Applying Agents for Engineering of Industrial Automation Systems

Thomas Wagner

Institute of Industrial Automation and Software Engineering
University of Stuttgart
Pfaffenwaldring 47, D-70550 Stuttgart – Germany
Fax: +49 711 685 7302, Phone: +49 711 685 7295
wagner@ias.uni-stuttgart.de

Abstract. Designing, operating and maintaining industrial plants require extensive and complex engineering processes. An integrated engineering process considering all different aspects, data and workflow of plant automation design as well as interoperability to other systems is the key to more efficiency and lower costs of engineering tasks in the plant life cycle. There exists no comprehensive and satisfying solution to this problem today. However, an agent-oriented view can lead to fundamentally new and promising approaches to an integrated plant engineering process. The goal of this paper is to clearly identify the specific goals and challenges in engineering industrial plants and to show that agents are a beneficial approach to meet them. To this end, an agent-oriented solution for integrated engineering of automation systems is presented, applying the advantages of agent concepts while considering the constraints of existing automation structures.

Keywords. Automation systems, plant design & engineering, agent applications

1 Introduction

Engineering of industrial automation systems is an extensive and complex process that differs strongly from systems engineering in other application domains. This results from the fact that automation systems in industrial plants (like pharmaceutical, production or power plants) are complex, large and persistent hardware / software systems being stamped by the characteristics of the technical processes they are designed to control. Hardware and software within these systems are strongly interrelated and in designing a plant the mechanical, electrical, process and control software aspects have to be considered. The software in classical automation systems accomplishes the typical automation functions for control of the technical process [1]: control of sequences, signaling, handling, operating, monitoring and supervision. To implement this functionality, special languages with low abstraction level are applied (defined in IEC 61131-3). Examples are function building blocks and sequential function charts.

However today the software in automation systems faces new challenges. Firstly, automation systems are subject to ongoing partial modifications: rapid development and decreasing costs of high-capacity hardware components cause changes in the

M. Schillo et al. (Eds.): MATES 2003, LNAI 2831, pp. 62–73, 2003.
© Springer-Verlag Berlin Heidelberg 2003

plant's hardware structure and functionality during its life cycle, which is up to 30 years. As existing plants cannot be rebuilt due to high investment volumes, the changes in hardware components affect particularly the software: it has to be modified continuously and its amount and complexity increases. Besides that plant operating managers increasingly require the possibility to use hardware and software components of different manufacturers and technologies in their plants [2]. Both trends result in increasing efforts for integration and configuration of the automation system. Hardware components of different manufacturers e.g. often require individual parameterization tools, resulting in a large set of tools for the engineer to handle [3].

Besides that, plant operating managers today want to manage and optimize the use and the maintenance of their assets during the operating of the plant. And finally, a strong need for the integration of functionality from enterprise management and business levels into the automation system arises (vertical integration) [2]. As automation systems have been evolved isolated from other enterprise software systems, every system has its own grown architecture and data structures and the efforts for establishing interoperability of the heterogeneous systems are high. These aspects are discussed in more detail in [4;5;6;7]. They all affect tasks and processes for the engineering of automation systems, leading to three main challenges:

1. **Faster development and modification of automation systems in plants**, e.g. engineer support in plant design (planning), implementation and commissioning.
2. **Support in plant management and engineering during operation**, e.g. condition monitoring, diagnosis, maintenance and asset management.
3. **Integration and interoperability of different heterogeneous systems,** e.g. enterprise resource planning systems, or parameterization tools for field devices.

Today a lot of these engineering and integration tasks are done manually with high effort and costs. There is a strong need in the industry for extensive support of these tasks by additional engineering functionality of the automation system. However, the listed challenges result in strong requirements on flexibility and adaptability of the software during the whole life cycle [8]. And, for the automation of engineering and integration tasks also more autonomy in the software is needed. As explained in [6], existing automation system software concepts with static architectures and functionality are not capable to meet these requirements.

The agent-oriented paradigm is a promising new approach for systems that are complex, distributed and changeable. It is a reasonable alternative to contemporary approaches in software engineering [9]. But not much work has been done yet to investigate how agent-oriented approaches could help on dealing with the imbalance of existing automation system structures compared to the new engineering challenges. This paper analyses the problem domain and presents a agent-oriented concept that brings a substantial new approach for the engineering and integration challenges within existing automation software.

This paper is structured as follows. In Section 2 the classical way of engineering automation systems in plants together with its problems is introduced. In Section 3 modern component-based approaches are investigated. The agent-based engineering approach is presented in Section 4. It shows the use and the advantages of agent concepts for engineering automation systems.

2 Engineering of Industrial Automation Systems

In this paper the term *engineering* is used with regards to automation systems in industrial plants. According to [10], engineering is defined as

"...all tasks and activities, that are carried out for the purpose of planning, installation, commissioning, operation and maintenance of technical plants."

This definition emphasizes that engineering not only concerns the functional aspects of automation software, but spans over the whole life cycle of a plant. Fig. 1 depicts example engineering activities during the plant life cycle. Several engineering areas participate in the engineering process of the automation planning project, like process planning, plant structure planning, electrical planning, or control software design. They regard different aspects of the automation system. In each engineering area particular aspects of the entities (devices) are parameterized and the entity is integrated in the automation system in different ways. For instance in the process planning mechanical assembly of devices and material flows are determined while in the electrical planning the electrical connections of devices are established.

The most important characteristic of the classical engineering process is the strong orientation to the different participating engineering areas. Their separated activities - each representing a specific view on the plant - are done by engineering experts who use their expert's tools (CAD/CAE-systems), create a view-specific partial model of the plant and generate specific data output. The sum of all view-specific data represents the plant documentation. Although the tools and data are separated it is obvious that the various engineering activities are not independent of each other, but have a lot of interrelations as all views regard the same real plant.

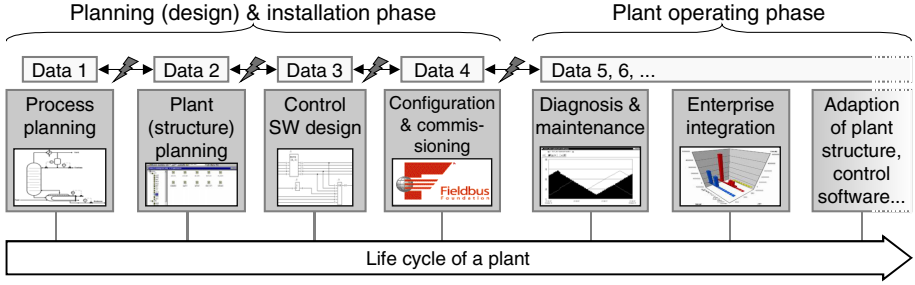


Fig. 1. Engineering activities during the life cycle of a plant (see also [11])

The isolated data storage and individual proprietary semantics of the view-specific tools cause a missing transferability (overall usability) of the created engineering data between the different tools. During the plant planning phase often changes have to be made in the automation planning project. Configurations of a particular entity, of a large number of entities or even a whole model may change frequently. As the view-specific activities have a lot of interrelations and dependencies, manual multiple data entry is necessary. For example, if changes in a process planning model occur, the control software has also to be adapted. The manual multiple entry and adaptation of

data causes the strong problem of inconsistent data (usually manual naming conventions are used to reduce this problem). Also an extensive support of engineering processes (like problem solving processes and workflow between engineering areas) is not possible. Another effect of the separated planning tools and data is the missing availability of planning data for engineering activities during the operation of the plant like diagnosis, maintenance, asset management, integration of enterprise systems (like ERP or MIS), asset optimization amongst others.

Today the separation of views and tools of the different engineering areas do not match the requirements for effective engineering. Instead an integrated consideration of all engineering activities as well as the activities during plant operating phase is necessary (see also [11]). Modern engineering approaches try to address the problems of data and process integration as explained in the following Section.

3 Modern Engineering Approaches with Technological Components

The goal of modern engineering approaches is to increase the productivity of the engineering activities by supporting of the engineering process in three points:

- Automation of task sequences during the engineering process
- Integration and transferability of engineering data
- Reuse of engineering solutions (automation planning projects or parts of them)

New approaches for the development of automation systems have to consider the fact that the automation system itself hardly can be changed [6]. Alternative solutions have to support existing system architectures and legacy systems to a large extend. To solve this problem, automation software companies developed modern approaches for the engineering of automation systems based on component-based ideas.

Starting point of modern engineering approaches are the different engineering views on a plant. For the integration of the engineering data, similarities between the different views can be identified: the link between all individual views on the plant is the real existing device within the plant with its various aspects for the automation of the plant. All participating engineers view their respective aspects of the device. Some aspects concern the device itself like mechanical / electrical characteristics, electrical / process connections and control software. Other aspects concern the actual position and tasks of the device within the automation system like device configuration, wiring diagrams and operating graphs.

The basic idea of the approach is to combine all different views in a single entity and thus to modularize the engineering data analogous to the real entities in the plant (physical decomposition, see [2]). The entities are called "*technological components*" [7] (other notions are e.g. "*planning objects*"). They represent a kind of data container, where the several engineering aspects are integrated and associated. Based on this component approach for the integration of engineering data the different views, their interrelations and the engineering processes as well as workflow can be supported. Fig. 2 illustrates the idea of data integration into technological components.

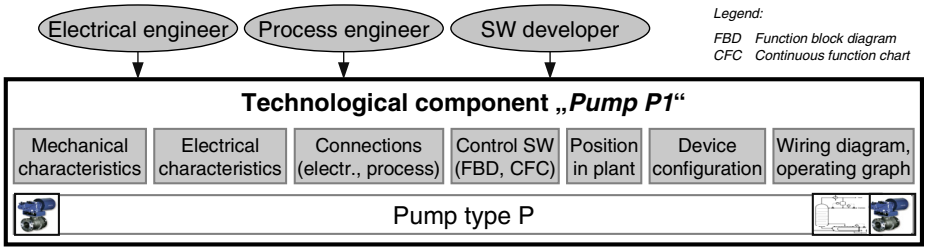


Fig. 2. Integration of engineering data with technological components

The technological components are the starting point for a higher-level engineering view on the plant's automation system. They represent a hierarchical structuring of the plant from an abstract technological perspective (e.g. mechanical assembly, location). Technological components represent mechanical or process elements (valve, pump), electronic elements (switch, contactor), automation hardware (PLC, I/O-card) or even automation software [7]. In the example in Fig. 3 on the right a project is designed with a boiler, consisting of a valve and a pump. Attached to the technological components are their particular views and the views they are part of. The example shows the boiler's operating graph and the function block diagram of the pump.

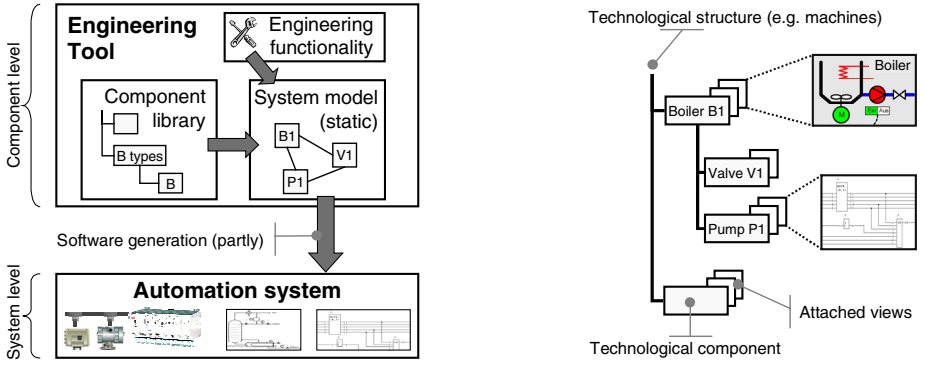


Fig. 3. Engineering with technological components: principle (left) and example (right)

Fig. 3 on the left shows the principle of a engineering with technological components. It represents a higher abstraction level of engineering on a component-based level. The engineering process for a particular automation project now is done by the creation of a system model of the automation system by instantiation, composition and parameterization of predefined 'empty' technological components from a library. Different technological structures (e.g. the mechanical assembly view) can be created and combined to an overall structure, whereas in every view different connections are established. For example in a control diagram the same elements are connected in another way than in a wiring diagram or in a operating graph.

This concept generally advances the engineering process in three main points:

1. Integration of engineering data in a high-level technological view with associated aspects: basis for integrated workflow and the reuse of solutions – of predefined library components as well as of created model (parts).
2. Support of engineering processes by creation of system model by component selection, instantiation, composition and parameterization.
3. Technology independent reuse by abstraction of the created system model from the target automation system technology (e.g. Siemens S7), partly generation of automation system software possible (depends on tool).

Examples of tools are Innotec Comos PT, ABB Industrial IT, Intergraph SmartPlant, Eplan PPE. They offer extensive engineering functionality (see Fig. 3) to support the various facets of the engineering process. They provide methods and functions to meet the prevailing challenges and problems, the engineer has to cope when designing a plant's automation system in a component-based way. Examples are:

- **Management of components' dependencies**. There are a lot of different types of complex associations between components, e.g. type-instance associations (library and planning project), associations and inheritance in technological hierarchy, or connections (like motor and wires). Associations depend on particular views. When changes are made in the model, conflicts have to be recognized and solved.
- **Model consistency**. E.g. ensuring data consistency between particular views.
- **Mass data changes**. E.g. when a parameter of a certain valve type has to be changed in a large number of valves. This requires support for search, selection of components and propagation of changes. Here also conflicts have to be recognized.
- **Reuse of engineering solutions**. Reuse addresses a wide range of problems, e.g. in what way reuse is possible (project internal or between projects), granularity of reusable solutions, intelligent management of components' dependencies, data reuse vs. know-how reuse (design decisions and patterns).
- **Support for different users and access rights**
- **Configuration management and deviation management**
- **Diagnosis of engineering errors**

The next Section discusses the constraints of modern engineering tools regarding theses challenges. An agent-oriented view leads to a basically improved approach.

4 Applying Agents for Plant Engineering

The merits of modern component-based engineering approaches result from the idea to create an abstract model of the automation system. The engineering process itself is a transformation process of the engineer's understanding about the automation system into a model of the system. In the depicted component-based approach this model is a *static* image of the automation system. The components are solely data containers for the engineering data (what from parts of the automation software can be generated). The transformation process is characterized by the specific challenges discussed in Section 3. Component-based engineering tools try to meet these challenges by supporting the transformation process by providing a set of engineering functionality. As the problem domain of automation systems and therefore the engineering process it-

self is quite multifaceted and complex, the functionality the tools have to provide is also extensive and complex (e.g. an "intelligent" copy-paste functionality for managing the dependencies when reusing model parts). It is arguable, if along with new challenges on the engineering process this functionality can be maintained.

A basically different approach in supporting the engineering process is to proceed decisions and handling of engineering functionality in the place where the most information on the current situation (position, tasks, associations, dependencies) of a particular entity in the model is available: within the entity itself. Therefore a concept is needed that provides more than static data containers.

The agent-oriented paradigm is a software paradigm that especially emphasizes the localization of knowledge and decision. In our understanding of agent-oriented software engineering, the concept of an agent is an "*encapsulated (software) entity with a defined objective. An agent tries to reach it's objective with autonomous actions and in doing so it can interact with it's environment and other agents, while keeping a persistent state.*[1]" Based on the agent concept a solution for the support of the engineering process can be developed, where the entities themselves perform the engineering functionality actively. Seeking this solution, the first step is analyzing the engineering of automation systems from an agent-oriented view.

4.1 Agent-Oriented Analysis of the Engineering Process

Agent-oriented analysis is the abstraction of a problem by means of agent concepts. In Section 2 the problem of engineering automation systems in plants was described in general. The main engineering activities, processes and goals can be regarded independently from a particular automation planning project. We can analyze these activities from an agent-oriented point of view. To this end, the concept of role analysis can be applied. In role analysis, the notion of a role focuses on the position and responsibilities of an entity within an overall structure or system [13]. Roles result from tasks a particular entity has to fulfill within the system, in order to provide the system's overall functionality. Applying this concept to the problem of engineering automation systems, both the challenges in the engineering process as well as the characteristics of the industrial automation systems domain that influence these challenges[2] have to be regarded. In the description in Section 2 we can identify that nearly all engineering tasks have an association to some kind of real existing entity within the plant. Each entity is integrated in the plant in different ways, where different engineering activities have to be accomplished. Focussing on the entities within the plant we can identify roles that e.g. a particular device obtains during the engineering process. Table 1 shows an overview of these roles.

The result of the agent-oriented analysis based on roles is an agent-oriented meta model of the engineering process. The overview in Table 1 shows the most important characteristic: the predominantly active elements in the engineering process are the

[1] This definition adapts those of Jennings & Wooldridge (see [9;12]). The author strongly agrees with their definition, but additionally wants to stress the fact that the notion of an agent has to be seen not only in the sense of an implemented computer system, but as a comprehensive concept for thinking about and designing software.

[2] The meaning of roles in the context of automation systems is discussed in [6].

Table 1. Roles of entities in the engineering process

Role	Explanation	Activities (examples)
Representation of particular view (multiple occurrence)	Reflects the different participating engineering areas. All activities are view-specific, but depend on settings in other views.	• Acquire view-specific data • Accomplish engineering sequences • Manage view-specific associations to other entities • Ensure consistency to other views
Instantiation, configuration	The entity's functionality has to be customized when it is instantiated for a planning project.	• Support selection of desired entity and functionality • Support setting of parameters • Adapt or reconcile parameters
Integration (composition)	On integrating an entity in the system, it is assembled in a particular place[3] in a particular environment.	• Establish associations to other existing entities (view-specific) • Consider existing rules / constraints
Access rights management	Manage data access of different user types.	• Provide / deny read and write access to engineering data
Configuration management	Entities or models have different configurations.	• Handle several configurations • Recognize configuration conflicts
Commission, monitoring, maintenance	Part of engineering processes during plant operating phase (see sect. 5).	• Manage deviation between model and automation system • Condition monitoring of an entity
Superordinate responsibilities	Roles that are spanning over several or all entities.	• Manage devices search data • Inspect model for engineering errors

planned entities in the plant. In every engineering activity one or more entities are concerned and obtain one or more roles. Obtaining a role the entities handle data and accomplish activities. These planned entities are the same abstract elements as the technological components used as data-containers in the component-based engineering approach. On one hand, this shows that both approaches are reasonable ones for the plant engineering process. On the other hand, the component-based approach falls short in separating engineering data (within technological components) and engineering functionality (provided by the tool). However, engineering with technological components is familiar to the engineer's way of thinking [7] and in most cases it is transparent to the user how a certain functionality in a tool is achieved. Therefore a combination of both approaches could bring the most value.

[3] The term "place" depends on the current view, it means either physically in mechanical assembly views or logically in other views.

4.2 Agent-Based Plant Engineering Approach

In the agent-oriented meta model of the engineering process based on role analysis the entities in the automation system are aware of their position and tasks. The conceptual model of technological components encapsulates all engineering data belonging to an entity. The combination of both enables an entity to handle actively all necessary engineering functionality. Fig. 4 on the left metaphorically shows the consolidation of this meta model with the conceptual model of technological components: the agent representing an entity within the plant is integrated in the technological component. The agent-oriented meta model can be implemented as an agent-based component library, providing both predefined engineering data and functionality. The agent represented as a technological component can - based on its roles - actively support the engineering process using the knowledge of its current position, data and dependencies in the planning project. Therefore the term of agent-based engineering in the context of this paper means **engineering based on agents**.

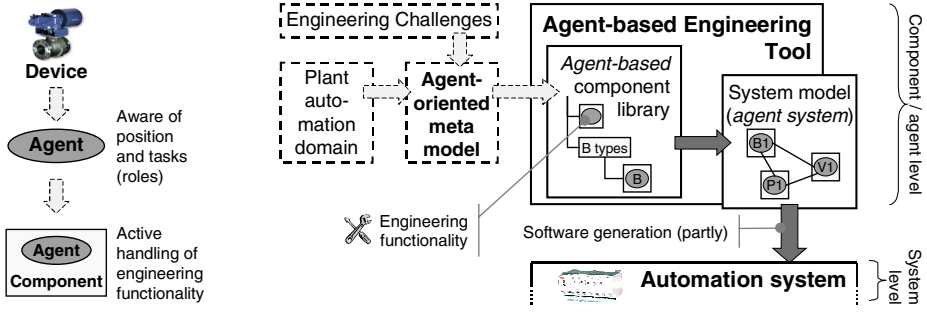


Fig. 4. Agent-based plant engineering: metaphor (left) and principle (right)

In designing the automation system with the agent-based component library the same engineering process is performed with selection, instantiation, integration and parameterization of the familiar components. However, in doing so now an agent system is build (Fig. 4 on the right). The whole engineering process is based on an agent-system which is transparent to the engineer. Metaphorically speaking the created agent system is a "living" system model, that takes care of itself.

Fig. 5 shows the roles and associations within an agent-based engineering environment. When a component is selected from the component library and instantiated in the system model (the planning project), a type-instance association is established between the two involved agents. Thereby a flexible handling of the various dependencies, e.g. the reconciliation of changes between library component and instances (delta generation) is realized. In the case of a previous individually changed instance later inheriting changes from the library component, inevitable conflicts can be recognized and handled immediately. When the component is integrated in the planning project, connections are established between agents, that can also be handled actively in the course of the engineering process. Integrating the component in different views then addresses different roles and their respective representations, associations and restrictions. Engineering process support now can also be done by the management of

requirements concerning particular "places" in the model. In Fig. 5 a requirement restricts the operating temperature for the Motor in the place of M1. It is represented by an agent. When the library component is integrated in this place, the two agents have to co-ordinate this requirement and in case the instantiated motor does not meet it, the user can be automatically informed.

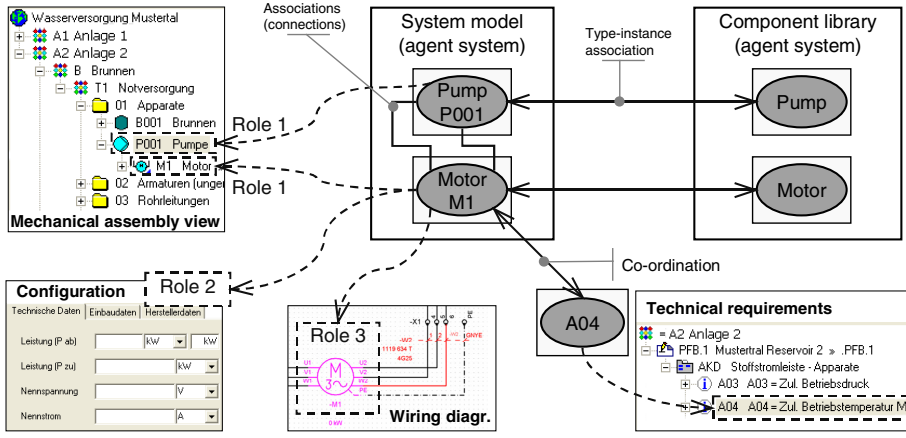


Fig. 5. Agent based plant engineering: roles and associations

The merits of the agent-based engineering approach are active support in instantiating and integrating components, management of components' dependencies, active propagation of changes and recognition of possible conflicts as well as engineering errors. Active management of dependencies also fundamentally improves the reuse of models or model parts. It represents a step towards the reuse of engineering know-how (like design decisions and patterns) instead of reuse of data. In the next Section the further possibilities of the approach are estimated.

4.3 Plant Integration with the Agent-Based System Model

In the agent-based engineering process a system model is created as an instantiated agent-system. It actively supports the engineering process and thus meets the first of the tree challenges mentioned in Section 1: *Faster development and modification of automation systems.* But moreover the model has further possibilities. It also absorbs the engineering knowledge about the plant like positions, tasks, associations and dependencies. And it is a standalone application, a kind of parallel high-level process control system. This high-level system with "active components" that are aware of their position and tasks contains the essential engineering knowledge to be the basis for meeting the remaining two engineering challenges discussed in sect. 1: *support in plant management and engineering during operation* and *integration of different heterogeneous systems in the environment of the automation system.* The active components (alias agents) can be used to acquire necessary information from the automation system, using the technical possibilities of existing interfaces and to coordinate all of

the information flows to connected systems. And in some cases it is even possible to intervene in the running automation system.

For example an agent representing a pump *P1* in the model could be assigned the goal to observe the physical pump. It can address the pump via the automation system's interfaces [3] and collect the diagnosis status signals. Moreover, by the maintenance system the agent could be given the goal to analyze the status signals and if required generate a maintenance order in the associated enterprise resource planning system. It even could guide or support the mechanic in repairing the pump and if necessary automatically re-configure the pump. Fig. 6 shows the correlation between system model, engineering tool, automation system and the environmental systems.

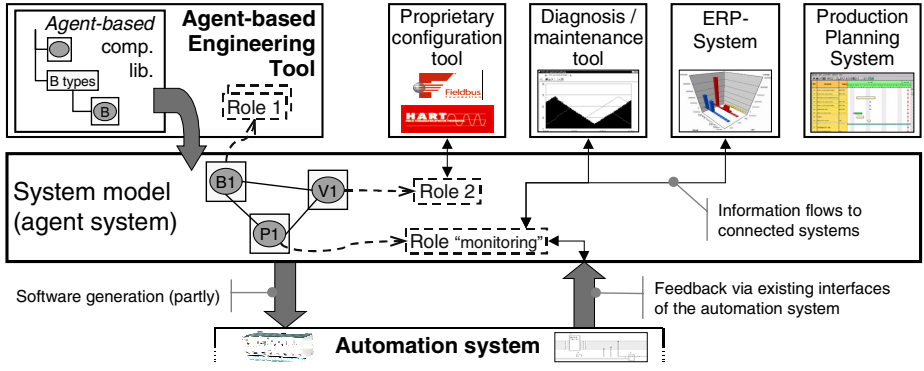


Fig. 6. Plant integration with an agent-based system model

All necessary infrastructure for the system model to fulfill its engineering tasks during plant operation is provided by the *Agent Plant Integration Framework*. It provides infrastructure like access to the automation systems interfaces. External systems can be encapsulated by agent adapters and data semantics of inhomogeneous systems can be mapped within the framework. In designing the system model of the plant the framework is tailored for the real automation system. Its infrastructure combined with the agent-based system model provides the basis for all engineering tasks during operation like monitoring, supervision, diagnosis, maintenance, simulation, quality analysis as well as integration of legacy and proprietary configuration tools, enterprise & business administration systems and electronic plant documentation.

5 Conclusion

The challenges on plant automation systems require improved support of engineering and integration tasks by the automation system itself. The classical fragmentation of the engineering process into separated views and view-specific tools leads to missing transferability of data, high engineering and integration effort in both plant design and plant operation phase, and to less reusable solutions. New component-based approaches integrate the specific views and data in a higher-level technological image of

the system. However, they fall short due to the facts of separating engineering data from functionality and the created system model remaining a static data model.

In this paper the idea of combining the component-based higher-level system model with an agent-oriented meta model of the engineering process was presented. It leads to the approach of agent-based engineering, where the engineering functionality is accomplished by the components themselves. Thereby a better and more flexible support of processes is possible. And, the essential engineering knowledge now is absorbed by the created system model as it is an agent-system. This knowledge can be used as the basis for all engineering tasks during the life cycle of a plant. The Agent Plant Integration Framework therefore provides all necessary infrastructure.

The presented approach employs the merits of agent concepts like autonomous acting, flexible interactions and goal-orientation to extensively support the engineering of automation systems in plants and the integration of heterogeneous systems in the environment of the automation system. Refinement and realization of this approach is subject to work in progress. Working points are the agent-oriented engineering model, design of the framework, "agentification" of legacy component-based engineering models and establishing technical communication to the automation system, like data access for automatic adaptation of the model to changes in the plant.

References

1. Lauber, R.; Göhner, P.: Prozessautomatisierung I, 3. Edition, Springer-Verlag Berlin, 1999
2. Shen, W. and Norrie, D.H.: Agent-Based Systems for Intelligent Manufacturing: A State-of-the-Art Survey. Knowledge and Information Systems, 1(2), 1999, p. 129–156
3. Barata, J.; Camarinha-Matos, L.; et.al.: Integrated and Distributed Manufacturing, a Multi-agent Perspective, 3. Workshop on European Scientific and Industrial Collaboration, 2001
4. Leitão, P.; Barata, J.; Camarinha-Matos, L.; and Boissier, R.: Trends in Agile and Co-operative Manufacturing. Proceedings of Low Cost Automation Symposium, 2001
5. Parunak, H.V.D.: Practical and industrial applications of agent-based systems. Environmental Research Institute of Michigan (ERIM), 1998
6. Wagner, T.: An agent-oriented approach to industrial automation systems. In: Kowalczyk et. al. (Eds.), Agent Technologies, Infrastructures, Tools and Applications for E-services 2002, Lecture Notes in Artificial Intelligence Nr. 2592, Springer-Verlag, 2003, p. 314–328
7. Löwen, U.: Informationstechnologie als Enabler für effizientes Anlagenengineering. Lecture: Methods of Software Engineering, Institute of Automation and Software engineering, 2002
8. Albrecht, H.; Meyer, D.: Ein Metamodell für den operativen Betrieb automatisierungs- und prozessleittechnischer Komponenten. at – Automatisierungstechnik, Heft 50, 3/2002
9. Jennings, N.R.: On agent-based software engineering. Artificial Intelligence 117, 2000
10. Simon., R.: Gerätemodell zur Feldinstrumentierung von verteilten Automatisierungssystemen. at – Automatisierungstechnik, Heft 10, 2002
11. Rauprich, G.; et. al.: PLT-CAE - Integration in gewerkeübergreifendes Engineering und Plant-Maintainance, atp - Automatisierungstechnische Praxis 44 (Heft 2), 2002
12. Jennings, N.R.; Sycara, K.; Wooldridge, M.: A Roadmap of Agent Research and Development. Autonomous Agents and Multi-Agent Systems, 1, 1998, p. 275–306
13. Kendall, E.: Role Modeling for Agent System Analysis, Design, and Implementation. First Internat. Symposium on Agent Systems and Applications (ASA'99), Palm Springs, 1999