# A Decentralized Resource Monitoring System Using Structural, Context and Process Information

Lisa Abele[1], Lisa Ollinger[2], Ines Heck[2], and Martin Kleinsteuber[3]

[1] Siemens AG, Otto-Hahn-Ring, München, Germany
[2] DFKI GmbH, Trippstadter Straße 122, Kaiserslautern, Germany
[3] Department of Electrical Engineering, Technische Universität München

**Abstract.** Over the past century there has been a dramatic increase in the consumption of resources such as energy, raw materials, water, etc. in the manufacturing domain. An intelligent resource monitoring system that uses structural, context and process information of the plant can deliver more accurate monitoring results that can be used to detect excessive resource consumption. Recent monitoring systems usually run on a central unit. However, modern plants require a higher degree of reusability and adaptability which can be achieved by several monitoring units running on decentralized autonomous devices that allow the components to monitor themselves.

To integrate structural, context and process information on such autonomous devices for resource monitoring, semantic models and rules are appropriate. This paper will present an architecture of a decentralized, intelligent resource monitoring system which uses structural, context and process knowledge to compute the state of the individual components by means of models and rules. This architecture might also be used for other manufacturing systems such as diagnostic or prognostic systems.

## 1 Introduction

An efficient use of resources in industrial plants is becoming increasingly important. Plant engineers have to be aware of the resource consumption – e.g. energy, raw materials, water, compressed air, etc. – of their plants on the level of the incorporated devices so that they are able to optimize the plants' structure and processes accordingly. A resource monitoring system (RMS) is needed to use resources efficiently. Nowadays, monitoring systems run on a central unit that collects all sensor data of the plant to compute the monitoring states of the different components. But modern plants require a higher degree of reusability and adaptability and thus a decentralized monitoring system where components are able to monitor themselves by means of intelligent autonomous devices such as active digital product memories (ADPMs). The advantage of such a decentralized system with ADPMs are manifold: (1) the manufacturer can produce intelligent components that can monitor themselves, using his extensive knowledge about his products, (2) the components can also execute additional monitoring rules defined by the plant engineer, (3) an exchange of single components does not require modification of the entire system and downtime, (4) failure of one unit of the RMS will not affect the operation of the entire RMS.

A RMS is more advanced as usual Condition Monitoring Systems and requires not only measurement of sensor data, but also an awareness of the components' environment and situation. This additional information is stored in models which explicitly store knowledge about the plant structure, the process steps and the plant context. Thus, it gets possible to reuse this information and new monitoring systems can be defined with lower effort.

The objective of this paper is to describe the architecture of a monitoring system implemented on ADPMs that combines structural, process and context information to allow a manual optimization of the control system of the plant. Especially, research to build a monitoring system with explicit knowledge-based models and logical rules [3] is derived from the experiences gained during design of an application scenario of an industrial plant within the RES-COM project [2]. This research project aims to automatically conserve resources in industrial plants through ADPMs and context-aware embedded sensor-actuator systems.

In this paper we introduce the system architecture of the monitoring system. Then, a detailed description of the structure, process and context information is given. Finally, we show first results with an application scenario addressing issues that can be solved by our system.

## 2 Related Work

The technologies that are relevant to our research fall in three categories: knowledge-based monitoring systems, industrial applications with DPMs (active and passive digital product memories) and automation systems using structural, context or process knowledge.

Typical knowledge-based monitoring solutions focus either on specific application areas (e.g. electro hydraulic linear drives [14]) or on the deployment of certain tools or methods (e.g. computational intelligence methods [17]). Our goal of research is to develop a generic, tool- and facility-independent RMS. The authors in [11] describe how a generic data exchange format can be used for an automatic configuration of a production monitoring and control system. But the usage of a data exchange format such as CAEX requires tool support. Currently, no wide-spread commercial tool supports CAEX directly or via converters. Some dedicated tools, e.g. the AutomationML Editor of Zühlke Engineering AG [7], offer currently only basic features.

Several industrial applications use DPMs to attach relevant information to plant products or components. In [15] the authors describe how to attach life cycle information to an industrial product to allow an information handover via several stages of the value chain with potentially different stakeholders. One of the stakeholders is the plant engineer. Based on this approach, he can extract the monitoring characteristics of an industrial product out of the life cycle information stored on the component. The authors in [13] present a flexible approach for product-driven manufacturing using a digital product memory. This flexible approach describes a scenario in which the DPM is attached to the product to control the environment and influence the entire production process. The main ideas of the existing approaches based on DPMs were continued and enhanced to define the architecture of the RMS.

A considerable amount of literature has been published on automation systems that use either structural, context or process knowledge or even a combination of them. A plant-wide diagnosis systems is presented in [4] using process knowledge in addition to structural knowledge. An infrastructure which describes context modeling concepts for pervasive computing systems was proposed by [8]. We discovered that much of the work in the field of context awareness [1] is concerned with providing either a framework to support the abstraction of context information from the field level of the plant or high-level models of context information to provide context services. Our approach combines these two levels to provide more accurate context information to the monitoring system. All the studies reviewed so far, however, are not combining all knowledge about the plant to efficiently use them for monitoring.

## 3 System architecture

Monitoring of resources in a production plant requires sufficient information granularity and clearly structured data to compute reliable monitoring states of components. One of the main advantage of our architecture compared to the current state-of-the-art is its decentralized character. This means that the manufacturers of components produce intelligent components that can monitor themselves. The main modules of the RMS as shown in figure 1 are:

1. Every component in the plant is equipped with an ADPM including a monitoring unit (MU). An ADPM consists of a knowledge base that contains knowledge about the plant in a machine processable way, including a rule base. For our monitoring purpose, we distinguish between three kinds of input, 1) structural information about the plant, e.g. motor is monitored by a temperature sensor and a smart meter, 2) current process step, e.g. conveyor that is driven by the motor is executing process “transport at maximum speed”, 3) context information of the plant, e.g. plant is producing at half load. This input and the knowledge-based models and rules are then used by an inference engine in the monitoring unit which computes the component state.
2. Collections of components are combined to groups. Every group has its own decentralized MU which takes the individual component states as input to compute a composite state of the group.
3. The heart of the system is the monitoring unit of the industrial plant. It gets component and group states as input to compute the state of the entire plant. A knowledge base editor assists here in the addition of new knowledge from the plant engineers and performs consistency checks on the updated knowledge base.
4. The plant monitoring state computed by the MU of the plant is provided to the plant engineer. Based on the resulting states provided by the monitoring system the plant engineers react to optimize the resource efficiency of the entire plant. Thus, the control system of the plant has to be adapted in order to optimize the parameterization of the components and the control procedures. The control procedure in form of a service orchestration (section 3.2) provides a high degree of adaptability so that adaptations can be realized with low effort [10].

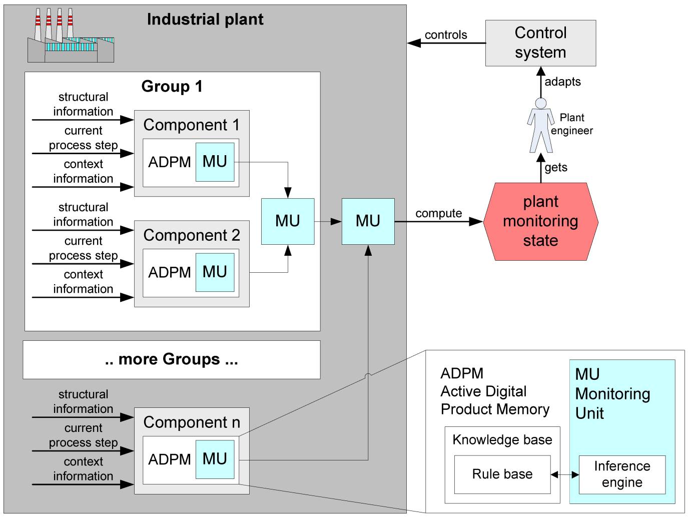


*Fig. 1:* Architecture of the monitoring system

In the following sections we'll describe the different input information needed for the monitoring task in more detail.

### 3.1 Structural information

To allow reusability and extensibility the structure model of an industrial plant has to be able to include all knowledge of previous knowledge-bases such as wide-spread modeling tools (e.g. Siemens COMOS) and additional knowledge.

The structure model defines two kinds of fundamental information: (1) the *taxonomy* identifies and names element classes and arranges them into a classification hierarchy and (2) the *plant topology* describes the *containment hierarchy* of plant components as defined by the *part of* relation and other functional relations between components as *connected to* or *energy flow*.

The component models also contain individual characteristics of the components including default parameters, e.g. nominal energy, monitoring thresholds and configuration parameters. The default parameters and default monitoring thresholds are stored by the manufacturer of the components on their ADPMs. During plant design, the plant engineer configures the individual components according to the plant environment and stores the configuration parameters on their ADPMs.

### 3.2 Process information

To determine the monitoring state of components, the current process of the component or remote components has to be considered. The required process information can be divided in two parts: a model of the production process and the dynamic information about the current process state. Today, this information is contained implicitly in the control procedures of the process control devices like programmable logic controllers (PLCs). Since the control procedures of these controllers are generated on a low implementation level, where the individual binary in- and outputs are processed, the code is complex and monolithic. Due to this the control procedure lacks of clarity, comprehensibility and adaptability. Therefore, this procedure should be developed on a higher abstraction level such as service orchestration [16].

This means that functions of the hardware components are regarded as services that represent the building blocks for the execution of the production process. To get an executable control procedure the services have to be arranged within a process logic in a formal representation The process logic contains states and state transitions with links to the respective services. Thus, the model of the production process stored in the knowledge base can be derived directly from this process logic. During run-time, the control system has to indicate the current active process step and provide process values and dynamic information to the monitoring unit.

### 3.3 Context information

To monitor the resource consumption of a production plant or its components, it is not enough to observe just the actual consumption values. An assessment whether the measured values are within acceptable range often depends on the context, for example what kind of product the machine is currently producing or what the average energy consumption was for the last products. To determine the current resource situation of the plant, this additional information has to be evaluated continuously and has to be represented in the underlying monitoring rules.

According to [6], context "is any information that can be used to characterize the situation of an entity. An entity is a person, place or object that is considered relevant to the interaction between a user and an application.". A monitoring system can make more intelligent decisions and determine more accurately the situation of the monitored item if additional information (=Context) is provided about all entities that are in some way relevant to the current monitoring task.

In our RMS, the plant components and their monitoring units are supplied with context information from a central server, the *context broker*, that collects the context information from all participating sources in the plant. The context broker supports several protocols (e.g. OPC UA, Web services, REST) to call the technical interface of the ADPM, read the enabled variables and store the values in its internal database. The individual MUs can request and register all context information that is relevant to their specific monitoring task according to their underlying context and situation model. This context information is used by the MUs to compute the individual monitoring state of a component or a group of components.

### 3.4 Decision support system

The monitoring units of the components contain a knowledge base with a rule base and an inference engine. Together they form a decentralized decision support system (DSS). The MU uses the structure, process and context models to annotate sensor data semantically and computes the state of the component. For example, take a group *g1* that contains a smart meter sm1 that measures the energy consumption of a motor *m1* and reports a value of 50. Then MU of *g1* annotates this as “motor m1 has an energy consumption of 50kW that is within acceptable range”.

We distinguish between two kinds of rules: (a) rules that infer the states of single components, (b) composite rules that infer the states of groups including context and process information. The simplest case of (a) is to use thresholds stored by the manufacturer of the components on the ADPMs to compute the current state.

For composite components or groups, the system computes composite rules, a simple example for case (b) is *if one of the motors in the motor group G has the state “error” then the state of G is “error”*. Then the DSS provides the states of the groups and the annotated data to other composite monitoring units and finally it computes the state of the entire plant.

## 4 Application Scenario

In the context of the RES-COM project, we will implement the RMS on an intelligent plant which produces smart key finders as shown in figure 2. We build a simulation tool based on this application scenario. Let us consider a concrete simulation of the monitoring approach, using the transportation block as example. The transportation block of the plant is used to transport the smart key finder on two conveyors *cv1* and *cv2* which are both running with the same constant speed. The two conveyors are driven by the engines *m1* and *m2* respectively. The entire transportation unit is grouped in the functional drive group *g*.

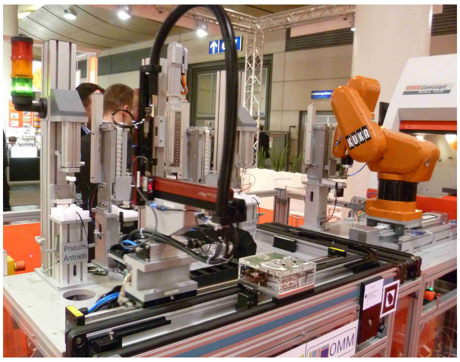


*Fig. 2:* Smart key finder production plant

In our simulation tool, a plant engineer can enter monitoring rules in a user interface as shown in figure 3. Parameters of the plant components, e.g. the resistance of the motor m1 can be addressed with “m1.R”. Thus, the plant engineer can insert complex equations.

Simple rules monitor thresholds of components which are stored by the manufacturer on the ADPMs, e.g. the maximum input power of the motor m1 is 500W. Additionally, the process information is considered. The drive group $G$ can execute three different processes: a) drive forward, b) drive backward, c) stop. In this application scenario, we can use the process information to compute the standby voltage $V_s$ of the group g. When the motor is in process “stop” a small standby voltage is acceptable.

A special feature of the plant is that the operator of the plant can choose between two production contexts: a) produce with lowest energy consumption, b) produce with minimum delivery time. If the operator chooses the first option (*low energy*), the engines’ rotational speed is adapted to reach the maximum energy efficiency and the two conveyors run at a lower speed. If the operator chooses the second option (*min delivery time*), the engines’ rotational speed is adapted to its maximum.

The knowledge-based models were stored in OWL axioms which allows inference mechanisms. We used *Prolog* as inference engine which stands for *programming in logic*. The rules are processed by the inference engine and finally, the resource monitoring system presents the resulting monitoring states to the plant engineer.

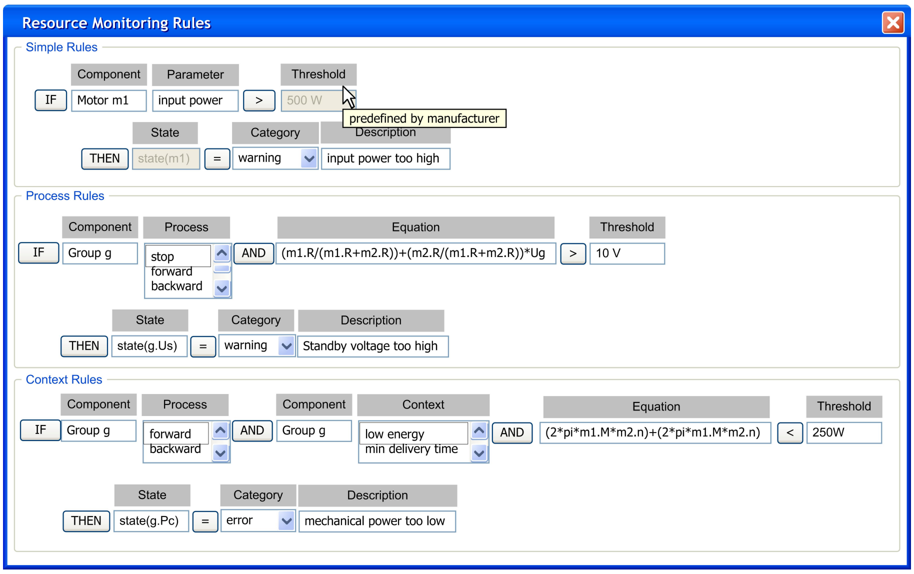


*Fig. 3:* Resource monitoring rule interface (R = resistance, Ug = measured voltage, Us = standby voltage, pi = $\pi$, M = torque, n = rotational speed)

## 5 Conclusion

In this paper, we have presented a decentralized architecture of a resource monitoring system. In our research, the aim was to describe how structural, process and context models and monitoring rules on ADPMs can be used to provide a high degree of adaptability and reusability. As a result, we implemented the resource monitoring system for a smart key finder production plant based on the proposed architecture.

## 6 Acknowledgment

This research was funded in part by the German Federal Ministry of Education and Research under grant number 01IA11001. The responsibility for this publication lies with the authors.

## References


1. M. Baldauf, S. Dustdar, and F. Rosenberg. A survey on context-aware systems. *Information Systems*, 2(4), 2007.
2. RES-COM - Resource Conservation through Context-dependent Machine-to-Machine Communication. http://www.res-com-project.org.
3. R. Brachman and H. Levesque. *Knowledge Representation and Reasoning (The Morgan Kaufmann Series in Artificial Intelligence)*. Morgan Kaufmann, 2004.
4. L. Christiansen, and A. Fay, B. Opgenoorth, and J. Neidig. Improved Diagnosis by Combining Structural and Process Knowledge. In *IEEE Conference on Emerging Technologies and Factory Automation (ETFA)*, 2011.
5. K. Dey. *Providing Architectural Support for Building Context-Aware Applications* . PhD thesis, 2000.
6. K. Dey. Understanding and using context. *Personal Ubiquitous Computing*, 5(1):4–7, January 2001.
7. R. Drath. *Datenaustausch in der Anlagenplanung mit AutomationML - Integration von CAEX, PLCopen XML und COLLADA*. Springer, 2010.
8. K. Henricksen, J. Indulska, and A. Rakotonirainy. Modeling Context Information in Pervasive Computing Systems. In *Pervasive Computing*, volume 2414 of *Lecture Notes in Computer Science*, pages 79–117. Springer Berlin / Heidelberg, 2002.
9. Interview with Ingmar Hofmann.
10. L. Ollinger, J. Schlick, and S. Hodek. Leveraging the agility of manufacturing chains by combining Process-Oriented production planning and Service-Oriented manufacturing. In *Proceedings of the 18th IFAC World Congress*, 2011.
11. M. Schleipen, R. Drath, and O. Sauer. The system-independent data exchange format CAEX for supporting an automatic configuration of a production monitoring and control system. In *2008 IEEE International Symposium on Industrial Electronics*, pages 1786–1791, June 2008.
12. M. Schleipen, M. Okon. The CAEX Tool Suite - user assistance for the use of standardized plant engineering data exchange. In *IEEE Conference on Emerging Technologies and Factory Automation (ETFA)*, 2010.
13. C. Seitz, C. Legat, Z. Liu. Flexible Manufacturing Control with Autonomous Product Memories. In *IEEE Conference on Emerging Technologies and Factory Automation (ETFA)*, 2011.
14. C. Stammen. *Condition-Monitoring für intelligente hydraulische Linearantriebe*. PhD thesis, RWTH Aachen, 2005.
15. P. Stephan, G. Meixner, H. Kößling, F. Flörchinger, and L. Ollinger. Product-mediated communication through digital object memories in heterogeneous value chains. In *Proceedings of the 8th IEEE International Conference on Pervasive Computing and Communications. (PerCom-2010)*.
16. A.Theorin, L. Ollinger, and C. Johnsson. Service-oriented process control with grafchart and the devices profile for web services. In *Proceedings of the IFAC Symposium on Information Control Problems in Maufacturing*, INCOM 2012
17. C. Vilakazi, T. Marwala, P. Mautla, and E. Moloto. On-Line Condition Monitoring using Computational Intelligence. page 23, 2007.