Agent Design Patterns for Assistance Systems in various Domains – a Survey

Philipp Marks, Timo Müller, Desirée Vögeli, Tobias Jung, Nasser Jazdi, Michael Weyrich

***Abstract*— The paradigm of multi-agent systems is used in various application scenarios like manufacturing control or multi-agent assistance systems that aim at providing assistance to the user of the system in a specific task or situation. With an increasing number of applications of multi agent assistance systems in the industrial automation domain, the different architectures of these applications can be compared and differences and similarities can be identified. This paper presents the results of the literature review of 18 agent-based assistance systems from different domains. Within the reviewed architectures, seven different agent types could be identified. In a further step, design patterns based on the application domain and the level of automation have been identified and discussed. These design patterns can be used to facilitate the development of future multi agent assistance systems.**

I. Introduction

The paradigm of multi-agent systems (MAS) has been a research subject since the 1990s [1]. Numerous architectures and applications based on the agent paradigm have been proposed and implemented in various domains, e.g. control of decentralized production systems [2, 3, 4], resource scheduling [5, 6] or simulation [7, 8].

Although no universally accepted definition of agents exists [9], a consensus about the typical agent properties can be found in the literature. Among the most named properties are: reaction to the environment, autonomy, goal-orientation and persistence [10]. VDI guideline 2653 defines agents as follows: "A technical agent is a delimited (hardware or/and software) entity with defined goals. A technical agent strives to achieve these goals through autonomous behavior, interacting with his environment and with other agents." [11]. Multiple agents can collaborate on a task in a multi-agent system. Approaches using the MAS implementation paradigm usually make use of the previously described agent properties. However, based on the purpose of the application, some properties are put more into focus than others.

Assistance systems that aim to support the user in a specified situation or during a specific action are another field where MAS can be applied. The goal orientation along with other agent properties can be used to provide adequate support to the user. The approaches in that field can be classified into different levels of automation, from simple assistance activities like data evaluation to highly automated or even autonomous execution of actions. When analyzing the MAS architectures of different assistance systems, the question arises whether similarities or differences can be identified in the architectures.

Such analysis has already been conducted by Lüder et al. in the field of agent-based production system control [12] where two different design patterns could be identified for agent- based control. However, this survey was focused on production system control and therefore the identified patterns are limited to that use case. In the production context, the role of production resources and the production order as well as aggregating production skills are in the focus of the paper.

Since multi-agent assistance systems consider different aspects and can be applied in various domains, an analysis of MAS architectures of assistance systems was conducted by the authors in this paper. The research questions that will be discussed are: (1) Which criteria can be used for the comparison of MAS architectures for assistance systems? (2) Do application domain and/or the level of automation of the provided assistance functionality have an influence on the MAS architecture? And finally, based on the previous findings: (3) Is it possible to identify general design patterns for agent-based assistance systems?

Accordingly, the aim of this paper is to analyze MAS architectures of agent-based assistance systems from various domains and to identify design patterns that can be used for future MAS development of assistance systems. Therefore, a literature review of existing MAS architectures has been conducted and the results have been analyzed and discussed.

The remainder of this paper is structured as follows: Section II gives an overview about assistance systems and application examples as well as their different levels of automation. The methodology of our literature review is described in Section III. Subsequently, the results of the analysis of different multi-agent architectures from our literature review are presented in Section IV. A discussion on findings and identified patterns is included in Section V. Finally, in Section VI the paper closes with conclusions and an outlook on future work.

II. Basics

Engineers are confronted with different types of tasks in their daily work. While some are repetitive, others are highly knowledge-based. Assistance systems which are aiming at providing suitable support or partly automating these tasks are in the focus of research. The following subsections will give a brief overview on assistance systems and explain the ten levels of automation.

All authors are with the Institute of Industrial Automation and Software Engineering; University of Stuttgart; Pfaffenwaldring 47, 70550 Stuttgart, Germany (corresponding author phone: +49 711 685 67301; fax: +49 711 685 67302; e-mail: ias@ias.uni-stuttgart.de).

A. Assistance Systems and their Functions

Human and machines working together has become more and more important over the last years and will be a topic within the next years [13]. One form of this collaboration are assistance systems to support users with their tasks and decisions. For more than 30 years, knowledge based systems are used to assist humans [14]. Nowadays, there are assistance systems like the lane and side assistant in a car, automated consistency check during the engineering [15], assistance for monitoring tasks [16], for scheduling problems [17] or for test management [18], and others. The rapid development in the field of artificial intelligence is also driving the field of assistance systems. Agent-based approaches are often used to realize assistance systems. Software agents [19] are autonomous software units that are able to make decisions and negotiate with each other. They can work together in a multi-agent system to solve problems cooperatively. There are multi-agent assistance systems that consist of just one type of agent [17] and others where the agents represent different roles [18, 15].

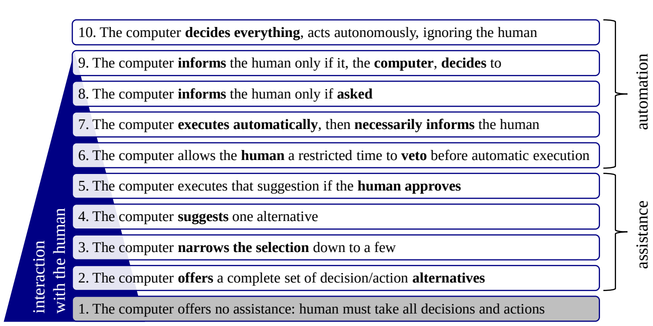
Functionalities which can by typically found in assistance systems, are according to [20]: (1) Creation and processing of information; (2) generation of alternatives; (3) evaluation of alternatives; (4) selection of alternatives; (5) monitoring of the decision execution and (6) controlling the decision execution.

For the classification of multi agent assistance systems, different criteria like the application domain or the level of automation (see following subsection) can be applied. For a broader overview on possible assistance systems classifications, the reader is referred to [20-23].

B. Levels of Automation

According to Sheridan [24] automation can be defined as a device or system that accomplishes (partially or fully) a function that was previously, or conceivably could be, carried out (partially or fully) by a human operator. Automation can be usefully characterized by a continuum of levels rather than as an all-or-none concept [25]. Fig. 1 shows how [24] distinguishes 10 levels of "Automation of Decision and Action Selection" ranging from 10, highest degree of autonomy, to 1, lowest degree of autonomy of the computer over human action. Since this paper addresses assistance systems, we focus on the levels 2 – 5:

Figure 1. Ten levels of automation according to [24]



• 2: The computer offers a complete set of decision/action alternatives

• 3: The computer narrows the selection down to a few

• 4: The computer suggests one alternative

• 5: The computer executes that suggestion if the human approves

Through this definition, it is possible to determine the level of automation of assistance systems.

III. Methodology and Comparison Criteria

Even with the restriction of assistance systems of production automation systems, a huge number of papers regarding multi-agent systems exists. Out of the first 150 found papers 18 were selected based on a categorization of the found papers. The categorization of the papers was done on the one hand by assigning the papers to the life cycle phases of a system (engineering, test, operation). On the other hand the papers were categorized into different domains according to the aim of the system and the task that the assistance system is used for. The determined domains in the field of industrial automation are engineering, test management, scheduling, monitoring and simulation. Additionally, the assistance systems ware categorized with respect to their level of automation.

After selecting the relevant literature, the papers were analyzed in depth and recurring agent roles were identified. To identify the roles of each agent in the system, the agents were analyzed and their main tasks and intended behaviors were identified. It is also possible that one agent represents the functionality of more than one role which is discussed later. Subsequently, frequently occurring agent types were derived from the identified roles.

With the identified agent types we tried to identify patterns in the investigated assistance systems. Several possible starting points exist for identifying patterns:

Firstly, as each assistance system has an interaction with the user, we researched how the interaction of the systems with the user was realized. Mainly, whether the interaction was realized with a dedicated (centralized) agent for user interaction or whether each agent interacts with the user when needed.

Furthermore we tried to identify a correlation between used agent types and whether the assistance system is coordinated in a central or decentralized way.

The third starting point was to search for dependencies with respect to the agent types of the assistance systems and the domains and use cases for which the assistance systems are applied.

Another starting point was to identify a dependency between the used agent types and the level of automation of the assistance system. Here the main idea is, that certain agent types may only be required in certain levels of automation.

We not only looked at those aspects separately, but also investigated whether combinations of two or more of these aspects can be identified and whether they can be used for the identification of patterns. For example, there could be similarities of the used agent types of assistance systems with a similar automation level and a similar domain. As sometimes some agents represent multiple agent roles and therefore can be matched to several agent types, we also tried to identify patterns with respect to agents which were assigned to different agent types.

Additionally to patterns in the used agent types we also tried to identify patterns in the architecture of the agent systems with respect to the above mentioned aspects.

Lastly we tried to identify special cases, which cannot be assigned to one of the identified patterns. For those special cases we tried to identify the reason, why they were realized in a completely different way than the other assistance systems.

IV. Analysis of different Agent Architectures

In this section, the investigated approaches in our literature review are listed in subsection A and have been mapped to different domains. Subsequently in subsection B, we analyze the architecture of these approaches and derive agent types that can be identified in the investigated architectures. Subsection C shows the summarized result of our literature review.

A. *Agent-based assistance systems in various domains*

In the literature review, approaches from different domains in the field of industrial automation have been considered. The investigated domains are: engineering, test management, scheduling, monitoring and simulation.

For supporting engineers in different actions during the engineering process of automation systems, multi-agent systems have been used to provide the needed support. Wagner [26] uses agents to assist the engineer proactively during the use of CAx tools. Using engineering knowledge about dependencies between components and modules from a library, interdependencies and problems in the current design are recognized and possible solutions are given to the engineer. Seemüller et al. [27] present an approach that also supports the engineering process of mechatronic systems. In contrast to [26], the agents do not only monitor changes in the design but also include models of single design activities and thus, proactively guide the engineer during the design process. Approaches which also guide the engineer through selected phases of the planning process are presented by Beyer et al. [28] and Marks et al. [29]. In [28], a multi-agent system is described which supports the rough planning of intralogistics systems. Marks et al. [29] describe a system which assists the engineer during the adaptation process of automated manufacturing machines which generates and evaluates suitable adaptation options for a given production request. To support the knowledge management during engineering, Monticolo et al. [30] use agents to create project memories which can be reused in future projects. Another assistance approach that is based on previously gained knowledge is presented by Li and Chen [31]. The approach aims at shortening the design cycle of new machine tools by applying a MAS that uses case-based reasoning.

Multi-agent assistance systems are also used for test management. Abele [18] describes a system that supports the planning of the regression test in distributed manufacturing plants by combining manual modeling with automated data processing coordinated by software agents. Malz et al [32] describe a multi-agent system that determines the prioritization order of test cases which increases the test effectiveness and the fault detection rate. Kumar and Bhatia [33] developed a multi-agent assistance system to speed up the test case generation using UML diagrams to identify changes.

Another task multi-agent assistance systems are often used for is scheduling. Agents are able to negotiate procedures and schedules of decentralized problems. The application areas range from space operations where human activities must be planed [17] over the coordination of chartering traffic airplanes [6] until flexible and efficient manufacturing systems scheduling [34].

Another field of application for agent-based assistance systems is the monitoring of automated systems. Lu and Sy [35] are using a multi-agent system to monitor the manufacturing line of an engine manufacturer to support the personnel in decision-making. In [36] a multi-agent system is used to increase the availability of a system by monitoring the performance of the system and also to ensure safety. Fernandez et al. [37] detect disruptive events in a system by monitoring to enable a more efficient scheduling.

In the field of simulation, multi-agent assistance systems are also often used either by modeling a production system using agents or by connecting agents to simulations to enable an assistance system. In [38] agents are used to model a repairable manufacturing system to increase the overall production rate. Vögeli [39] and Jung [40] use MAS as a connection to different simulation tools. Vögeli [39] utilizes a multi-agent system to parallelize a multi physical simulation and [40] enables a dynamic co-simulation of IoT-systems by using a multi-agent system.

Besides those reviewed systems, additional multi-agent assistance systems exist, which provide an even higher level of assistance, so that the demarcation to an automation system becomes difficult. Mubarak [41] for example realizes a self-management system for automation systems, in which the level of assistance can be regulated up to level 6, where the multi-agent system actively controls the automation system, so that the levels of assistance are exceeded. As it is not possible to determine which agents are used for assistance functionalities and which are used for automation functionalities, those systems are not considered in the research.

B. *Derivation of agent types*

Through reviewing the different architectures of the mentioned assistance systems, we were able to derive the following seven (abstract) agent types:

- **Algorithm-Agent**: used to reach objectives like optimization, evaluation of options etc. This agent type merely contains the necessary algorithms for this task and requests or receives the required data from other agents.

- **UI-Agent:** enables the communication between user and assistance system via a dedicated agent which is responsible for handling the user interface.

- **System-Component-Agent (active):** is representing a component of the system as well as its corresponding interests, knowledge and data model within the assistance system.
- **System-Component-Agent (passive):** acts as a passive interface between the assistance system and external tools or components. Its main task is to encapsulate the external component and to translate bidirectional messages and to forward them to other agents or the component.
- **Database-Agent:** manages the data or knowledge of a specified data source.
- **Aggregation-Agent:** aggregates knowledge or information from various data sources.
- **Coordination-Agent:** coordinates the internal processes of the multi-agent systems and acts as a central instance within the commonly decentralized multi-agent system.

Besides these agent types, we figured out that the infrastructure of the agent-environment, e.g. the Yellow Pages or the Message Transport Service were either provided by the used implementation framework, e.g. JADE or were implemented via an own agent, e.g. in [31]. As those agent types only provide agent specific infrastructure and usually do not affect the corresponding architecture for the specific use case, we did not include them in our further considerations.

As mentioned above, some of the architectures combine different roles and therefore different agent types in one agent e.g. [17, 36, 39].

C. Overview and comparison of reviewed MAAS

The aggregated results of our literature review are summarized in Table I. The table includes the domain where the multi-agent assistance system (MAAS) is applied, the used agent types as well as the level of automation.

TABLE I. COMPARISON OF MAAS (●: DEDICATED AGENT FOR THIS TYPE EXISTS, ◐: AGENT TYPE IS INCLUDED IN AN AGENT)

Domain	MAAS	Algorithm-Agent	UI-Agent	System-Component-Agent (active)	System-Component-Agent (passive)	Database-Agent	Aggregation-Agent	Coordination-Agent	Level of Automation
Engineering	[26]	●		●		●			3/4
	[27]	●	●	●	●	●	●	●	3
	[30]	◐	◐	◐		●	●		5
	[28]	●		●		●		●	3
	[29]	●	●	●	●	●	●	●	3
	[31]	●	●		●	●	●		3
Test Management	[32]	◐	●	◐		●	●		3
	[18]	●	●	●		●	●	●	3
	[33]	◐	◐			●	●	◐	3
Scheduling	[6]	●	●	●	●			●	4
	[17]	◐	◐	◐				◐	3/4
	[34]	●		●				●	4
Monitoring	[35]	●	●	●		●			3
	[36]	●,◐		●,◐		●, ◐		●	3/4
	[37]	●	●	●		●, ◐	●	●, ◐	3
Simulation	[38]			●					2/3
	[39]	◐	●,◐		◐	●		◐	4
	[40]				●			●	2

V. FINDINGS ON PATTERNS

Based on the literature review, we identified some patterns regarding the used agent types based on the domain and the level of automation. The results and findings are discussed in this section.

Finding 1: The architectural design of the necessary user interface in the multi agent system differs although all assistance systems need to get input from and provide information to the user. While some architectures include a dedicated *UI-Agent*, other researchers decided not to depict them specifically or to distribute the user interface over different agents each providing their own interface if necessary. A general statement if a dedicated *UI-Agent* should be used seems impossible and seems to depend on the use case and the preferences of the system architect.

Finding 2: Concerning the *Level of Automation*, it can be concluded that agent-based assistance systems with automation level 2 ("The computer offers a complete set of decision/action alternatives", cf. Fig. 1) are not likely to contain an *Algorithm-Agent* while all investigated approaches with automation level 3 ("The computer narrows the selection down to a few", cf. Fig. 1) or higher use at least one agent of this type. As per definition of the *Levels of Automation*, below level 3 no narrowing of possible solutions is required and therefore there is no need for a corresponding algorithm which performs this usually knowledge-based task.

Finding 3: Every reviewed assistance system uses *System-Component-Agents*, either active or passive except for [33]. In this assistance system no *System-Component-Agent* is required because no system components have to be represented, as the system which has to be tested, is considered as a whole. In the other reviewed assistance systems the system is modularized by using *System-Component-Agents*.

Finding 4: In the domain of scheduling, the investigated approaches do not contain *Database-* or *Aggregation-Agents* in their architecture. Additionally, all investigated approaches in the scheduling domain use a dedicated *Coordination-Agent* to handle the complexity of scheduling at a centralized location rather than using a decentralized scheduling approach. In other domains, however, a dedicated central *Coordination-Agent* can only be found in some of the architectures. The usage of a centralized or decentralized architecture seems to be depending on the specific use case and not on the domain.

Finding 5: In the scheduling domain we observed the usage of *System-Component-Agents (active)* in each investigated assistance system architecture, which were used to represent the interests of the corresponding components of the automated system that shall be scheduled.

Finding 6: In contrast to approaches of the scheduling domain, the approaches of the test management domain use *Database-Agents* as well as *Aggregation-Agents* to aggregate and manage the different sources of information and knowledge. To provide assistance in a test management scenario, it is necessary to combine distributed information sources in order to optimize the selection of test cases rather than using brute force approaches.

Finding 7: Engineering is another domain which is heavily based on (a-priori) knowledge and available system models. Thus, assistance system architectures in the engineering domain commonly use agents of the type *Database-Agent* and often also *Aggregation-Agents* to aggregate information.

Finding 8: Additionally, we could not find a dependency between the *Level of Automation* and the usage of a *Coordination-Agent*. Centralized as well as decentralized approached seem to be applicable for different levels of automation.

VI. Conclusions and Future Work

This paper presented the results of the literature review of 18 multi-agent assistance systems that are used in different application domains of industrial automation systems, namely engineering, test management, scheduling, monitoring and simulation. The reviewed approaches were discussed regarding the different agent types that are included in the multi-agent architecture and the level of automation that the approach is able to provide to the user.

Besides the basic agent infrastructure like Agent Management System or Directory Facilitator, seven agent types could be identified in the investigated approaches. In many cases, the agent architecture used dedicated agents for the different types, however some architectures included more than one type in one agent. This observation goes hand in hand with the Gaia design methodology which allows to assign multiple roles to one agent. It seems to depend on the application and the preferences of the MAS architect whether to use dedicated agents for one type or to combine them, e.g. for performance reasons.

Based on the literature review of multi-agent assistance systems, patterns for the agent architecture in the investigated domains and the level of automation were identified. Within most domains, it could be observed that some of the agent types are typically used in assistance systems of that domain. This observation could be confirmed by deeper analysis of the assistance function that the assistance system is providing. Regarding the level of automation, no significant patterns could be identified for the levels 3 to 5. However, only low assisted processes (automation level 2) which do not require a narrowing of a complete set of decision or action alternatives, usually do not use dedicated agents for the execution of optimization algorithms since they are not required.

Since only a limited number of papers in the wide field of agent-based assistance systems could be reviewed and summarized in a conference paper, future research will investigate a greater number of approaches to verify our observations. The research methodology can also be extended towards other aspects like the usage of different knowledge management techniques or different forms of knowledge representation. The next steps will also include the comparison of our results with the ones of other research groups that work in the field of agent design patterns, e.g. the Expert Committee 5.15: Agent Systems of the VDI / VDE Society Measurement and Automatic Control (GMA).

References

[1] M. Wooldridge and N.R. Jennings: Intelligent agents: Theory and practice. The knowledge engineering review, 10(2), 115-152, 1995.

[2] P. Leitao: Agent-based distributed manufacturing control: A state-of the-art survey, Engin. Appl. of Artif. Intell. 22, pp. 979–991, 2009.

[3] S. Bussmann, N.R. Jennings, M. Wooldridge: Multiagent systems for manufacturing control: A design methodology. In Series on Agent Technology, Springer-Verlag: Berlin, Germany, 2004.

[4] G. Cândido, J. Barata: A Multiagent Control System for Shop Floor Assembly, within V. Mařík, V. Vyatkin, A.W. Colombo (Eds.): HoloMAS 2007, LNAI 4659, pp. 293 – 302, 2007.

[5] W. Shen, L. Wang & Q. Hao: Agent-based distributed manufacturing process planning and scheduling: a state-of-the-art survey. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 36(4), 563-577, 2006.

[6] C. Dannegger: Ready for Take-Off–Softwareagenten disponieren Flugcharter-Verkehr. In Agentensysteme in der Automatisierungstechnik (pp. 243-254). Springer, Berlin, Heidelberg, 2013.

[7] L. Tan, M. Hu & H. Lin: Agent-based simulation of building evacuation: Combining human behavior with predictable spatial accessibility in a fire emergency. Information Sciences, 295, 53-66, 2015.

[8] E. Zankoul, H. Khoury & R. Awwad: Evaluation of agent-based and discrete-event simulation for modeling construction earthmoving operations. In ISARC. Proceedings of the International Symposium on Automation and Robotics in Construction (Vol. 32, p. 1). Vilnius Gediminas Technical University, Department of Construction Economics & Property, 2015.

[9] M. Wooldridge: Intelligent agents: The key concepts. In ECCAI Advanced Course on Artificial Intelligence (pp. 3-43). Springer, Berlin, Heidelberg, 2001.
[10] S. Franklin, A. Graesser: Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents. University of Memphis, Institute for Intelligent Systems, 1996.
[11] VDI/VDE 2653-1: Multi-agent systems in industrial automation - Fundamentals, VDI/VDE-Richtlinien, VDI, Düsseldorf, 2017.
[12] A. Lüder, A. Calá, J. Zawisza and R. Rosendahl, "Design pattern for agent based production system control — A survey," 2017 13th IEEE Conference on Automation Science and Engineering (CASE), Xi'an, 2017, pp. 717-722.
[13] G.S. Gartner, G. S.: Hype Cycle for Emerging Technologies maps out evolving relationship between humans and machines. Press Release, Stamford, Connecticut (www.gartner.com/newsroom/id/2575515), 2013.
[14] B. Chandrasekaran, B.: Generic tasks in knowledge-based reasoning: High-level building blocks for expert system design. IEEE expert, 1(3), 23-30, 1986.
[15] M. Rauscher & P. Goehner: Agent-based consistency check in early mechatronic design phase. In DS 75-9: Proceedings of the 19th International Conference on Engineering Design (ICED13), Design for Harmonies, Vol. 9: Design Methods and Tools, Seoul, Korea, 19-22.08. 2013.
[16] F. Aiello, F.L. Bellifemine, G. Fortino, S. Galzarano, R. Gravina: An agent-based singnal processing in-node environment for real-time human activity monitoring based on wireless body sensor networks. In: Engineering Applications of Artificial Intelligence Vol. 24, Issue 7 (2011), pp. 1147-1161, 2011.
[17] D. Schreckenghost and R. P. Bonasso, "Multi-agent system for managing human activities in space operations," 2006 IEEE Aerospace Conference, Big Sky, MT, 2006.
[18] S. Abele, P.Göhner and M. Weyrich: Enhancing an Agent-based Test Case Prioritization System by Event Evaluation. MATES 14, 2014.
[19] W. Brenner, R. Zarnekow, H. Witting: Intelligent Software Agents: Foundations and Ap-plications. Springer Publishing Company, Incorporated, 2012.
[20] D. Blutner, S. Cramer, S. Krause, T. Mönks, L. Nagel, A. Reinholz, M. Witthaut: Assistenzsysteme für die Entscheidungsunterstützung. In: Große Netze der Logistik: Die Ergebnisse des Sonderforschungsbereichs 559, Buchholz, P.; Clausen, U. (Hrsg.), Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 241–270.
[21] M.R. Endsley: Level of automation: Integrating humans and automated systems. In Proceedings of the Human Factors and Ergonomics Society Annual Meeting (Vol. 41, No. 1, pp. 200-204). Sage CA: Los Angeles, CA: SAGE Publications, 1997.
[22] B. Hellingrath, & A. Kuhn: Supply chain management: Optimierte Zusammenarbeit in der Wertschöpfungskette. Springer-Verlag, 2013.
[23] H. Wandke: Assistance in human-machine interaction: a conceptual framework and a proposal for a taxonomy. Theoretical Issues in Ergonomics Science, Vol. 6, No. 2, p. 129-155, 2005.
[24] R. Parasuraman, T. B. Sheridan, and C. D. Wickens: A model for types and levels of human interaction with automation. IEEE Trans. Syst., Man, Cybern. A, vol. 30, no. 3, pp. 286–297, 2000.
[25] R. Parasuraman and V. Riley: Humans and Automation: Use, Misuse, Disuse, Abuse. Human Factors, vol. 39, pp. 230-253, 1997.
[26] T. Wagner: Agentenunterstütztes Engineering von Automatisierungs-anlagen. Dissertation, Institut für Automatisierungs- und Softwaretechnik, Universität Stuttgart, Stuttgart, 2008.
[27] H. Seemüller, M. Chami, M & H. Voos: Agentenbasierte Koordination und Überwachung des Designs mechatronischer Systeme. In Agentensysteme in der Automatisierungstechnik (pp. 255-273). Springer, Berlin, Heidelberg, 2013.
[28] T. Beyer, R. Yousefifar, S. Abele, M. Bordasch, P. Gohner & K.-H. Wehking: Flexible agent-based planning and adaptation of material handling systems. In Automation Science and Engineering (CASE), 2015 IEEE International Conference on (pp. 1060-1065). 2015.
[29] P. Marks, M. Weyrich, X. L. Hoang and A. Fay, "Agent-based adaptation of automated manufacturing machines," 2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), Limassol, 2017, pp. 1-8.
[30] D. Monticolo, S. Mihaita, H. Darwich & V. Hilaire: An agent-based system to build project memories during engineering projects. Knowledge-Based Systems, 68, 88-102, 2014.
[31] F. Li & N. Chen: An intelligent assistant design system based on CBR and MAS. In Computer-Aided Industrial Design & Conceptual Design, 2009. CAID & CD 2009. IEEE 10th International Conference on (pp. 1335-1339). 2009.
[32] C. Malz; N. Jazdi & P. Göhner: Prioritization of test cases using software agents and fuzzy logic. In Software Testing, Verification and Validation (ICST), 2012 IEEE Fifth International Conference on. IEEE, S. 483-486, 2012.
[33] P.K. Arora & R. Bhatia: Agent-Based Regression Test Case Generation using Class Diagram, Use cases and Activity Diagram. Procedia Computer Science, 125, 747-753, 2018.
[34] I. Badr, Agent-based dynamic scheduling for flexible manufacturing systems. Aachen: Shaker Verlag Gmbh, 2011.
[35] K.Y. Lu & C.C. Sy: A real-time decision-making of maintenance using fuzzy agent. Expert Systems with Applications, 36(2), 2691-2698, 2009.
[36] V. Ebrahimipour, K. Rezaie & S. Shokravi: Enhanced fmea by multi-agent engineering fipa based system to analyze failures. In Reliability and Maintainability Symposium (RAMS), 2010 Proceedings-Annual (pp. 1-6). 2010.
[37] E. Fernández, C.M. Toledo, M.R. Galli, E. Salomone & O. Chiotti: Agent-based monitoring service for management of disruptive events in supply chains. Computers in Industry, 70, 89-101, 2015.
[38] E.F. Alsina, G. Cabri & A. Regattieri: An agent-based approach to simulate production, degradation, repair, replacement and preventive maintenance of manufacturing systems. In Computational Intelligence in Production and Logistics Systems (CIPLS), 2014 IEEE Symposium on (pp. 24-31). 2014.
[39] D. Vögeli, S. Grabmaier, M. Jüttner, M. Weyrich, P. Göhner & W. M. Rucker: Intelligent and Distributed Solving of Multiphysics Problems Coordinated by Software Agents – An Intelligent Approach for Decentralized SImulations. In Proceedings of the 10th International Conference on Agents and Artificial Intelligence, vol. 1, pp. 200–207, 2018.
[40] Jung, T.; Jazdi, N.; Weyrich, M.: A Survey on Dynamic Simulation of Automation Systems and Components in the Internet of Things. 22nd IEEE ETFA, Limassol, Cyprus, 2017
[41] H. K. Mubarak, Agentenbasiertes Selbstmanagement von Automatisierungsanlagen, 1st ed. Aachen: Shaker, 2013.