Metadata of the chapter that will be visualized in SpringerLink

Book Title	Recent Advances in Robotics and Automation
Series Title	7092
Chapter Title	Process Rescheduling and Path Planning Using Automation Agents
Copyright Year	2013
Copyright HolderName	Springer-Verlag Berlin Heidelberg

Corresponding Author	Family Name	**Merdan**
	Particle	
	Given Name	**Munir**
	Suffix	
	Division	Vienna University of Technology
	Organization	Automation and Control Institute
	Address	Vienna, Austria
	Email	merdan@acin.tuwien.ac.at
Author	Family Name	**Lepuschitz**
	Particle	
	Given Name	**Wilfried**
	Suffix	
	Division	Vienna University of Technology
	Organization	Automation and Control Institute
	Address	Vienna, Austria
	Email	lepuschitz@acin.tuwien.ac.at
Author	Family Name	**Grössing**
	Particle	
	Given Name	**Benjamin**
	Suffix	
	Division	Vienna University of Technology
	Organization	Automation and Control Institute
	Address	Vienna, Austria
	Email	groessing@acin.tuwien.ac.at
Author	Family Name	**Helbok**
	Particle	
	Given Name	**Markus**
	Suffix	
	Division	
	Organization	COPA-DATA GmbH
	Address	Karolingerstrasse 7b, 5020, Salzburg, Austria
	Email	markush@copadata.com

Abstract

The process industry faces a permanently changing environment, where sudden component failures can significantly influence the system performance if not treated in an appropriate amount of time. Moreover, current market trends have to be met such as short production times, a low price as well as a broad spectrum of product and process varieties. Distributed intelligent control systems based on agent technologies are seen as a promising approach to handle the dynamics in large complex systems. In this chapter, we present a multi-agent system architecture capable to answer to the major requirements in the process domain. The architecture is based on agents with diverse responsibilities as well as tasks and separates the control software of agents controlling hardware components into two levels, the high level control and the low level control. Our system architecture has also the ability to flexibly reschedule allocated jobs in the case of resource breakdowns in order to minimize downtimes. This goes hand in hand with a dynamic path finding algorithm to enhance the flexibility of transport tasks. The system is currently tested and evaluated in the Odo Struger Laboratory at the Automation and Control Institute.

Process Rescheduling and Path Planning Using Automation Agents

Munir Merdan, Wilfried Lepuschitz, Benjamin Grössing and Markus Helbok

AQ1

Abstract The process industry faces a permanently changing environment, where sudden component failures can significantly influence the system performance if not treated in an appropriate amount of time. Moreover, current market trends have to be met such as short production times, a low price as well as a broad spectrum of product and process varieties. Distributed intelligent control systems based on agent technologies are seen as a promising approach to handle the dynamics in large complex systems. In this chapter, we present a multi-agent system architecture capable to answer to the major requirements in the process domain. The architecture is based on agents with diverse responsibilities as well as tasks and separates the control software of agents controlling hardware components into two levels, the high level control and the low level control. Our system architecture has also the ability to flexibly reschedule allocated jobs in the case of resource breakdowns in order to minimize downtimes. This goes hand in hand with a dynamic path finding algorithm to enhance the flexibility of transport tasks. The system is currently tested and evaluated in the Odo Struger Laboratory at the Automation and Control Institute.

Based on “Advanced Process Automation Using Automation Agents”, by Munir Merdan, Wilfried Lepuschitz, and Emilian Axinia which appeared in the Proceedings of the 5th International Conference on Automation, Robotics and Applications (ICARA 2011) © 2011 IEEE.

M. Merdan (✉) · W. Lepuschitz · B. Grössing
Vienna University of Technology, Automation and Control Institute,
Vienna, Austria
e-mail: merdan@acin.tuwien.ac.at

W. Lepuschitz
e-mail: lepuschitz@acin.tuwien.ac.at

B. Grössing
e-mail: groessing@acin.tuwien.ac.at

M. Helbok
COPA-DATA GmbH, Karolingerstrasse 7b, 5020 Salzburg, Austria
e-mail: markush@copadata.com

G. Sen Gupta et al. (eds.), *Recent Advances in Robotics and Automation*,
Studies in Computational Intelligence 480, DOI: 10.1007/978-3-642-37387-9_5,

Layout: T1 Standard SC Book ID: 272920_1_En Book ISBN: 978-3-642-37386-2
Chapter No.: 5 Date: 30-3-2013 Page: 2/11

1 Introduction

The process industry faces a permanently changing environment and has to meet current market trends such as short production times, a low price as well as a broad spectrum of product and process varieties. Due to these reasons, the importance of automation in the process industry has increased dramatically in recent years. It has become a vital force in the entire chemical, oil, gas and biotechnology industries [1]. The proper function of an automation system is critical for the operation of the majority of plants today, since it is the automation system that performs control and other advanced functions including optimization, scheduling, planning, monitoring, etc. [2]. Traditional approaches for solving scheduling problems encounter great difficulties when applied in real situations, because these scheduling methods use simplified theoretical models and are centralized in the sense that all computations are carried out in a central computing unit [3]. Besides, to make appropriate decisions, an automation system relies on exchanging different kinds of data such as process measurements, diagnostic data or historical data. Nevertheless, due to different standards in the domain, information is represented in different syntaxes as well as semantics and on different levels of abstraction. Considering the ongoing information and knowledge explosion, intelligent process control systems are needed that include features for knowledge and information management to effectively manage and access information for efficient decision making [1, 4]. In addition, deviations from the original schedule and information about equipment breakdowns provided by the control and monitoring system will eventually trigger rescheduling mechanisms and in some cases require a reconfiguration of the system. The automation system should be able to change quickly and cost-effectively from its current configuration to another configuration without being taken off-line [5]. Hence, online modification of the system configuration is a key requirement for future process automation systems [2].

The introduction of artificial intelligence techniques is seen as a promising trend in the process industry [1]. The application of a multi-agent system (MAS) is recognized as a convenient way to handle the dynamics in large complex systems reducing the complexity, increasing flexibility and enhancing fault tolerance [6]. Agents cooperate and coordinate their actions in order to achieve their own as well as the system's aims, which are beyond the capabilities possessed by an individual agent. Moreover, by being able to use parallel computation and to apply diverse strategies and methods for solving their simple local problems, agents can significantly improve the efficiency and increase the performance of the entire system [7]. A further advantage of these approaches is their better reactivity and adaptability in highly dynamic environments. Most of the scheduling models for chemical processes assume that all required data is certain and therefore do not include uncertainty in the calculation. However, it can be shown that a schedule generated by a deterministic model based on nominal values of the parameters may be infeasible upon the occurrence of uncertainty [8]. Besides, the presence of real-time information permanently forces the revision of pre-established schedules

Layout: T1 Standard SC | Book ID: 272920_1_En | Book ISBN: 978-3-642-37386-2
Chapter No.: 5 | Date: 30-3-2013 | Page: 3/11

and the consideration of new conditions and constraints related to certain environments. Under such circumstances a fast reaction is required in terms of identifying alternative resources and respective time slots to continue production. Besides, the ability to flexibly reschedule allocated jobs is needed in the case of resource breakdowns in order to minimize downtimes. Furthermore, these rescheduling actions should be performed autonomously and with a modest computational effort. Despite a resource failure, the system should be able to autonomously continue its activities (as may be the case in a degraded mode) while the broken resource is repaired. Furthermore, the reallocation of jobs awarded to a broken resource is required, so these jobs are performed by other resources and do not unnecessarily block other jobs. Effective resource reallocation processes minimize the amount of overcapacity needed to cope with unpredictable events and consequently reduce the related inventory costs needed for these overcapacities. Moreover, the maximal usage of potential process redundancies and flexible routing capabilities is required in order to minimize the influence of a failure [9].

In this chapter, we introduce a new architecture of automation agents to answer to the requirements in the process domain mentioned as mentioned above. The architecture separates the control software into two levels, the high level control (HLC) and the low level control (LLC), related to specific concerns based on their ability to execute particular activities, e.g. to perform failure detection and recovery. This combination enables efficient process scheduling, monitoring and diagnosis tasks, where the information from diverse data sources is combined in a flexible manner to make conclusions about the overall state of the process [10, 11].

The chapter is structured as follows. The subsequent section introduces a motivation example. Section 3 presents the architecture of the MAS. In Sect. 4 we describe system rescheduling and path planning and Sect. 5 concludes this chapter.

2 State of the Art

Conventional control approaches based on hierarchical architectures are limited in dealing with emerging requirements due to their inflexible structures and operating rules [12]. Agent-based control systems have the capability to respond quickly and correctly to dynamic changes in the production environment, and differ from the conventional approaches with their inherent capabilities to adapt to emergence without external intervention [13]. The approaches mentioned in this section show some developments in this field.

The management of chemical batch plants is based on collecting and processing huge amounts of data, which is subsequently exploited. This data is a valuable source of information for decision-making, scheduling, process control, fault analysis, etc. A common representation of knowledge is needed to merge the mass of heterogeneous process information. There are many standards for information representation in the field, but the agents need to be able to address both low level and high level information in the same context [10]. One approach for the integration of information systems is the usage of shared ontologies. Ontologies can be used to describe the semantics of the information sources and make the contents explicit, thereby enabling the integration of existing information repositories [4]. In this context, Sesen et al. [14] developed an ontological framework for automated regulatory compliance in pharmaceutical manufacturing, Batres et al. [15] provided a brief overview of an upper ontology based on ISO 15926, Venkatasubramanian et al. [4] introduced an ontological informatics infrastructure for pharmaceutical product development and manufacturing, and Muñoz et al. [16] presented a batch process ontology. We use these ontologies as the roots for a description of a process automation system and for a definition of the ontology-based world model of each agent types. The usage of ontologies for knowledge representation and high level reasoning is a major step ahead in the area of agent-based control solutions [17].

Most of the earlier developed scheduling methods have difficulties in solving actual industrial problems such as changed schedules resulting from changed production orders, due to the complexity of real-life manufacturing constraints [18]. Hamaguchi et al. [5] presented an approach for solving scheduling problems. Seilonen et al. [19] introduced an approach suitable also for handling failures. To incorporate uncertainty in the process industries, it is necessary to investigate a new unified framework for planning and scheduling under uncertainty [20]. Besides, it is important to define an overall framework for specifying the system as well as component behavior in the occurrence of exceptions [21]. This goes hand in hand with the requirement of integrating a dynamic path finding method for determining appropriate material paths between the tanks. In comparison to hard-coded paths as often applied currently in industrial practice, dynamic path finding with agent technology can greatly enhance a production system's flexibility during runtime. But also extensions to the system can be applied without having to reprogram any hard-coded paths.

3 System Architecture

In order to reduce its complexity, the control of a process automation system can be organized in several "hierarchically" ordered layers as it is also done in current industrial solutions. We specified four layers: Management, Planning, Scheduling and Executive Layer. The Management Layer is responsible for the entire system functionality. It is also concerned with the communication with the external environment and provides solutions for complex problems related to the global environment. It accepts orders on a routine basis. The Planning Layer links process planning with product design. It is basically concerned with the sequencing of process steps as well as with the identification of product types and quantities to be produced. The Scheduling Layer is concerned with the synchronization of production needs with available resource capacities. This layer is responsible for negotiating with the resources and the task allocation between resources.

The Executive Layer is related to the process system's equipment. On this layer, the production tasks are executed considering the resources' constraints and abilities, their performances are measured and if a failure or disruption is diagnosed, the higher layers are informed. The architecture of the MAS is based on the specified layered structure with a particular agent type for each layer (see Fig. 1). In the following the different types of agents and their general activities are described.

The Task Agent determines the equipment for performing the operations of the recipe by searching for Automation Agents with suitable equipment in the list of the Directory Facilitator Agent. When appropriate Automation Agents are found, the tasks for the first operation of the recipe are created including transport tasks.

Work Agents receive tasks from Task Agents and manage Automation Agents. The tasks can either be transport or production tasks. After receiving a transport task with the associated source and destination information, the Work Agent generates a path using the path finding algorithm as described in Sect. 4. Ultimately, the Work Agent sends orders to Automation Agents to have them make state changes on hardware components. These changes can be for instance opening a valve or activating a pump. Newly created or changed routes due to system modifications are automatically considered by this agent and the system.

Automation Agents, key components of our architecture, are those agents incorporating some kind of physical representation. They control manufacturing resources (such as a valve or a pump) providing particular processes or services. Each Automation Agent manages its local scheduling and negotiates with the Task

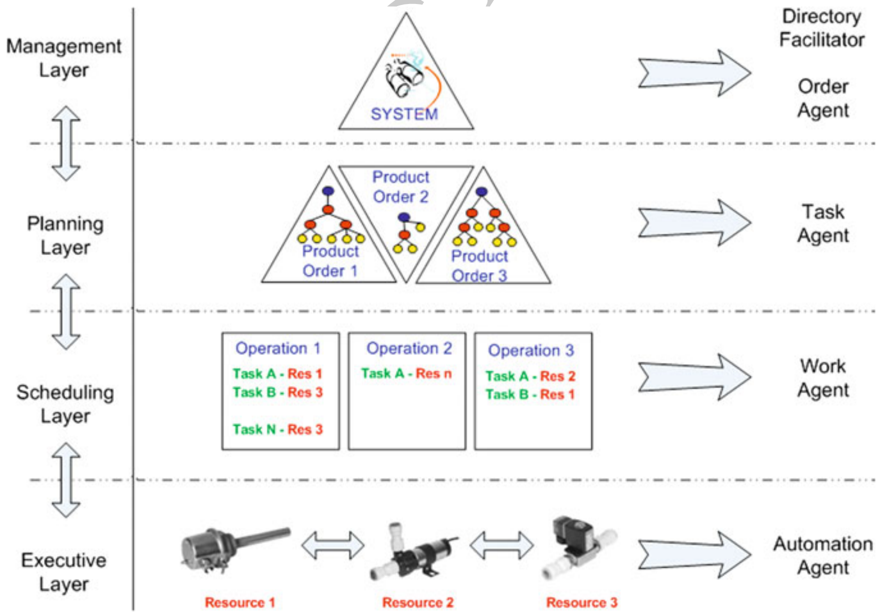

Fig. 1 Architecture of the developed MAS

Layout: T1 Standard SC Book ID: 272920_1_En Book ISBN: 978-3-642-37386-2
Chapter No.: 5 Date: 30-3-2013 Page: 6/11

Agent about supply and free timeslots in which the particular operations can be performed. One of the main features of the Automation Agent architecture is the distinction between the HLC and the LLC within each agent [22].

The HLC layer is in charge of higher level diagnostics, of the coordination with other agents and of adaptation based on the representation of the world. It incorporates an ontology-based world model that provides an explicit representation of the agent's immediate environment and supports reasoning about its state. The application of the ontology enables the systematic integration of different functionalities for the controlled process (i.e. in particular planning, scheduling and plant/unit control). Also this provides a common language for a better communication and creates a modular and internally consistent standard that can help reduce engineering costs [16].

The LLC is responsible for managing the physical component by using a limited set of reactive behaviors and requires means to access the physical I/Os of the controllers to gather sensor data as well as to issue commands to actuators. We base our LLC on the standard IEC 61499 [23], which is well-suited for distributed applications due to its event-driven execution model and offers a framework for the integration of run-time control and basic diagnostic applications. Our architecture was successfully used for controlling physical components that integrate "on the fly" reconfiguration abilities of the LLC [22], for monitoring and diagnostic tasks [24], as well as to enhance reconfigurability, robustness and fault tolerance of a transportation system [25].

Finally, the Directory Facilitator Agent (DF Agent) manages lists of Automation Agents and their services. Automation Agents register themselves in this list to be found by the Task Agent for the execution of tasks. In this context it is important to notice that Order Agent, Task Agent and Work Agent are designed likewise to the Automation Agents' HLC.

4 Rescheduling and Routing

The overall economic effectiveness of the entire manufacturing system is strongly coupled with the production schedule, since it needs to synchronize the entire system activities to achieve particular production goals. The application of distributed intelligent agent technology is considered as a promising approach and has been applied in manufacturing process planning and scheduling as well as the shop floor control domain [26].

4.1 Rescheduling

We use an event-driven rescheduling policy with rescheduling actions triggered upon the recognition of an exceptional event which could cause a significant

Layout: T1 Standard SC | Book ID: 272920_1_En | Book ISBN: 978-3-642-37386-2
Chapter No.: 5 | Date: 30-3-2013 | Page: 7/11

disruption of the system [27]. Rescheduling is started once the related Automation Agent notices a failure of its component and informs all related Work Agents about this issue. Each Work Agent reschedules assigned jobs that are already scheduled to the failed resource and excludes this component from participation in subsequent negotiations for future jobs until it is repaired. In order to maximize the overall system throughput as well as to minimize the flow time and make span, each Work Agent balances the jobs that need to be rescheduled between alternative components using the Contract Net Protocol [28]. Firstly, the Work Agent sequences all jobs originating from the agenda of the failed component according to their urgency, asks the DF agent for alternative resources for the particular job and sends a call for proposal to them. After the Work Agent has received the proposals for a particular job from the Automation Agents, it compares the proposals and awards the job to the best suited component. These steps are repeated until all jobs from the agenda of the failed component are rescheduled.

4.2 Dynamic Path Planning

Considering the complexity of process systems and their dynamic nature, where particular anomalies such as a component breakdown or an overload can cause that parts of the systems are not in function, the process of choosing the best route at such a specific moment can be difficult. Besides, when a component (pump, valve, etc.) breaks down, the related pipes are usually blocked and cannot be used until the component is repaired. If the system is inflexible and therefore not able to adequately reroute the medium using alternative routes, additional expenses are caused that again reduce the profitability of the production process. The route planning and best path algorithms have been highly researched topics in computer science for many years [29]. Based on Dijkstra's algorithm [30], we implemented a simple, but very efficient Shortest Path Algorithm, which is used by the Work agents for calculating an appropriate path in the case of a transport task. The laboratory process plant located in the Odo Struger Laboratory (see Fig. 2) is represented within the ontology.

The process system ontology is represented by an XML schema (Fig. 3) and a compatibility matrix. The main components such as tanks, valves and pumps are specified in the schema with their name, usage status and various other attributes providing further information for routing algorithms, business logic and monitoring applications. Constraints regarding media flow directions can be defined at any connection point between a pipe and a component. Combining this information with a media compatibility matrix enables advanced routing concepts and determines, if pipes need to be cleaned for routing incompatible media. Hence, the constraints of system components (e.g., pipe currently used or dirty as well as the compatibility between mediums) are also incorporated in the ontology and regulated through appropriate agent behaviors. The Work Agent is able to use the representation of the process plant and applies Dijkstra's algorithm to find the shortest usable path for performing the transport task.

Layout: T1 Standard SC | Book ID: 272920_1_En | Book ISBN: 978-3-642-37386-2
Chapter No.: 5 | Date: 30-3-2013 | Page: 8/11

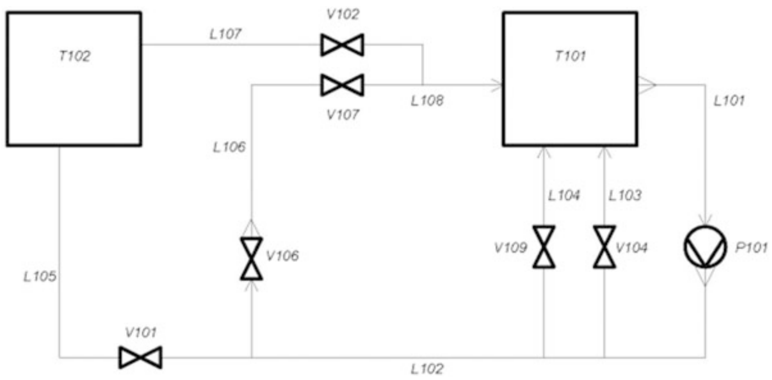

Fig. 2 Pipe and instrumentation diagram of the laboratory process plant


```xml
<ontology>
    <components>
        <tank name="T101" medium="opc:T101.MEDIUM" transferState="opc:T101.TRANS_STATE" functionalState="opc:T101.FUNC_STATE" capacity="10" currentLevel="opc:T101.LEVEL" unit=""/
        ...
        <valve name="V109" medium="opc:V109.MEDIUM" transferState="opc:V109.TRANS_STATE" functionalState="opc:V109.FUNC_STATE" serviceState="opc:V109.SERV_STATE" routing="opc:V1
        <pump name="P101" medium="opc:P101.MEDIUM" transferState="opc:P101.TRANS_STATE" functionalState="opc:P101.FUNC_STATE" serviceState="opc:P101.SERV_STATE" routing="opc:P101
    </components>
    <pipes>
        <pipe name="L101" medium="opc:L101.MEDIUM" transferState="opc:L101.TRANS_STATE" length="45" diameter="1.5" routing="opc:L101.ROUTING" unit="">
            <link direction="in">T101</link>
            <link direction="out">P101</link>
        </pipe>
        ...
        <pipe name="L107" medium="opc:L107.MEDIUM" transferState="opc:L107.TRANS_STATE" length="30" diameter="1.5" routing="opc:L107.ROUTING" unit="">
            <link>T102</link>
            <link>V102</link>
        </pipe>
        <pipe name="L108" medium="opc:L108.MEDIUM" transferState="opc:L108.TRANS_STATE" length="45" diameter="1.5" routing="opc:L108.ROUTING" unit="">
            <link>V102</link>
            <link>V107</link>
            <link direction="out">T101</link>
        </pipe>
    </pipes>
    <compatibility file="./CM-compatibility.csv" />
</ontology>
```


Fig. 3 XML schema of the system ontology

5 Conclusion and Outlook

In this chapter we presented a system architecture that integrates various types of agents in a consistent architecture able to manage an entire batch process system. The agents cooperate and coordinate their activities in order to accomplish particular orders. As a core of our architecture, we integrated a concept of Automation Agents that offers the opportunity of a new approach focusing on the definition of system components with every component having its own intelligence. Considering that deviations from an original production schedule happen more and more often in the present dynamic production environment, we integrated an appropriate rescheduling mechanism able to trigger rescheduling and in some cases the reconfiguration of the system. Additionally, this architecture enables a fast reaction on component breakdowns or traffic jams and, by identifying alternative routes and the consideration of new conditions and constraints, an adequate reconfiguration of the system as well as the rerouting of material.

The presented MAS architecture is currently evaluated on an existing laboratory process plant located in the Odo Struger Laboratory at the Automation and Control Institute. In our ongoing work, we are measuring and evaluating the performance of the presented architecture on this real system and will compare the results with similar approaches performed with other architectures. Furthermore, we tend to integrate our architecture with the real batch management software zenon[1] developed by COPA-DATA.

Acknowledgments This work has been supported by the Austrian Research Promotion Agency (FFG) project PrOnto (829576) under the BRIDGE program.

References

AQ2

1. S. Jämsä-Jounela, Future trends in process automation. Annu. Rev. Control **31**(2), 211–220 (2007)
2. T. Samad, P. McLaughlin, J. Lu, System architecture for process automation: review and trends. J. Process Control **17**(3), 191–201 (2007)
3. R. Brennan, Toward real-time distributed intelligent control: a survey of research themes and applications systems. IEEE Trans. Man Cybern. Part C Appl. Rev. **37**(5), 744–765 (2007)
4. V. Venkatasubramanian, C. Zhao, G. Joglekar, A. Jain, L. Hailemariam, P. Suresh, P. Akkisetty, K. Morris, G. Reklaitis, Ontological informatics infrastructure for pharmaceutical product development and manufacturing. Comput. Chem. Eng. **30**, 1482–1496 (2006)
5. T. Hamaguchi, T. Hattori, M. Sakamoto, H. Eguchi, Y. Hashimoto, T. Itoh, in *Multi-Agent Structure for Batch Process Control*. Proceedings of the IEEE International Conference on Control Applications, vol. 2 (2004), pp. 1090–1095
6. N. Jennings, S. Bussmann, Agent-based control systems: why are they suited to engineering complex systems? Control Syst. Mag. IEEE **23**(3), 61–73 (2003)
7. J. Reaidy, P. Massotte, D. Diep, Comparison of negotiation protocols in dynamic agent-based manufacturing systems. Int. J. Prod. Econ. **99**, 117–130 (2006)
8. C.A. Floudas, X. Lin, Continuous-time versus discrete-time approaches for scheduling of chemical processes: a review. Comput. Chem. Eng. **28**, 2109–2129 (2004)
9. H. Aytug, M.A. Lawley, K. McKay, S. Mohan, R. Uzsoy, Executing production schedules in the face of uncertainties: a review and some future directions. Eur. J. Oper. Res. **161**, 86–110 (2005)
10. T. Pirttioja, A. Pakonen, I. Seilonen, A. Halme, K. Koskinen, in *Multi-Agent Based Information Access Services for Condition Monitoring in Process Automation*. Proceedings of the 3rd IEEE International Conference on Industrial Informatics (INDIN '05), (2005), pp. 240–245

[1] http://www.copadata.com/

11. B. Sahovic, Automation Agents with a Reflective World Model for Batch Process Automation, Master Thesis, Technische Universität Wien (2010), Online: http://media.obvsg.at/AC07808679
12. N. Chokshi, D. McFarlane, *A Distributed Coordination Approach to Reconfigurable Process Control* (Springer, London, 2008)
13. R. Srinivasan, Artificial intelligence methodologies for agile refining: an overview. Knowl. Inf. Syst. **12**, 129–145 (2007)
14. M.B. Sesen, P. Suresh, R. Banares-Alcantara, V. Venkatasubramanian, An ontological framework for automated regulatory compliance in pharmaceutical manufacturing. Comput. Chem. Eng. **34**, 1155–1169 (2010)
15. R. Batres, M. West, D. Leal, D. Price, Y. Naka, An upper ontology based on ISO 15926. Comput. Aided Chem. Eng. **20**, 1543–1548 (2005)
16. E. Muñoz, A. Espuña, L. Puigjaner, Towards an ontological infrastructure for chemical batch process management. Comput. Chem. Eng. **34**(5), 668–682 (2010)
17. M. Obitko, V. Mařík, in *Ontologies for Multi-Agent Systems in Manufacturing Domain*. Proceedings of the 13th International Workshop on Database and Expert Systems Applications (DEXA '02), (2002)
18. X. Hong, S. Jiancheng, in *Multi-Agent Based Scheduling for Batch Process*. Proceedings of the 8th International Conference on Electronic Measurement and Instruments (ICEMI '07), (2007), pp. 2-464–2-467
19. I. Seilonen, K. Koskinen, T. Pirttioja, P. Appelqvist, A. Halme, in *Reactive and Deliberative Control and Cooperation in Multi-Agent System Based Process Automation*. Proceedings of the IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA '05) (2005), pp. 469–474
20. Z. Li, M. Ierapetritou, Process scheduling under uncertainty: review and challenges. Comput. Chem. Eng. **32**(4–5), 715–727 (2008)
21. A. Bonfill, A. Espuña, L. Puigjaner, Proactive approach to address the uncertainty in short-term scheduling. Comput. Chem. Eng. **32**, 1689–1706 (2008)
22. W. Lepuschitz, A. Zoitl, M. Vallée, M. Merdan, Towards self reconfiguration of manufacturing systems using automation agents. IEEE Trans. Syst. Man Cybern. Part C Appl. Rev. **41**(1), 52–69 (2011)
23. A. Zoitl, Real-Time Execution for IEC 61499, ISA (2008)
24. M. Merdan, M. Vallée, W. Lepuschitz, A. Zoitl, Monitoring and diagnostics of industrial systems using automation agents. Int. J. Prod. Res. **49**(5), 1497–1509 (2011)
25. M. Vallée, M. Merdan, W. Lepuschitz, G. Koppensteiner, in *Decentralized Reconfiguration of a Flexible Transportation System*, Industrial Informatics, IEEE Trans. (2011 in press)
26. W. Shen, Q. Hao, H.J. Yoon, D.H. Norrie, Applications of agent-based systems in intelligent manufacturing: an updated review. Adv. Eng. Inform. **20**, 415–431 (2006)
27. D. Ouelhadj, S. Petrovic, A survey of dynamic scheduling in manufacturing systems. J. Sched. **12**, 417–431 (2009)
28. R. Smith, The contract net protocol: high-level communication and control in a distributed problem solver (1981)
29. S. Russell, P. Norvig, *Artificial Intelligence: A Modern Approach* (Prentice Hall/Pearson Education, Englewood Cliffs, 2003)
30. E.W. Dijkstra, A note on two problems in connexion with graphs. Numer. Math. **1**, 269271 (1959)
31. Q. Guo, M. Zhang, An agent-oriented approach to resolve scheduling optimization in intelligent manufacturing. Robot. Comput. Integr. Manufact. **26**(1), 39–45 (2010)
32. Y.S. Ng, R. Srinivasan, Multi-agent based collaborative fault detection and identification in chemical processes. Eng. Appl. Artif. Intell. **23**(6), 934–949 (2010)
33. I. Seilonen, T. Pirttioja, K. Koskinen, Extending process automation systems with multi-agent techniques. Eng. Appl. Artif. Intell. **22**(7), 1056–1067 (2009)
34. IEC TC65/WG6, IEC 61499, Function blocks for industrial-process measurement and control systems—Parts 1–4. International Electrotechnical Commission [IEC], Geneva (2004)

35. IEC TC65, IEC 61512-1, Batch Control—Part 1: Models and Terminology. *International Electrotechnical Comission* (1997)
36. M. Merdan, W. Lepuschitz, B. Šahović, M. Vallée, in *Failure Detection and Recovery in the Batch Process Automation Domain Using Automation Agents*. Second International Conference on Advances in Computing, Control and Telecommunication Technologies (ACT), (Jakarta, Indonesia, 2010)
37. R.F. Babiceanu, F.F. Chen, R.H. Sturges, Real-time holonic scheduling of material handling operations in a dynamic manufacturing environment. Robot. Comput. Integr. Manufact. **21**, 328–337 (2005)
38. W. Lepuschitz, M. Vallée, A. Zoitl, M. Merdan, in *Online Reconfiguration of the Low Level Control for Automation Agents*. In Proceedings of the 36th Annual Conference of the IEEE Industrial Electronics Society (IECON ,10) (2010), pp. 1365–1370
39. W. Covanich, D. McFarlane, Assessing ease of reconfiguration of conventional and holonic manufacturing systems: approach and case study. Eng. Appl. Artif. Intell. **22**, 1015–1024 (2009)

Author Query Form

Book ID :	**272920_1_En**
Chapter No.:	**5**

Springer
the language of science

Please ensure you fill out your response to the queries raised below and return this form along with your corrections

Dear Author
During the process of typesetting your chapter, the following queries have arisen. Please check your typeset proof carefully against the queries listed below and mark the necessary changes either directly on the proof/online grid or in the 'Author's response' area provided below

Query Refs.	Details Required	Author's Response
AQ1	Please confirm the corresponding author is correctly identified and amend if necessary.	
AQ2	References [31–39] are given in list but not cited in text. Please cite in text or delete from list.	

MARKED PROOF

Please correct and return this set

Please use the proof correction marks shown below for all alterations and corrections. If you wish to return your proof by fax you should ensure that all amendments are written clearly in dark ink and are made well within the page margins.

Instruction to printer	*Textual mark*	*Marginal mark*
Leave unchanged	• • • under matter to remain	Ⓘ
Insert in text the matter indicated in the margin	⋏	New matter followed by ⋏ or ⋏⊗
Delete	/ through single character, rule or underline or ⊢—⊣ through all characters to be deleted	ꝺ or ꝺ⊗
Substitute character or substitute part of one or more word(s)	/ through letter or ⊢—⊣ through characters	new character / or new characters /
Change to italics	— under matter to be changed	⌙
Change to capitals	≡ under matter to be changed	≡
Change to small capitals	= under matter to be changed	=
Change to bold type	~ under matter to be changed	~
Change to bold italic	≂ under matter to be changed	≂
Change to lower case	Encircle matter to be changed	≢
Change italic to upright type	(As above)	⊥
Change bold to non-bold type	(As above)	⊥
Insert 'superior' character	/ through character or ⋏ where required	Y or ⋏ under character e.g. Y or ⋏
Insert 'inferior' character	(As above)	⋏ over character e.g. ⋏
Insert full stop	(As above)	⊙
Insert comma	(As above)	,
Insert single quotation marks	(As above)	Y or ⋏ and/or Y or ⋏
Insert double quotation marks	(As above)	Y or ⋏ and/or Y or ⋏
Insert hyphen	(As above)	⊢⊣
Start new paragraph	⌐	⌐
No new paragraph	↩	↩
Transpose	⊔⊓	⊔⊓
Close up	linking ⁀ characters	⁀
Insert or substitute space between characters or words	/ through character or ⋏ where required	Y
Reduce space between characters or words	| between characters or words affected	T