III INTERNATIONAL CONFERENCE
TRANSPORT SYSTEMS TELEMATICS TST'03

ZESZYTY NAUKOWE
POLITECHNIKI ŚLĄSKIEJ 2003
TRANSPORT z.51, nr kol. 1608

Distributed Systems,
diagnosis, automatic generation

Markus BREGULLA[1]
Daniel GROSSMANN[2]

UNIFIED DIAGNOSIS OF DISTRIBUTED SYSTEMS BY MEANS OF GENERIC FUNCTIONALITY

As today's Automation solutions become more and more distributed Systems the expense for implementing system diagnosis as part of Engineering rises. This paper presents a concept to decrease the cost for implementing system diagnosis by generating diagnostic funtions automatically out of the layout information created in the planning phase of a plant. It also describes a concept for reducing the complexity of the system by forming logical groups, each embracing a part of the whole system. The Concept has been implemented and validated based on the PROFInet architecture for component based automation in cooperation with SIEMENS AG Automation & Drives.

ZUNIFIKOWANA DIAGNOSTYKA SYSTEMÓW ROZPROSZONYCH PRZY POMOCY STANDARDOWEJ FUNKCJONALNOŚCI

Z uwagi na to, że obecnie w automatyce coraz częściej stosowane są systemy rozproszone, koszt realizacji funkcji diagnostycznych dla kompletnych systemów niezmiernie wzrasta. Artykuł prezentuje koncepcję obniżenia kosztów i czasu implementacji diagnostyki systemów rozproszonych poprzez generowanie funkcji diagnostycznych automatycznie na podstawie planów rozmieszczenia komponentów i ich struktury technologicznej, tworzonych w fazie projektowania. Opisuje się tu również koncepcję redukcji złożoności systemu poprzez tworzenie podzespołów logicznych. Koncept został zrealizowany i przetestowany przy pomocy bazującej na komponentach automatyki architektury PROFInet we współpracy z SIEMENS AG Automation & Drives.

1. INTRODUCTION

In the lifecycle of today's automation solutions, engineering represents a significant expense. While on the communication level, established standards simplify the cooperation of field devices the functionality for plant diagnosis has to be costly reimplemented for each automation solution "from scratch". The growing functionality and the different operating behaviour of field devices, especially in multi manufacturer environments, complicate this

[1] Automation and Computer Engineering, University of Applied Sciences, Esplanade 10, 85049 Ingolstadt, Germany, markus.bregulla@fh-ingolstadt.de
[2] Automation and Computer Engineering, University of Applied Sciences, Esplanade 10, 85049 Ingolstadt, Germany, daniel.grossmann@fh-ingolstadt.de

effort. These aspects increase the complexity of the system and therefore also the expense for implementing system diagnosis as part of engineering.

During the planning phase of an automation solution (e.g. Plant) computer aided planning tools facilitate the process of designing the layout of the automation solution. These systems contain various informations on the system like used components etc. Based on the information contained in the plant layout it is possible to automatically generate functions for the diagnosis of the whole system.

2. TECHNOLOGICAL HIERARCHY

One step to solving this problem is to build the automation solution on a component based structure. In this conception a component is an entity consisting of both hard- and software, providing a given automation function. Such a component can be see a black box which hides its complexity and owns a baselined interface to the "outside world". This interface also provides defined functions for diagnostic purposes. In the context of the PROFInet architecture the so called "Runtime Automation Object" (RTAuto) represents such an entity implementing a given automation function which can be accessed through baselined interfaces.

Based on this architecture, subsystems can be introduced, reducing the complexity of the system. Subsystems can be build on any level within the system. In this regard, a subsystem is a logical group which embraces a given part of the system, hiding all the underlying compelxity. The introduction of logical subsystems also simplyfies matters for the operating crew since the almost unmanageable complexity of the plant is encapsulated on different levels.

Figure 1 shows the layout of an automation solution. It also shows the seperation of the whole system into subsystems.

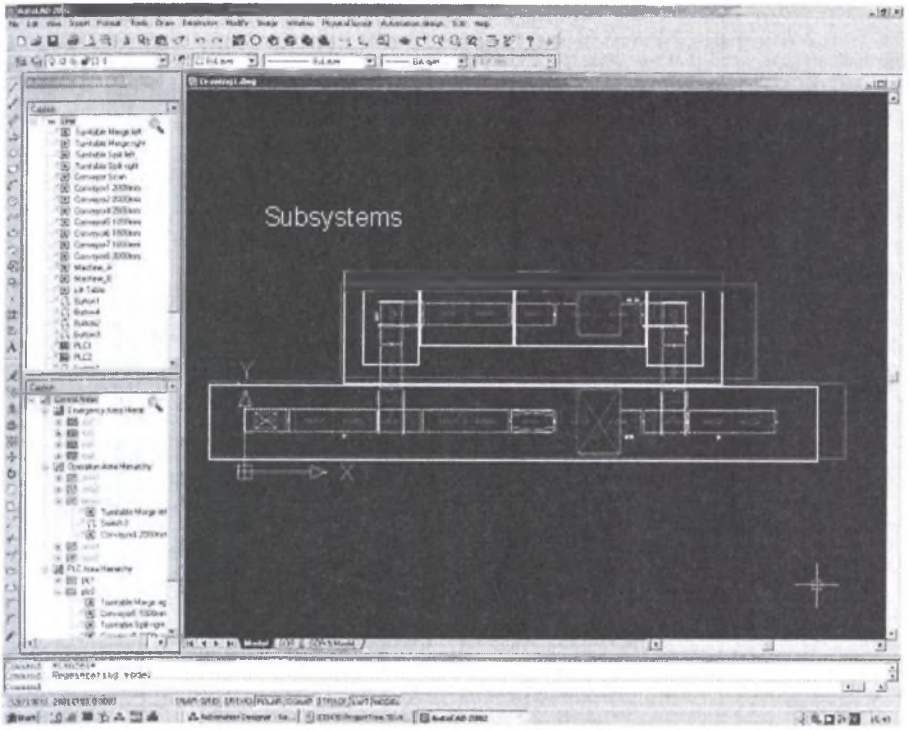

Fig.1. Plant Layout

In the context of PROFInet the automation function of a single component is provided by its RTAuto. This means that a subsystem contains either further subsystems or on the lowest level the RTAuto of the according component.

Figure 2 shows the technological hierarchy that can be derived automatically from the plant layout.

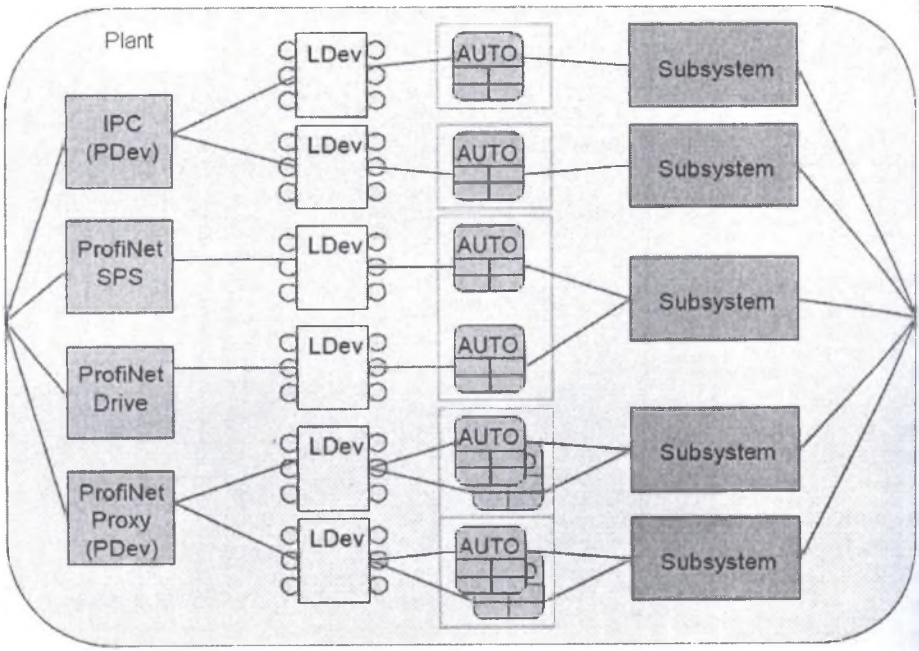

Fig.2. Technological Hierarchy

3. GENERATION OF DIAGNOSTIC FUNCTIONS

Based on the technological hierarchy, it is now possible to generate diagnostic functions for each element of the hierachy with the same sematics (e.g. as script functions). In the context of PROFInet the following functions can be generated:

- The diagnostic function of a Logical Device (LDev) checks the LDev Status. Depending on the result, the function reports an error with the according explanation.
- On the level of the RTAutos the generic function checks the status of the RTAuto and calls the diagnosis of the associated LDev. Depending on the result, the function reports an error with the according explanation.
- On the lowest level of the Subsystems within the technological hierarchy the generated diagnostic function calls the diagnosis of the underlying RTAutos.
- This priciple is applied recursivly up to the top level (the plant). This means that each subsystem diagnosis calls the diagnosis of the underlying subsystem.

Besides this delegation of diagnostic functions from a higher level to the underlying level, more significant diagnostic functionality can be generated by means of induction. For

this purpose the logic of the automation solution has to be taken into account. This system logic can be devided into two categories:

- „Single Level Logic" represent the logic of one element of the technological hierarchy. In this regard internal information of the element is associated. For example the logic of a subsystem „Rolling Gate" states that there has to be a failure (e.g. a Sensor) if the Gate reports that it is „open" and „closed" at the same time.
- „Multi Level Logic" represents the logic of a subsystem based on the underlying subsystems it contains. For example a subsystem consisting of a flow control and a valve has to be defective if the flow control states a „flow greater zero" while the valve states „valve closed".

During the recursive generation of the diagnostic functions (e.g. as scripts) the rules for single and multi level logic are taken into account:

- In the case of „Single Level Logic" the defined rules are directly applied to the generation process.
- In the case of „Multi Level Logic" the plant layout determines which rules apply in the given case, depending on which components interact with each other.

The basis for the creation of the single and multi level logic is the definition of device classes with fixed functions and properties. Rules for both Single and Multi Level Logic can then be defined, associating functions and properties. These rules can then be applied in the process of the automatic generation of diagnostic functionality. Within the prototype, these functions are generated as script functions. Figure 3 shows the basic principle of the automatic generation of diagnostic functionality.

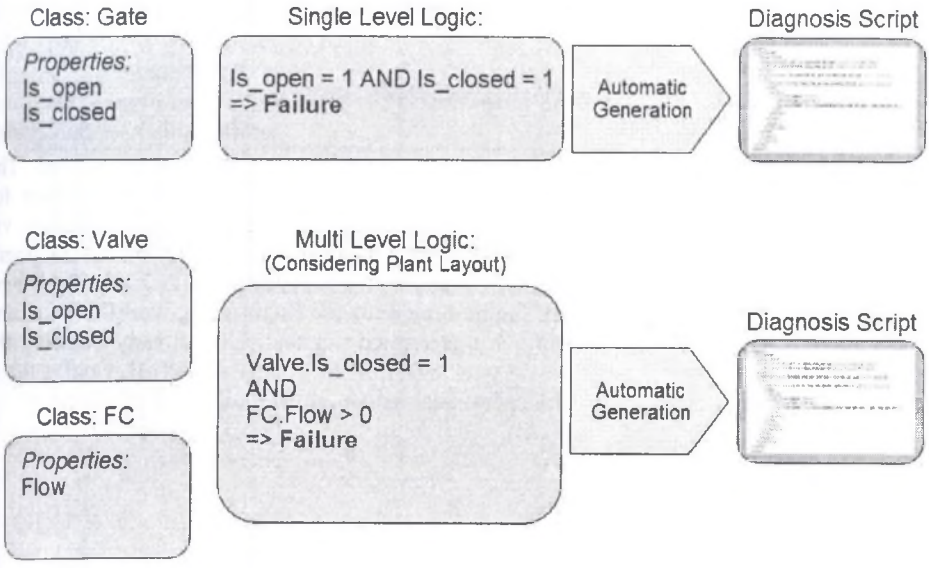

Fig.3. Automatic generation process

The process generates a diagnosis hierarchy with elements that can be seen as fractals since they are all similar and provide a diagnostic function with the same semantics. This principle contains even the subsystems although these logical elements do not exist as a real automation component. All informations that are necessary for this process can be retrieved from the plant layout.

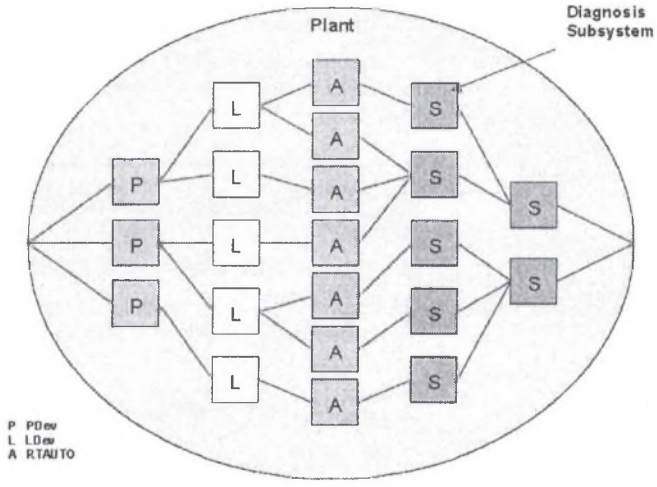

Fig.4. Generated diagnosis hierarchy

4. IMPLEMENTATION

In order to validate the described process a prototype has been implemented. The Implementaion consists of a so called „Diagnosis Server" and a Client Application for visualization purposes. The communication between Client and Server is handled via Webservices and the HTTP protocoll. Both Server and Client can be used for various plants without the need of adjustments since the Server is initialized via a XML Document containing the necessary informations. Figure 5 presents the Engineering workflow. Starting with the plant layout, the XML Config file is created automatically, already containg the generated diagnosis hierarchy and the diagnostic (script) functions. This XML Config file is passed to the Server which is then fully operational without further adjustments.

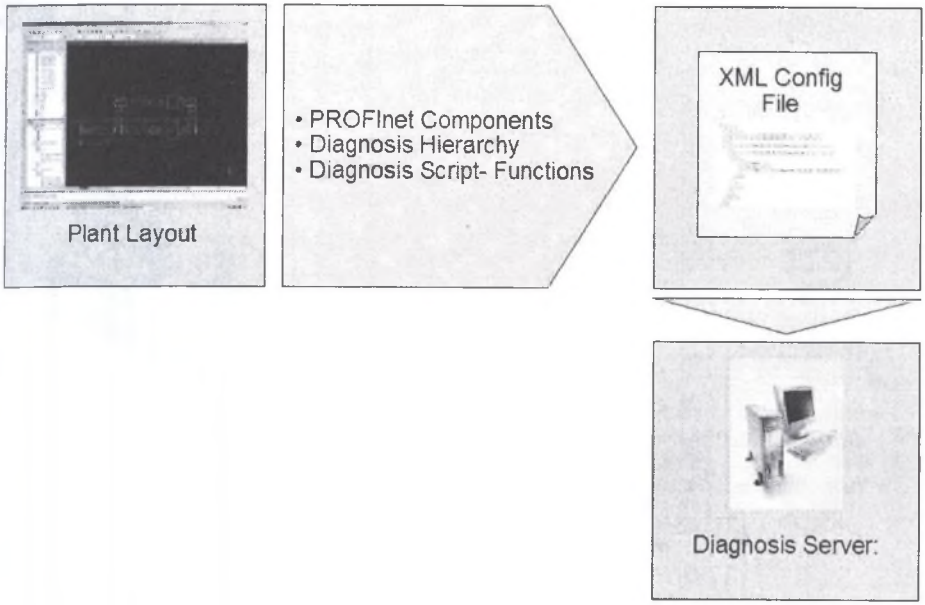

Fig.5. Engineering Workflow

The Client application connects to the Server and accesses the diagonstic functions. Since the generated functions exist as scripts within the XML Config File, the Server processes the script with the build in Script processor. The results are transmitted to the Client application for visualization.

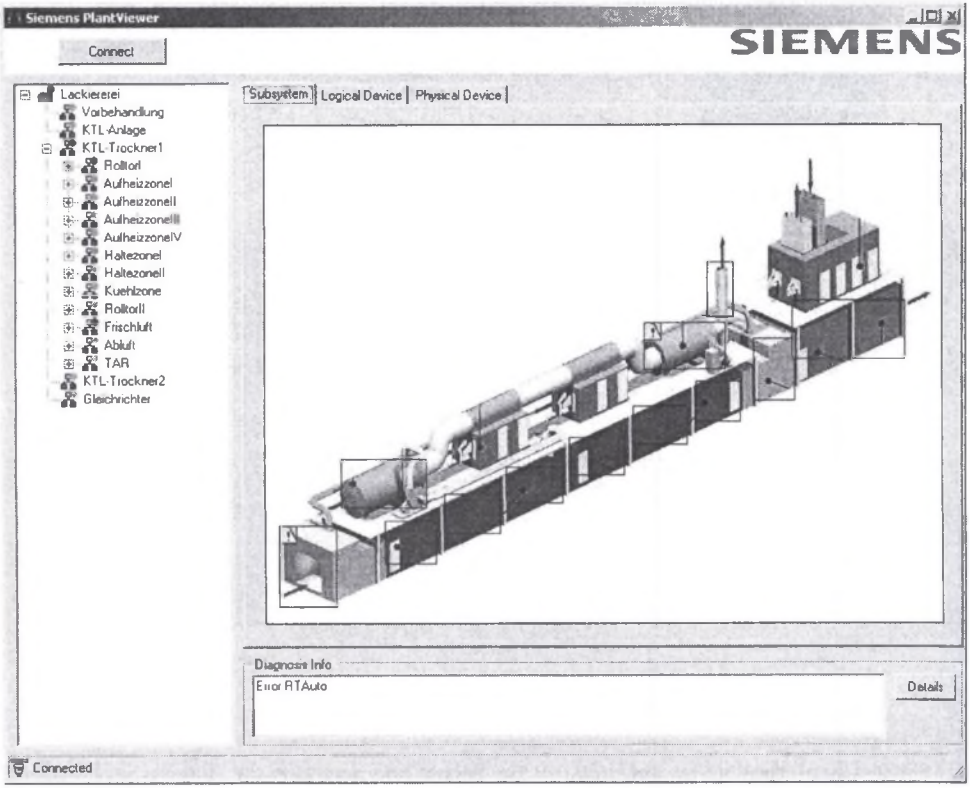

Fig.6. Client application

5. CONCLUSION

The project created a possibility to significantly reduce the effort and cost for the implementation of system diagnosis as part of engineering. Furthermore these diagnostic functions are less subject to implementation errors, since they are generated automatically. From a user's point of view, this concept reduces the complexity of the system by forming subsystems hiding the underlying complexity. This enables the operating crew to identify the plant's status "at a glance". In the case of a fault condition the diagnostic functions on a lower level provide more detailed information.

Reviewer: Ph. D. Stanisław Krawiec