Empowering Adaptive Manufacturing with Interactive Diagnostics: A Multi-Agent Approach

Thomas M. Hubauer, Christoph Legat, and Christian Seitz

Abstract. This paper presents a novel approach towards proactive manufacturing control that integrates automated diagnostics with human interaction, resulting in a flexible adaptation of machine capabilities which helps to avoid damage in case of abnormalities. The model-based interpretation process supports predictive diagnostics using abductive reasoning, relying on plausibility thresholds and human intervention to resolve the resulting ambiguity between competing solutions. This enables the system to detect and avoid potential failure states before they actually occur. The proposed architecture additionally integrates intelligent products as mobile sensors, improving robustness and dependability of the production system.

1 Introduction

A growing demand for improved reactivity to market trends and towards provision of individualized products results in an ongoing trend from supplier-driven to customer-driven manufacturing. This results in higher flexibility requirements for manufacturing systems, namely (i) provision of flexible processes to enable small lot sizes and (ii) robustness to failures within the technical system. Especially the customization of products challenges traditional engineering approaches unattainably. For this reason, a number of projects explore new ways of addressing these requirements on different levels in accordance to the IEC 62264 [6] standard.

The application of agent-technologies within production systems is a lively research topic, recent examples include the ILIPT project [13] which focuses on enabling adaptable supply chains using market-based approaches, the EUPASS project [12] focusing on reorganizing automation systems dynamically using modular assembly components, and the PABADIS'PROMISE project [8] which aims at

Thomas M. Hubauer · Christoph Legat · Christian Seitz
Siemens AG - Corporate Technology - Intelligent Systems & Control - Agent Technologies
e-mail:{thomas.hubauer.ext,christoph.legat.ext,
ch.seitz}@siemens.com

Y. Demazeau et al. (Eds.): Adv. on Prac. Appl. of Agents and Mult. Sys., AISC 88, pp. 47–56.
springerlink.com

realizing a distributed manufacturing execution system including reconfiguration, but without using diagnostic information. Agent-based approaches have also been applied to real-time control of production systems [17], the capabilities of these agents are however very limited due to the hard real-time constraints that have to be addressed. Several other concepts from computer science have recently been adopted in the context of production system control, e. g. the development of modular material flow systems [3] based on the internet of things [7], and the use of diagnostics to enhance the detection of system failures which is explored in the MAGIC project [1]. In contrast to the approach presented here, the latter does not consider a direct interaction with the automation control system leading to an automated reaction and does not integrate intelligent products. Thus, the proposed approach provides a benefit over existing systems due to a tighter interaction between diagnostics, adaptation and product interaction facilitated by generic service description.

Product-driven manufacturing systems supported by intelligent products are a promising way to overcome limitations with respect to small lot sizes. An intelligent product enhances the currently used Auto-ID technologies [10], in which a computing entity is attached to (part of) a product during the whole product life cycle, collecting information and cooperating with other devices in its ambient environment. We go one step further and beyond existing approaches surveyed in [11] by (i) enabling the intelligent product to interact and cooperate with the production control system and (ii) alleviating the lack of integration between the shop floor control system and higher level supervisory operations accomplished by human personnel. Although no real-world production system operates completely unsupervised, there is to our best knowledge no other approach investigating the interaction of automated diagnostics and human supervision within an autonomous flexible production environment. The result of our approach is a highly flexible and robust framework for product-driven manufacturing based on cooperation of a model-based diagnostic system and human operators.

We demonstrate the concept in the context of a flexible production scenario enhanced with autonomous products. More specifically we consider a T-style component of a (directed) inner logistic system composed of one incoming conveyor, a switch and two distinguishable outgoing conveyors (typically leading to different machines). Each conveyor section is impelled by a live axle equipped with a sensor measuring axle motion. As some products may be sensitive to vibrations at certain time points during production (e. g. due to a freshly glued joint), products are equipped with a digital product memory and an acceleration sensor during production, storing acceleration measurements for quality assurance. However, this information can also be combined with other measurements to diagnose the conveying system: If the axle of a conveyor turns but a product located on the belt does not move, this indicates a transmission problem which renders this section of the conveyor unusable until repair. Irregular movement with sudden accelerations or vibrations may indicate bearing problems, which only excludes vibration-sensitive products from using this conveyor. Depending on whether the affected section is the shared incoming or one of the two available outgoing conveyors, the usability restrictions either hold for the complete conveying system or just for one of the two possible routes. These restrictions have to be incorporated into production planning to optimize machine usage and throughput while guaranteeing product quality.

The remainder of this paper is structured as follows. In Sect. 2 we present the conceptual architecture of the proposed multi-agent system and introduce the semantic models employed. Some selected details on the realization of the components are presented in Sect. 3, before we conclude the presentation of our ideas in Sect. 4.

2 Conceptual Architecture

This section presents the general architecture of the proposed multi-agent system. The functionality of the system is realized by agents implementing four roles which are detailed in the upcoming subsections: symptom provider, situation analyzer, production controller, and supervision mediator. The upper half of Fig. 1 depicts these roles and their associated meta-models. The realization of these roles by agents (which is focus of the successive Sect. 3) and exemplary concrete models are illustrated in the lower section of the same figure. Please note that in favor of conciseness we only depict concepts referenced in the remainder of this paper, rather than the complete models. The framework additionally provides a discovery service enabling agents to find interaction partners, implementing yellow pages functionality.

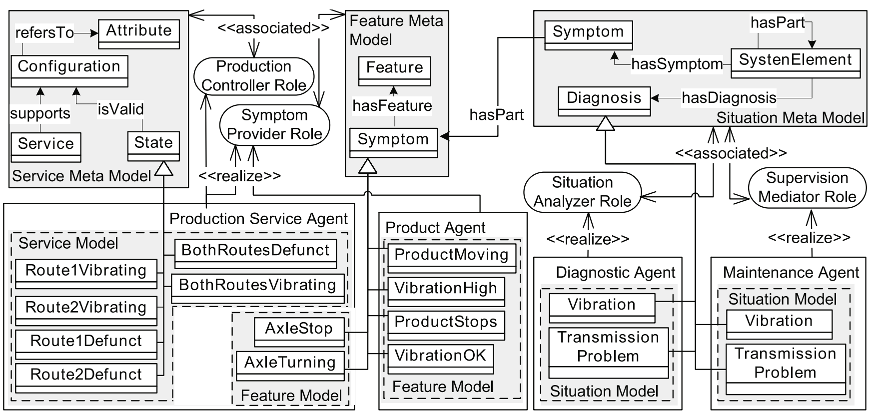

Fig. 1 Roles, Agents and corresponding Models

The meta-models incorporate the relevant domain knowledge ranging from process information to definitions of diagnoses, whereas the specific models encapsulate the knowledge about the concrete products and processes. The models are used for diagnostics, but also as a common vocabulary for service discovery bringing together information providers and consumers based on the type of information offered and requested. To facilitate automated information processing and to increase reusability and interchangeability of the model, it is employed with formal semantics based on the Web Ontology Language (OWL). More specifically, the situation meta model is based on $\mathscr{EL}$, a subset of lightweight profile OWL 2 EL [16] allowing for polynomial-time query answering, and the feature meta model uses the profile OWL 2 RL [16] to facilitate interpretation by a rule engine; both profiles were chosen to optimize the tradeoff between expressivity and performance w. r. t. the intended evaluation method.

2.1 Symptom Provider

A symptom provider (SP) realizes the gathering of `Features` within a given field of responsibility defined by a set of `SystemElements` of the proposed metamodel. To offer its functionality to the agent community, a SP registers with the discovery service by announcing the ids of the `SystemElements` it is responsible for. Consumers can register with a SP for certain event classes denoted by subconcepts of `Symptom`. Upon receiving data (a `Feature`) from an associated information source or preprocessing component, e. g. a piece of machinery, the SP automatically infers the corresponding `Symptom` based on its feature model. Subsequently, all consumers having subscribed for this event or any of its superconcepts are notified by the SP.

2.2 Situation Analyzer

The situation analyzer role (SA) encapsulates the process of interpreting symptoms based on the diagnosis meta model depicted in Fig. 1. To receive the necessary input data, the SA subscribes for the `Symptoms` it wants to be notified about as follows: First, it queries the discovery service for symptom providers responsible for the `SystemElements` allocated to this agent for surveillance. Then, the SA subscribes with each of them for the concrete symptoms (i. e. subconcepts of `Symptom`) of the associated component it is interested in.

Upon receiving a `Symptom` the SA triggers the interpretation process which determines a set of defects that might possibly cause the observations. By allowing to assume the presence of symptoms that have not been detected yet, the SA is even able to detect faults which have not yet fully manifested. Obviously a `Diagnosis` not requiring the assumption of additional symptoms will more likely apply than another one based on assumptions, the SA therefore attributes its interpretations with a measure of their plausibility. Due to a potentially high number of calls to its services, performance of a SA is a vital aspect. To ensure reactivity and performance, the proposed realization of the situation analyzer presented in Sect. 3.3 uses an anytime approach and a lower bound $pl_{\downarrow}$ on the quality of solutions.

Dependent on configurable thresholds pl_{μ} and pl_{Δ} the SA triggers both automated reactions and operator interaction: Firstly, the SM is informed about the complete set of diagnoses, their necessary assumptions and plausibility values. Then, if and only if the plausibility of the best `Diagnosis` determined by the SA exceeds

pl_μ and in the second-best alternative is at least by an amount of pl_Δ less plausible than the best one, the SA additionally commands the production controllers in charge for the components in question to automatically initiate protective steps, sending a notification about this decision to the supervision mediator (SM).

2.3 Production Controller

The production controller (PC) is responsible for the interaction with the automation system. It realizes higher-level management and control functionality of the associated `SystemElements`, which in turn provide specific capacities within the production system. The functionality realized by the PC is offered to higher level control systems (as e. g. production planning) as a service. In order to be discoverable, the PC registers with the discovery service (DS) as described in [9].

In case of a reliably (w. r. t. pl_μ and pl_Δ) identified malfunction of the system, the PC is informed about the `Diagnosis` by a situation analyzer (SA). This information enables the PC to restrict the production services it provides via the discovery service in order to reduce the probability of machine and product damage. The semantic model and the detailed failure state identified by the SA allow the PC to restrict as few machine capabilities as necessary, as compared to the complete temporal decommissioning of a machine in case of an emergency stop in reaction to an unspecific general failure state.

2.4 Supervision Mediator

The supervision mediator (SM) provides an interface between the agent-based automated control system and the human operator, integrating her into the diagnostics and control cycle: Using the SM, the operator assigns situation analyzers (SA) to selected components of the system and configures the plausibility threshold values pl_μ, pl_Δ (used by the production controller) and $pl_\downarrow$ (used by the situation analyzer). Upon detection of a set of plausible diagnoses by a SA, the SM enables the operator to inspect the alternatives, manually change interpretations, and revoke automated reactions if necessary. Additionally, the SM supports the operator in acquiring additional information not available by automated sensors and feeding it into the diagnostic process. The SM is therefore central to the proposed interactive approach to diagnostics in production.

3 Realization

The prototypal realization of the proposed framework in a real-world industrial production environment is depicted in Fig. 2. As to not disturb the real time communication of the automation system based PROFINET/Profibus, we decided to implement the communication of the multi-agent system using installed enterprise links based on TCP/IP. We use a specialized in-house multi-agent platform; the basic system could however be ported straightforwardly to other, public platforms since we only rely on services available in most available systems. The following subsections present selected aspects of the prototype system.

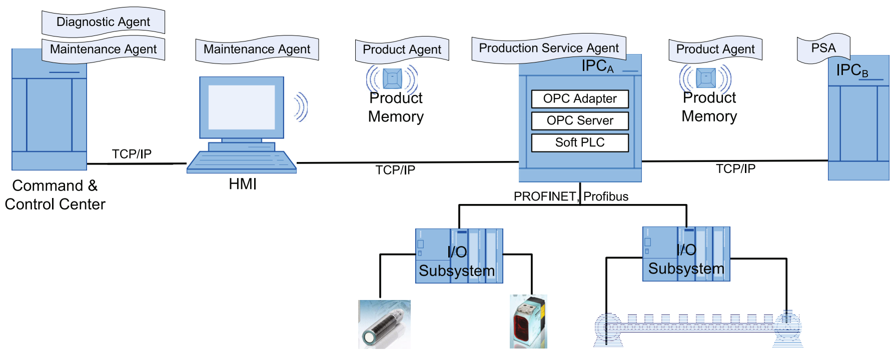

Fig. 2 Architectural overview of the realized system

3.1 Production Service Agent

A production service agent (PSA) acts both as a symptom provider and a production controller. In this latter role, the PSA allows the operator to configure the production control system, and it controls the production process to ensure that configurations which might damage the machinery of the product are avoided. In the symptom provider role, the PSA interprets data provided by machinery-mounted sensors yielding `Features`, which it combines subsequently using the feature model to derive `Symptoms`. This behavior of the PSA is realized straightforwardly by a rule engine, as the feature meta model and the feature models are restricted to OWL 2 RL. The necessary sensor and parametrization data can be accessed as process variables of the real-time control kernel the upcoming OPC UA Standard [5] based on a locally executed client-server protocol. All components including the OPC UA infrastructure and the agent itself are realized on an industrial PC (IPC) equipped with a TPC/IP communication link and a software programmable logic controller (PLC) that exercises the real-time control as shown in Fig. 2.

In the running example, the PSA allotted to the exemplary transportation system uses the axle motion sensors to derive one of the symptoms `AxleStop` or `AxleTurning` depicted in Fig. 1, which is then sent to the diagnostic agent (DA) registered for the respective symptom. If the PSA is notified by a DA about a plausible `Diagnosis`, the actual `State` is derived in order to to restrict the set of `Configurations` of the production service which are announced to the higher-level planning systems.

3.2 Product Agent

A product memory is an embedded device attached to a product during its complete life-cycle and equipped with sensors to autonomously observe the ambient environment of the product. A product agent (PA) implementing the symptom provider role controls the product memory and makes its measurements available to the production control system as `Symptoms`, which can be mapped to `SystemElements` based on product location. `Symptoms` are derived from `Features` based on the feature model using a rule engine installed on the product memory; details on this embedded reasoning approach can be found in [15]. Figure 3(a) depicts the conceptual architecture of a product memory, a photograph of a prototype is shown to its right in Fig. 3(b).

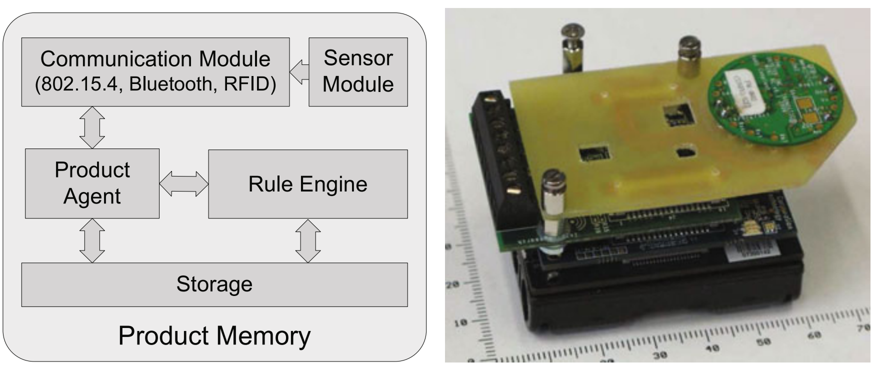

Fig. 3 Architecture of a digital product memory (a) and photograph of a prototype (b)

Regarding the exemplary inner logistics component, the PA uses the local feature model to interpret the measurements of its onboard 3-axis motion sensor, determining whether or not the product is moving (`ProductMoving` or `ProductStops`), and whether the motion trajectories indicate excessive vibrations that might damage the product (`VibrationHigh`). The `Symptoms` are sent to subscribed diagnostic agents as they are detected.

3.3 Diagnostic Agent

The main task of a diagnostic agent (DA) implementing the situation analyzer role is knowledge-based diagnostic reasoning which has been a vivid research topic over years, leading to a multitude of approaches (see [2] for a survey). The current implementation uses logic-based abduction [14] for this task as abductive reasoning naturally handles incomplete information, enabling predictive diagnostics. More specifically, we use a method extending the ideas presented in [4] to determine a set of plausible diagnoses along with the assumptions they require and the resulting plausibility score. Omitting most technical detail due to space limitations, diagnosis generation can be seen as finding optimal paths in a hypergraph, where each path represents a valid derivation. The structure of the graph is determined by the models, the observed and assumed `Symptoms`, and the set of possible diagnoses; its size is polynomial in the size of the situation model due to the restrictions on the representation language. The plausibility of a path depends on two factors, namely the observations it explains and the assumptions it requires to be made, inducing a partial order. As the number of paths can be exponential, we use an incremental anytime algorithm to determine them one by one in order of decreasing plausibility, stopping as new information arrives or the lower plausibility bound $pl_{\downarrow}$ is reached. As delineated in Sect. 2.2, the parameters pl_{μ} and pl_{Δ} determine whether an instant automated reaction is taken via the production service agent, additionally the complete set of competing diagnoses is provided to the maintenance agent along with information on required assumptions and plausibility.

In the running example, assume that a product located on outgoing belt 1 of the logistics component detects increased vibration values (`Vibration`) while the respective component signals a turning axle (`AxleTurning`). The most straightforward explanation for both observations requiring no assumptions then states that only this route of the logistic system vibrates (`Route1Vibrating`). Alternatively, assuming for example that the shared incoming belt does not run smoothly either gives rise to the diagnosis `BothRoutesVibrating`.

3.4 Maintenance Agent

Maintenance agents (MA) realize user interaction as defined by the supervision mediator role. To this end, a MA offers two separate graphical user interfaces, the supervision view integrated into the command and control center of the factory, and the maintenance view built into the SCADA system WinCC. The former allows the operator to assign diagnostic agents (DA) to components, set the thresholds pl_{μ}, pl_{Δ} and $pl_{\downarrow}$, introspect diagnoses derived by the DA and manually pick one of them, in which case the new diagnosis is signaled back to the DA where it is forwarded to the responsible production service agent. The maintenance gui located at the machine control panel supports the operator or technician in sharpening the result of the analysis by pointing out missing relevant data (determined from the assumptions created during reasoning), and feeding measurements made by the operator back into the diagnostic process.

In context of the running example, the operator might for example decide to make sure the other belts to not exceed vibration limits by visual inspection to reduce the risk of products being damaged, using the maintenance gui.

4 Conclusions and Future Work

We have presented an architecture for an agent-based flexible production system which integrates a service-based approach to production with intelligent products and interactive diagnostics of the production machinery. We extend previous work on using diagnostic reasoning in automated production control by integrating mobile intelligent products and by enabling a human operator to directly interact with the diagnostic agent. Here, digital product memories serve as additional mobile sensor units within the production system providing both product- and machine-centered data. Using a Human-in-the-Loop approach to interactive diagnostic reasoning solves several prominent problems of automated diagnostics: Firstly, it alleviates the problem of decision making in the context of a huge amount of possibly imperfect data by automated pre-processing and default decisions which can be modified due to human experience. Furthermore, lack of information is handled by enabling the diagnostic system to draw hypothetical conclusions which are then validated or refuted on the basis of additional data provided by the operator. Moreover, the proposed system is based on real industrial automation systems and hardware, thus providing a direct integration path into existing products.

We are currently building a prototype system. Major challenges include optimizing memory consumption of the embedded reasoner (e. g. by forgetting facts), fast sensor data provision from real-time control systems using OPC, and addressing real-time requirements in reconfiguration. For the future, we intend to integrate model-based planning algorithms into the production service agent to realize optimized configuration scheduling and distributed production planning, and market-based mechanisms for the coordination of production service agents and product agents. Another interesting topic of research is the (semi-)automated extraction of the required models from the huge amounts of engineering information produced during factory planning and available e. g. in plant lifecycle management systems, which could significantly reduce the effort of implementing such a model-based approach. Another idea is to extend the approach towards more expressive models and complex structures such as groups of factories and supply chains. Using a hierarchical approach, diagnoses determined for a factory can serve as symptoms for the whole supply chain, allowing to infer even more information which can be employed to reduce production loss. Finally, we intend to evaluate system performance as well as quality of diagnoses and automated reactions.

References

1. Albert, M., Längle, T., Wörn, H., Capobianco, M., Brighenti, A.: Multi-agent systems for industrial diagnostics. In: Staroswiecki, M., Wu, N. (eds.) Proceedings of the 5th Symposium on Fault Detection, Supervision and Safety of Technical Processes 2003, vol. 1, pp. 459–464. IFAC, Elsevier Science Ltd., Washington D.C., USA (2003)
2. Dressler, O., Puppe, F.: Knowledge-based diagnosis - survey and future directions. In: Puppe, F. (ed.) XPS 1999. LNCS (LNAI), vol. 1570, pp. 24–46. Springer, Heidelberg (1999)
3. Günthner, W., ten Hompel, M. (eds.): Internet der Dinge in der Intralogistik. Springer, Heidelberg (2010) (in german)

4. Hubauer, T.M., Lamparter, S., Pirker, M.: Automata-based abduction for tractable diagnosis. In: Haarslev, V., Toman, D., Weddell, G.E. (eds.) Proceedings of the DL Home 23rd International Workshop on Description Logics (DL 2010), CEUR Workshop Proceedings, pp. 360–371. CEUR-WS.org., Waterloo, CA (2010)
5. International Electrotechnical Commission. IEC 62541 - OPC Unified Architecture Specification
6. International Electrotechnical Commission. IEC 62264 - Enterprise-control system integration (2003)
7. ITU Strategy and Policy Unit: ITU Internet Reports 2005. The Internet of Things. Itu internet reports, ITU (2005)
8. Kühnle, H. (ed.): Distributed Manufacturing - Paradigm, Concepts, Solutions and Examples. Springer, Heidelberg (2010)
9. Legat, C., Lamparter, S., Seitz, C.: Service-oriented product-driven manufacturing. In: Proceedings of the 10th IFAC Workshop on Intelligent Manufacturing Systems (IMS 2010), pp. 161–166 (2010)
10. McFarlane, D.: Auto-ID based control - an overview. Whitepaper, Auto-ID Centre. University of Cambridge, UK (2002)
11. Meyer, G.G., Främling, K., Holmström, J.: Intelligent products: A survey. Computer in Industry 60(3), 137–148 (2009)
12. Onori, M., Barata, J., Frei, R.: Evolvable assembly systems basic principles. In: Proceedings of the 7th IFIP BASYS 2006, Ontario, Canada, vol. 220, pp. 317–328 (2006)
13. Parry, G., Graves, A. (eds.): Build To Order - The Road to the 5-Day Car. Springer, Heidelberg (2008)
14. Paul, G.: Approaches to abductive reasoning: An overview. Artificial Intelligence Review 7(2), 109–152 (1993)
15. Seitz, C., Lamparter, S., Schoeler, T., Pirker, M.: Embedded rule-based reasoning for digital product memories. In: Proceedings of the 2010 AAAI Spring Symposium. AAAI Press, Palo Alto (2010)
16. W3C OWL Working Group: OWL 2 Web Ontology Language: Profiles. W3C Recommendation (2009), http://www.w3.org/TR/owl2-profiles/
17. Wannagat, A., Vogel-Heuser, B.: Agent oriented software-development for networked embedded systems with real time and dependability requirements in the domain of automation. In: Chung, M.J., Misra, P. (eds.) Proceedings of the 17th International Federation of Autonomic Comtrol World Congress, 1. International Federation of Automation Control (IFAC). Elsevier Science Ltd., Seoul (2008), http://www.ifac2008.org/