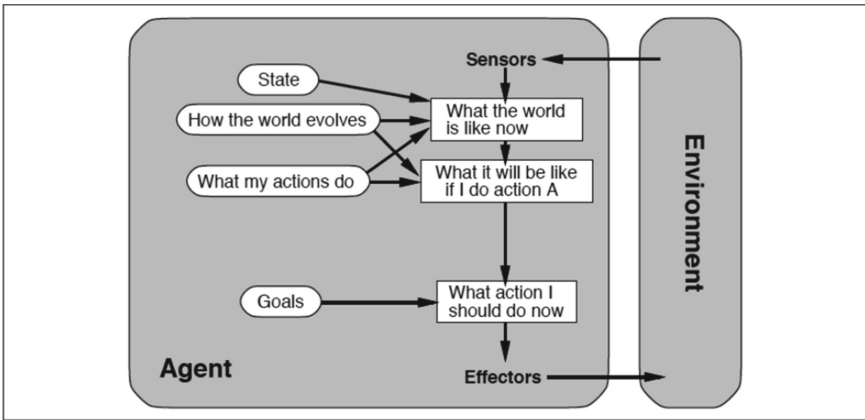

Fig. 2. Agent with explicit goals and reasoning, according to Russell and Norvig [2].

2   Situational Awareness and Assistance

The next step is to apply the agent-based Assistance Units to situations where timely interventions are required by the A/Us that are monitoring the shop-floor processes. We now bring observations, knowledge resources and collaborative activities into play, as proposed in the following conceptual model: in order for the A/U and the worker to be able to collaborate they must be in agreement with respect to the process in which they are involved. They also need to have some mutual understanding what knowledge is relevant for the process at the current step, and at least the A/U should have a basic understanding which knowledge can be assumed as given, for the worker. A concrete assistive intervention would then be to ask the worker whether he/she needs additional information for this step, or the next step, or whether they need any other form of assistance. Figure 3 below illustrates such a situation.
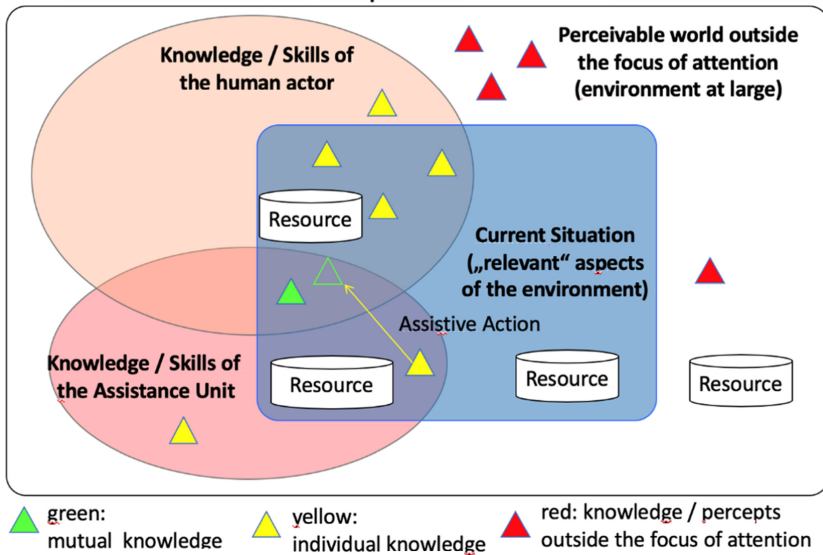
Fig. 3. Establishing a mutual focus of attention between human worker and assistance unit (Color figure online)

The diagram of Fig. 3 shows knowledge items that may come from observations or from prior knowledge, as triangles. The triangles outside the focus of attention (in red) could be any input that is recognized by the sensors of the actors, but which is not having any relevance to the current situation. There could also be red triangles inside the current situation, but outside the mutual focus of attention – these would then represent knowledge items that are not yet, or no longer, of immediate relevance. The arrow going from the yellow triangle of the A/U to the (green) frame of a triangle in the mutual focus of attention, indicates that an assistive intervention is occurring, i.e. the assistance unit is transferring knowledge to the worker, e.g., by reminding him/her to check the polarity of some electrical connection. The objects named "Resource" refer to repositories accessible to the workers and/or the A/Us. In the case of the worker, a printed manual would qualify as a resource, whereas in the case of an A/U we would require digital assets preferably accessible via a URL.

Note that the diagram represents a snapshot at a particular time step in the manufacturing process. The intersection of Assistance Unit, Worker and current overall situation constitutes the mutual focus of attention. It shows one green triangle representing relevant knowledge that the worker has and a green frame of a triangle representing a piece of knowledge that the worker is missing and that the Assistance Unit can bring into play (yellow triangle with arrow to green frame). Going back to the assembly use case introduced earlier, this could be the situation where the worker is reminded to check the polarity of an electrical connection and wants to reassure him-/herself by asking the assistance unit to display the connection diagram.

3 Process Description as Frame for Situational Awareness

Having a model of the overall manufacturing task is necessary for the A/U to be synchronized with the human worker w.r.t. the manufacturing process in which the assistance is supposed to happen. This may look like paying a high price (i.e. heavy modelling) for the sole purpose of getting a reminder to check some quality feature. However, behind the simple example is a more ambitious objective, namely to develop a process modelling methodology that can be used not only for simple assistive interventions, but also for highly synchronized collaborative work, as would be required when a collaborative robot helps a worker to place a truck engine in a chassis frame.

In the use case of the power unit assembly, four sensory inputs are used and related to the manufacturing process in question: (1) a video camera is analyzing the worker's movement in order to recognize certain actions, such as picking a screw and tightening it. (2) a second video stream recognizes artefacts on the work table, such as the side panels of the power unit. (3) A third video stream comes from a Hololens used by the worker, detecting the worker's gaze (direction of viewing) and also recognizing those artefacts that are directly in view of the worker. (4) A microphone picks up utterances of the worker, e.g. when he/she requests additional information, such as the procedure for assembling a specific part of the power unit. The detection systems of these four sensor streams translate the raw data into discrete observations that can then be used by the situational awareness module. In the next section we specify the modelling language that is able to merge process knowledge with observational knowledge, in order to arrive at the required situational awareness.

3.1 Pseudo-Natural Language for Process Steps, Observations and Interventions

The following is an example of the orchestration language for collaboration and assistance on the manufacturing shop floor. It has some syntactic sugar to make it look like a structured natural language, but it has a direct translation into a logic based programming formalism:

```
<Assembly>
Assembling product [PowerUnit-1] involves steps:
  place () material [Base] on equipment [Worktable] with step-
range <10-15s>,
  place () material [Bottom_cover] on top of [Base] with step-
range <10-15s>,
  place (from the right), component [Powersupply] on top of
[Bottom_cover]
     on trigger (show-how) show user (tutorial "Powerunit-1/Pow-
ersupply.mp3");
     on trigger (delay) ask user (need-help-in-task?)
     on trigger (behaviour) ask user (are-you-sure?)
     else do either:
     place (from the left), material [Side_panel_left] on side
of assembly then
place (from the right), material [Side_panel_right];
         or
place (from the right), material [Side_panel_right] then
place (from the left), material [Side_panel_left];
  place () material [Housing] on top of assembly [PowerUnit-1];
  inform about [Bottom_cover]: myfirm/PowerUnit-1/Connect-
ors_bottom_cover.jpg;
  inform about [Powersupply]: ]: myfirm/PowerUnit-1/Powersup-
ply.mp3;
  inform about [PowerUnit-1]: myfirm/PowerUnit-1/Assembly.mp3;
  assist on demand [PowerUnit-1]: Lifting_robot_Lifty;
  in_panic_do [PowerUnit-1]: emergency_plan_call_the_doc-
tor.help
</Assembly>
```

A similar description is conceivable for another task, e.g. the packaging of the power supply unit after assembly[1] :

```
<Packaging>
Packing product [PowerUnit-1] involves steps:
  Pick () material [cardboard-VPK1] and place it^i on equipment
[worktable],
  Transform [cardboard-VPK1] through [make-PU1-box] into mate-
rial [PU1-box],
  Place (bottom down) product [PowerUnit-1] into [PU1-box],
[…]
</Packaging>
```

Here, the construct `transform [X] through [P]` allows us to declare a subtask `make-PU1-box` which needs to be further specified elsewhere.

At this stage, our task description language is not complete yet, but the fragment above has been formalized and used for the demonstration of the assembly use case. In

---

[1] We allow simple anaphoric references: "and place **it**" refers to the cardboard.

the research implementation, we have focused on the formal constructs and it would be the task of productization, to add a compiler/interpreter that transforms the structured natural language into the respective formal constructs.

The following table summarizes our current set of constructs. The middle column shows the language construct, on the right we give examples or comments and on the left we explain the function of the construct for the purpose of expressing either manufacturing task steps or interventions by the assistance unit (Table 3).

Table 3. Constructs of the task description language for assistance units

Meta-language explanation	Language construct	Examples/Comments
Assistance trigger	**trigger** (show-how | delay | behavior | panic)	Show-how is initiated by the worker; delay gets triggered when the time is used up; behavior is used when a deviation is observed; panic triggers a defined emergency procedure
A/U Intervention control	On trigger T <action> … else	If trigger T is raised then do […], else do […]
Object	**product** [ID]	We associate assembly tasks with products, e.g. PowerUnit-1 with some catalogue-ID
Object	**equipment** [ID]	Instrumental objects such as tools, e.g. a screwdriver or a workbench, or an info-screen
Object	**material** [ID]	A part made of homogeneous material, e.g. a metal base of some product
Object	**component** [ID]	A specific functional part of a product that may itself be a subassembly or product
Object	**assembly** [ID]	A non-functional intermediate state of a product during the manufacturing process
Object	**user** [ID]	A human worker involved in the task
Action	**place (+ qualifiers)**	E.g. place (from the left) …
(More actions will need to be defined)	**pick X from Y**	E.g. taking a screw from a container
	(do) show X [to user]	What the A/U should display to the user
	(do) ask [user]	What the A/U should ask the user
	(do) help [user]	Refers to physical or reasoning help
A high- level action pointing to another task specification	**Transform** [X] through [P] into <object> newX	References a sub-process P, which transforms X into a new shape or form "NewX" e.g. creating a box out of some sheet of cardboard
Spatial qualifiers	**on top of**	These are qualifiers that have a local interpretation from the worker's point of view
	from the right **from the left below, from below**	They must be used carefully because they are relative to the worker's assumed position
Domain-specific terms	[Dom-Spec-Term]	E.g. material [Bottom-cover] – the referenced objects must be recognizable for the assistance system through some identification mechanism (e.g. RFID, image recognition)

At present, the domain-specific language only uses the action constructs "pick", "place" and "insert" for assembly tasks. This corresponds to the fact that the observational modules cannot distinguish any further activities, e.g. "gluing" or "screwing" etc. As soon as we extend the scope of manufacturing activities to e.g. machine re-tooling or to machine maintenance, we will have to extend the vocabulary, e.g. to express that some machine part may have to be disassembled. However, such distinctions only make sense if they can be detected by the observational modules. One simple "fallback"-detection mechanism is of course, to require the worker to confirm that step X has been done. What should become clear though is, that we already have a sufficient set of language constructs

to combine actors' task specifications, observations and triggered interventions, as the minimally required vocabulary.

4   Summary, Work in Progress and Beyond

We have presented a conceptual model and an orchestration language for specifying tasks on a manufacturing shop floor. The language allows the user to also specify assistive interventions in relation to observations that come as information streams from different sensory channels. At the time of writing, a software demonstrator for the described assembly use case is being implemented by the research partners and will be validated by the industrial use case partner.

The implementation of the situation awareness module is inspired by the Situation Calculus, a modelling approach for cognitive robotics pioneered in the works of Reiter [14], Levesque, Lesperance et al. [13]. We use the word "inspired" because at present, we do not make full use of e.g. the IndiGolog programming environment developed by that group at the University of Toronto [15]. Instead, we remain with the more straightforward task specification and use the unification algorithm of Prolog to activate the assistance triggers when certain conditions in the observation streams are met. In a future project, we plan to map our task specifications to equivalent actions and preconditions/post conditions as required by the Situation Calculus.

References

1. Apt, W., Schubert, M., Wischmann, S.: Digitale Assistenzsysteme - Perspektiven und Herausforderungen für den Einsatz in Industrie und Dienstleistungen. In: Hrsg.: Institut für Innovation und Technik (iit)in der VDI/VDE Innovation + Technik GmbH, WeApWi 2018, p. 80 (2018)
2. Russell, S., Norvig, P.: Artificial Intelligence – A Modern Approach, 3rd edn. Prentice Hall, Upper Saddle River (2012)
3. Hold, P., Ranz, F., Sihn, W.: Konzeption eines MTM-basierten Bewertungsmodells für digitalen Assistenzbedarf in der cyber-physischen Montage (2016)
4. Hagemann, S: Optimierung des Fertigungsplanungsprozesses im Karosseriebau durch die Entwicklung eines Konzepts zur automatisierten Generierung von Fertigungsanlagen mithilfe eines wissensbasierten Systems, p. 202. Master-Arbeit, Otto von Guericke Univ. Magdeburg (2016)
5. Bokranz, R., Landau, K.: Handbuch Industrial Engineering: Produktivitätsmanagement mit MTM. Schäfer-Peschel (2012)

6. Proctor, F.M., Balakirsky, S.B., Kootbally, Z., Kramer, T.R., Schlenoff, C.I., Shackleford, W.P.: The canonical robot command language (CRCL). Ind. Robot. **43**(5), 495–502 (2016). https://doi.org/10.1108/IR-01-2016-0037

7. Kelterborn, D.: Erweiterung eines Systems vorbestimmter Zeiten zur Bewertung der körperlichen Belastung in der Produktionslogistik. Dissertation, TU München (2016). http://www.fml.mw.tum.de/fml/images/Publikationen/Kelterborn.pdf

8. Mertens, A., Schlick, C.: Vorlesungsfolien 2016/17. RWTH Aachen, IAW. https://www.iaw-aachen.de/files/iaw/vorlesungen/Winter/2017/IEE/IEE_LE05_WiSe17_Slides.pdf

9. Benter, M.: Analyse von Arbeitsabläufen mit 3D-Kameras. Dissertation (2018). https://tore.tuhh.de/bitstream/11420/1920/1/Dissertation_BenterMartin.pdf

10. Hengstebeck, A., Weisner, K., Klöckner, M., Deuse, J., Kuhlenkötter, B., Roßmann, J.: Formal modelling of manual work processes for the application of industrial service robotics. In: 48th CIRP Conference on MANUFACTURING SYSTEMS - CIRP CMS 2015, Procedia CIRP, vol. 41, pp. 364–369 (2016)

11. Bauer, S.: Prozesssprachenbasiertes System zur Ansteuerung digitaler Menschmodelle als Teilkomponente einer Software zur Planung und Visualisierung menschlicher Arbeit in der Digitalen Fabrik. Dissertation (2015). https://monarch.qucosa.de/api/qucosa%3A20388/attachment/ATT-0/

12. Korn, O.: Context-aware assistive systems for augmented work - a framework using gamification and projection (2014). http://www.oliver-korn.de/wp-content/uploads/2014/05/Context-Aware-Assistive-Systems-for-Augmented-Work.-A-Framework-Using-Gamification-and-Projection_WEB.pdf

13. Levesque, H., Reiter, R., Lespérance, Y., Lin, F., Scherl, R.: GOLOG: a logic programming language for dynamic domains. J. Logic Program. **31**, 59–84 (1997)

14. Reiter, R.: Knowledge in Action. Logical Foundations for Specifying and Implementing Dynamical Systems. MIT Press, Cambridge (2001)

15. De Giacomo, G., Lespérance, Y., Levesque, H.J., Sardina, S.: IndiGolog: a high-level programming language for embedded reasoning agents. In: El Fallah Seghrouchni, A., Dix, J., Dastani, M., Bordini, R.H. (eds.) Multi-Agent Programming, pp. 31–72. Springer, Boston, MA (2009). https://doi.org/10.1007/978-0-387-89299-3_2