4th INTERNATIONAL CONFERENCE
TRANSPORT SYSTEMS TELEMATICS TST'04

ZESZYTY NAUKOWE
POLITECHNIKI ŚLĄSKIEJ 2004
TRANSPORT z.55, nr kol. 1657

open interface,
mobile devices,
webservices

Markus BREGULLA[1]
Daniel GROSSMANN[2]

OPEN INTERFACE FOR DISTRIBUTED ACCESS TO DIAGNOSTIC INFORMATION

This paper describes a concept to encapsulate different communication protocols and operating philosophies used throughout an automation solution by providing an open high level interface as a Webservice. This XML based Webservice can be accessed by a wide variety of end devices since the extensible Markup language provides a platform independent format. The concept has been implemented as a central server unit and a mobile client solution.

STANDARDOWY INTERFACE DLA GLOBALNEGO DOSTĘPU DO INFORMACJI DIAGNOSTYCZNYCH

Niniejszy referat opisuje koncepcję zamknięcia różnych protokołów komunikacyjnych i eksploatacyjnych wykorzystywanych w rozwiązaniach automatyki poprzez dostarczenie wysokiego poziomu interfejsu jako usługi sieciowej. Taka usługa Webservice w oparciu o XML może być dostępna za pomocą bardzo różnych urządzeń końcowych, ponieważ rozległy język Markup dostarcza formatu niezależnego od platformy. Koncepcja została wprowadzona jako rozwiązanie dla centralnej jednostki serwerowej i ruchomego klienta.

1. INTRODUCTION

In today's automation world established standards simplify the cooperation of field devices on the communication level. Therefore multivendor plants become more and more popular. While on one hand this fact gives the user the freedom to choose the most appropriate device for his needs he gets confronted with often differing operating

[1] Automation and Computer Engineering, University of Applied Sciences, Esplanade 10, 85049 Ingolstadt, Germany, markus.bregulla@fh-ingolstadt.de
[2] Institute of Information Technologies in Mechanical Engineering, Technical University of Munich, Boltzmannstr. 15, 85748 Garching, Germany, grossmann@itm.tum.de

philosophies on the other hand. Especially for the diagnosis of complex automation solutions this fact can increase the effort for the diagnosis since each manufacturer has its own diagnosis philosophies. This fact of course also leads to an increased timely effort for finding the cause of a fault situation, which also generates increased costs.

2. HIGH LEVEL INTERFACE

The problem mentioned above can be solved by an open high level interface for the access to diagnostic information. This interface is implemented by a central server unit, which is responsible for gathering diagnostic informations from the field devices installed throughout the plant.

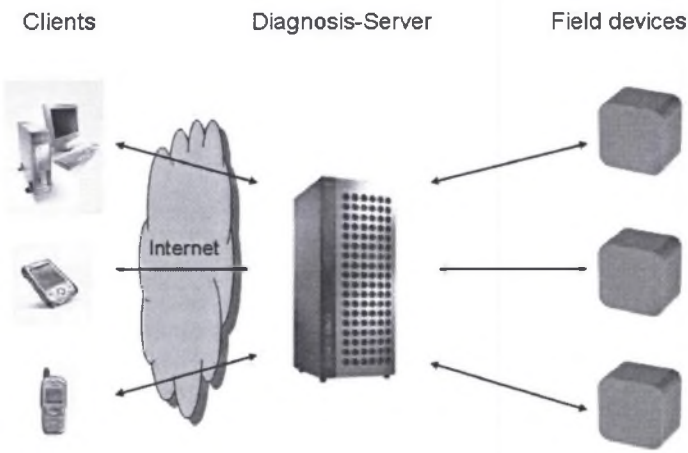

Fig.1. Diagnosis-Server concept

The interface is designed as a Webservice to achieve full platform independence. Webservices expose a given functionality for invocation via Internet. Since the Simple Object Access Protocol (SOAP) used by the Webservice is based on the Extensible Markup Language (XML) this Webservice can be accessed by a wide variety of end devices ranging from PC based solutions to mobile devices and cell phones. As a transport mechanism the Hypertext Transfer Protocol (HTTP) can be used which itself is based on TCP/IP. By establishing this layered architecture, layer 2 independence is achieved, giving the freedom to utilize a wide variety of layer 2 protocols as for example Ethernet (WLAN), GPRS, GSM etc.

7. Application Layer	SOAP (Webservice) HTTP
4. Transport Layer	TCP
3. Network Layer	IP
2. Data Link Layer	Ethernet, ISDN, GPRS, GSM
1. Physical Layer	Cable or wireless transmission

Fig.2. ISO/OSI Reference Model

3. INTERFACE DESIGN

The interface that is deployed as a Webservice draws advantage of the flexibility provided by the Extensible Markup Language. Therefore, the interface only consists of two Functions: GetPlantDescription and ProcessRequest.

- **GetPlantDescription**: The Function GetPlantDescription provides the description of the plant and its structure by means of the field devices it consists of. Furthermore also a description of the diagnostic functionality provided by the Diagnosis-Server is included in this XML-Document returned by the Function GetPlantDescription. The Client that invokes this Function can use this XML-based description to adapt itself to the given plant configuration.

- **ProcessRequest**: The Function ProcessRequest receives an XML-based request. This request contains the Diagnostic functions that are to be executed and the information about the target (field device) of the request. As shown in figure 3 one call to the Function ProcessRequest may contain several target nodes to reduce the communication load. For each target node several function nodes may be attached each providing the name of the function to call. The in and out nodes list the parameters passed to or retrieved from the Webservice.

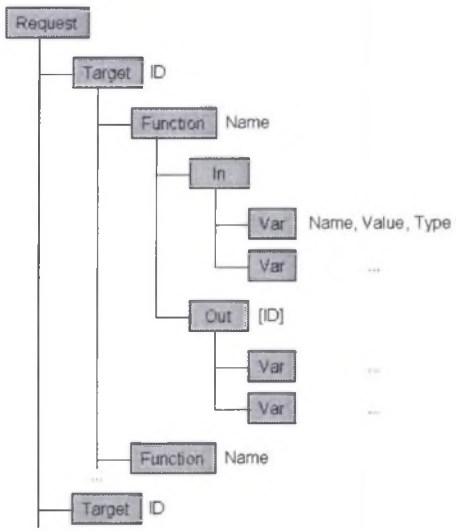

Fig.3. Structure of XML Function call

4. DIAGNOSIS-SERVER

The Diagnosis-Server that implements the Webservice has to gather the information that is necessary and relevant for diagnosis from the field devices installed throughout the plant. The field devices of the automation solution are represented as Device Objects within the Diagnosis-Server. The Device Objects form the Diagnosis Hierarchy that reflects the structure of the plant [1]. Each of this Device Objects exposes a unified interface for diagnostic purposes that is invoked upon a call to the Webservice Function ProcessRequest. In addition the Device Objects encapsulate different operation philosophies in regard to the diagnosis of a field device. For access to the field devices the Diagnosis-Server deploys several communication objects, one for each protocol used within the automation solution. Each Device Object is linked to its own specific Communication Object. Within the implemented prototype Communication Objects for the PROFINET CBA [2] Architecture for component based plant solutions. PROFINET CBA uses Ethernet and the distributed component object model DCOM [3]. This concept gives the freedom to have different protocols within the automation solution that are all transparent to the Client since the Client only needs to access the Webservice. Additionally with this concept the Diagnosis-Server can be used for differing plant configurations because the Diagnosis-Server is configured with a XML configuration file.

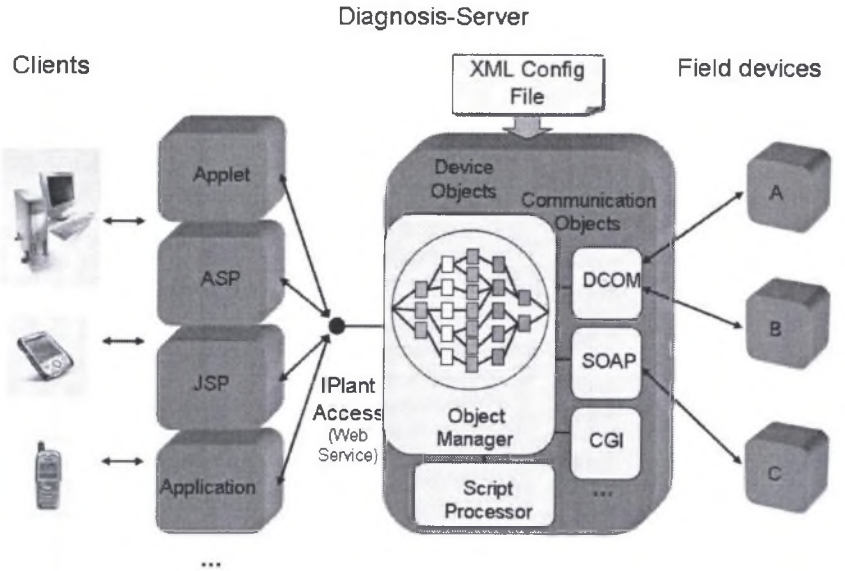

Fig.4. System Architecture

5. MOBILE CLIENT

As mentioned before the Webservice can be invoked from a wide variety of Clients ranging from Windows applications to Mobile Devices. A client solution implemented as a windows application was already presented in [1]. To show the flexibility of the concept an implementation for a Mobile Device (PocketPC) has been performed. The Client application consists of a generic graphical user interface (GUI) to adapt to different plant configurations. Once the Client connects to the Diagnosis-Server it retrieves the plant description and configures its GUI to the structure contained in the plant description retrieved by a call to the Webservice function GetPlantDescription. The Client may use communication technologies like for example WLAN or GPRS. The Implementation is based on then .Net mobile Framework for PocketPC using VB.Net as programming language.

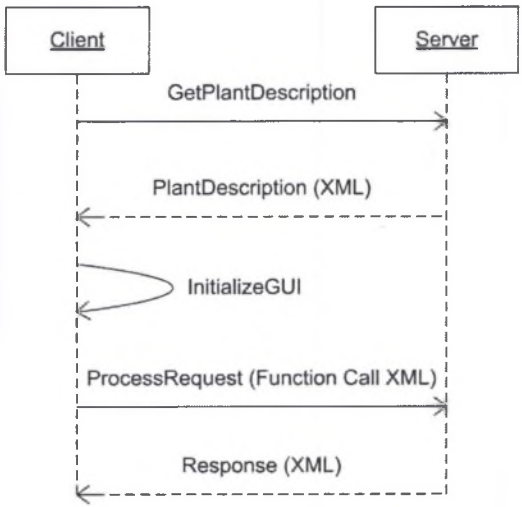

Fig.5. Sequence chart Client - Server

6. CONCLUSION

The described concept presents a possibility to encapsulate the different protocols used for access to Diagnostic information within a given automation solution by introducing an open high level interface designed as a Webservice. This Webservice can be accessed by using platform independent technologies like XML or HTTP. By introducing Device Objects that provide a unified diagnostic interface different operation philosophies are encapsulated by the Diagnosis-Server, making it easier for the user to deal with.

The outlined concept has been implemented as a Diagnosis-Server that gathers the informations from the field devices to the plant side and that exposes these informations to the Client side by implementing the described Webservice. The Client application is realized by the use of the .Net mobile Framework and a PocketPC. This Client can connect via WLAN but also communication technologies like GPRS are possible.

BIBLIOGRAPHY

[1] BREGULLA M., GROSSMANN D., Unified Diagnosis of distributed Systems by means of generic Functionality, Transport Systems Telematic III International Conference, Katowice 2003.
[2] PROFIBUS International, PROFINet Architecture Description and Specification V1.0, Karlsruhe 2001.
[3] Microsoft Corporation, The COM Specification Draft Version 0.9, 1995.

Reviewer: Ph. D. Stanisław Krawiec