Intranet based Management of Fieldbus Systems

Martin Wollschlaeger
Institute for Measurement Technology and Electronics (IPE)
Otto-von-Guericke-University Magdeburg, Germany

ABSTRACT

The increasing use of PC-based solutions in control systems allows the introduction of Intranet technologies to fieldbuses, especially for management purposes. The paper shows concepts, strategies and requirements for an Intranet based management of fieldbus systems. Exemplary solutions with P-NET and CAN are described. They provide data access as well as management functions for different fieldbuses using a browser as an unique, easy to handle user interface. Different strategies for a mapping of the bus systems' management functions to web-based solutions are described. Problems, benefits and further trends are discussed from the users' and vendors' points of view.

INTRODUCTION

One of the most outstanding features of modern process control solutions is the integration of data acquisition systems into hierarchically organized process information systems. These systems are implemented using different layers for data exchange (Fig. 1). A fieldbus is the state-of-the-art data transport solution in the lower layers, while common network implementations, derived from LANs, can be found in the upper layers. The single transport systems are connected together via gateways, implemented in PLCs, Controllers, or even PCs.
The different connected transport systems provide the chance of an easy integration of process control data into business applications. So data acquisition as well as system's management can be performed by means of standardized software packages, such as SCADA and HMI systems, and by de-facto standard office applications.

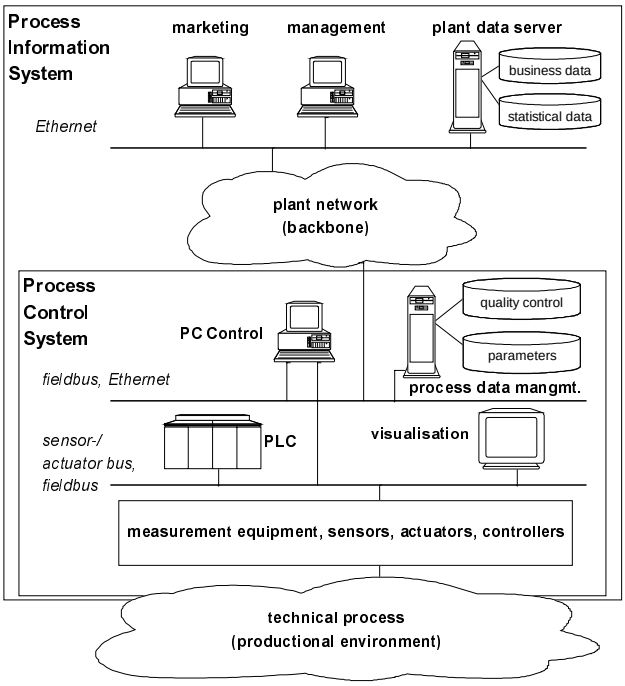

Figure 1. Process information system

Dedicated to the globalism in today's business, effective data transport solutions are required to exchange data around the world. So the Internet has become the most outstanding technique for an integration of the exchange of data with different contents and an easy to use interface. Intranets use Internet-related technology, such as HTTP-based data exchange,

on top of existing LAN installations (Fig. 2). More and more Intranets can be found in process information systems. They provide easy access to distributed plant and business data. Intranets are very integrative solutions, in terms of data sources and data types as well as concerning the networked components. These integrative aspects are applied to the data exchange itself as well as to the tasks of network management. Intranet based management solutions for LANs have been introduced during the last years. Such an unified, easy way of management is essential for fieldbus systems, too. The increasing use of PC-based solutions in control systems allows the introduction of Intranet technologies to fieldbuses.

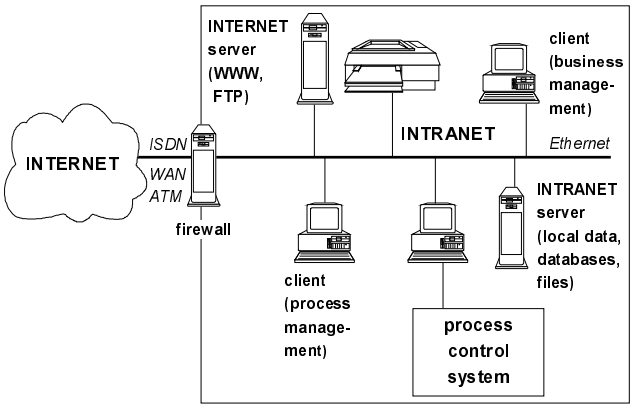

Figure 2. Relationship between Intranet and a process control system

MANAGEMENT OF FIELDBUSSES

MANAGEMENT TASKS

The management of fieldbusses includes tasks from different stages of a fieldbus system's life cycle. However, the most important management tasks can be found in configuration.
Fieldbus systems and their components have to be configured to meet the needs of the automation tasks they have to perform. This configuration includes functional and topological steps. The designed application programs have to be implemented into the components by means of function blocks. Depending of the bus systems used and of the components' hard- and software structure, the function blocks are implemented and configured in different ways. They can be pre-installed and configurable by parameters, like attributes of embedded objects in SDS (Fig. 3) or channels in P-NET, or they can be downloaded via the network (controllers, PLCs) using different services defined in the bus systems' protocols.
The communicational relations between the distributed function blocks are described by communication references, addresses, identifiers, softwire numbers and so on, depending on the used fieldbus system.

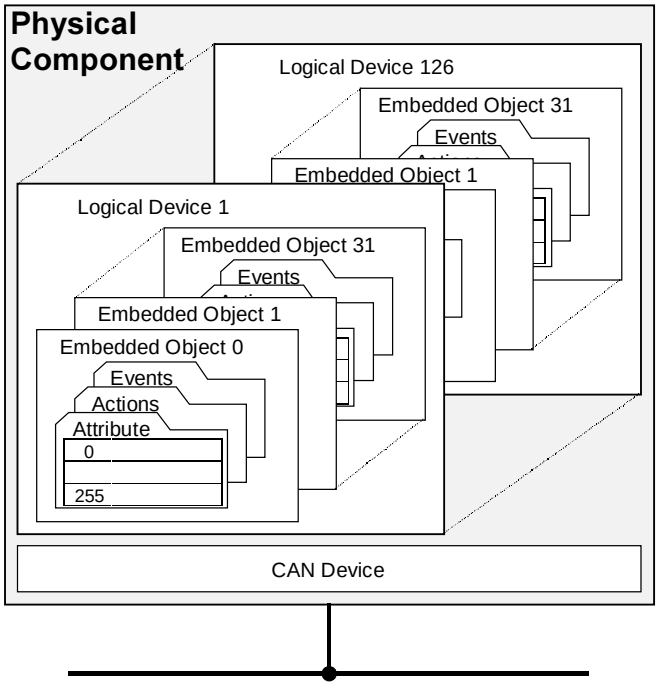

Figure 3. Embedded Objects in a SDS-device as function blocks

The topological tasks of the configuration include address and identifier management, assignment of function blocks to components, time- and resource management etc.

MANAGEMENT TOOLS

Traditionally, there are management tools provided by the systems' vendors. These tools are specific to the used fieldbus and represent the vendors' points of view. Modern software tools are components, that are concentrated around a database (Fig. 4). This database contains libraries and descriptions of available components and their capabilities as well as descriptions and initial data of the current project. The physical realization of the database is vendor specific. The software tools use this database for information exchange and data storage. A

standardized or open interface (e.g. SQL- or ODBC-interfaces, OLE-interfaces) allows the integration of business applications and PC control solutions into the set of tools.

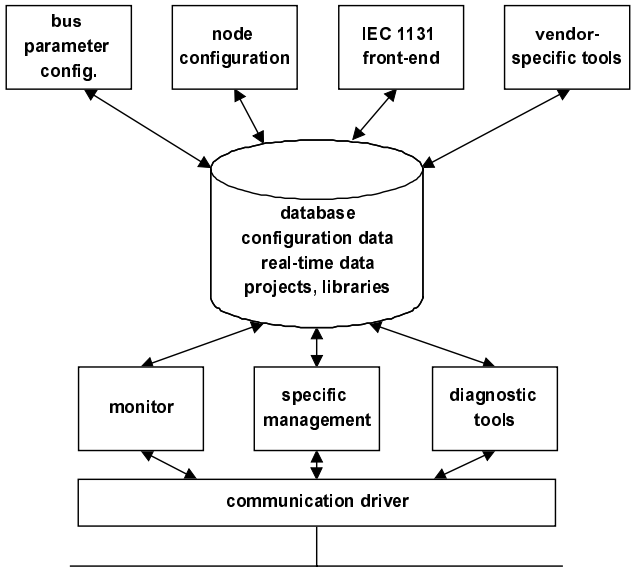

Figure 4. Principal structure of software tools for fieldbusses

INTRANET BASED MANAGEMENT CONCEPT

The integration of management tasks into Intranet based solutions has to be performed by mappings of fieldbus systems' components and function blocks to web-related objects. These objects are part of hypertext documents, that interact as user interfaces (Fig. 5). These user interfaces can be activated in any standard web browser.

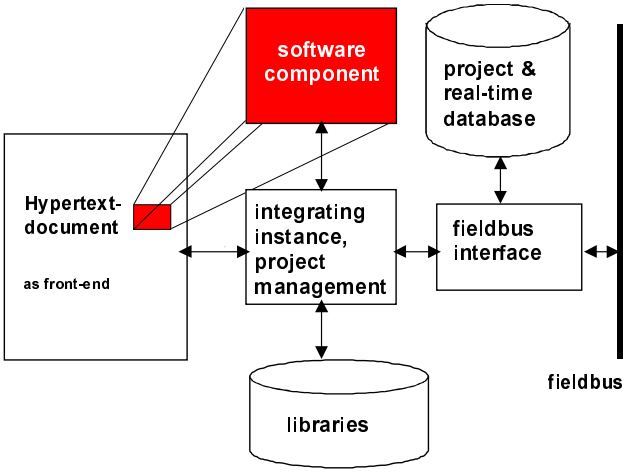

Figure 5. Structure of an Intranet based fieldbus management solution

The hypertext documents implement different views of the fieldbus system. Typically, topological views are provided by a treeview component. Fieldbus modules and function blocks views are represented by graphical symbols with interactive components. So, for example, the value of a parameter can be changed by double-clicking on the icon and entering the value into the opened dialog box. The fieldbus specific database and the fieldbus interface itself provide the data to be displayed in the hypertext document. So the consistency of the data source and the interfacing with the bus specific tools is guaranteed.

Scripts and special applications implement an integrating instance, that coordinates data access, project management, security aspects and error handling. The timing conditions are controlled by this instance, too. Using push technology, the clients can be reduced in complexity. However, in most cases timing considerations are less important for management purposes than for data acquisition tasks.

The main ideas behind using a web browser are its acceptance by the users as a de-facto standard user interface and its integrative effects. Compound documents contain user supporting information, like PDF-coded components' descriptions, interface definitions, function block descriptions and so on. As a side effect, updates of software and documentation may be performed using the same user interface - over the Internet. In addition, the tasks in remote diagnostics become much easier for the user, when using the concepts described above.

IMPLEMENTATION STRATEGIES

Following the trends in software development, there are two major strategies for that mapping. The first one is using ActiveX and DCOM, the second one is an implementation in JAVA.

ACTIVEX-IMPLEMENTATION

The growing number of PCs and Microsoft Windows applications in process control systems provides a good platform for the implementation of ActiveX software components. These components [1] can be specialized in function and are linked together using the

browser. The large number of existing reusable ActiveX components (controls) reduces development time and shows the acceptance of this technology.
The data exchange between the components and the fieldbus requires a Windows driver for the bus system. This is no limitation, because of the de-facto standard Windows (NT). On the other hand, there are integrative developments for universal ActiveX-based drivers for fieldbusses, e.g. OPC (OLE for Process Control) [2]. Using OPC, vendor-independent data exchange between PC applications and fieldbusses becomes possible. ActiveX data access objects using OPC technology are available [3]. OPC provides a chance for universal software tools for data access, handling and management of fieldbusses (Fig. 6).

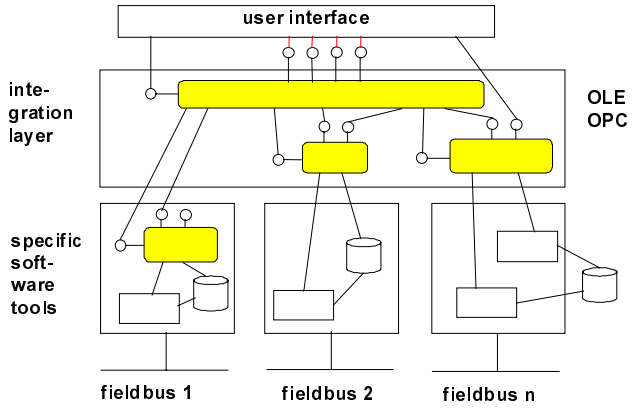

Figure 6. OLE/OPC as an integration layer for different fieldbusses

Other interesting projects providing ideas for universal communication interfaces, like NOAH and RACKS [4], are also in concern to ActiveX- and DCOM-technologies. The trends from LANs and CORBA [5] have to be considered, too.

JAVA-IMPLEMENTATION

There is no doubt, that JAVA is one of the most interesting trends in Internet technology. JAVA provides platform-independence based on JAVA virtual machines (VMs). JAVA coded programs can run as applets in web browsers or as stand-alone applications. There is a constantly growing set of application programming interfaces (APIs) [6] for different purposes, from database access, security and graphic representation up to LAN-like directory and naming services, remote procedure calls and embedded JAVA.
The JAVA enterprise API, consisting of naming services (JNDI), remote method invocation (RMI) and database access (JDBC) is very interesting for management purposes [7]. The official release is expected for the end of January, 1998. The existence of such an API shows the importance of JAVA and Internet technology for process control solutions.
Using JAVA, distributable management components can be implemented, that will run on any JAVA supported platform. However, platform independence is concerned with some problems when accessing hardware, such as interface cards for fieldbusses. Special drivers are necessary, implementing a data exchange method, that JAVA can handle. On the other hand, JAVA classes for data access can be split up into an implementation independent interface part, and a specific driver part (Fig. 7). This driver's part then can use OPC or other ActiveX-components or any other specific drivers for data access. So both of the implementation strategies are supporting each other.

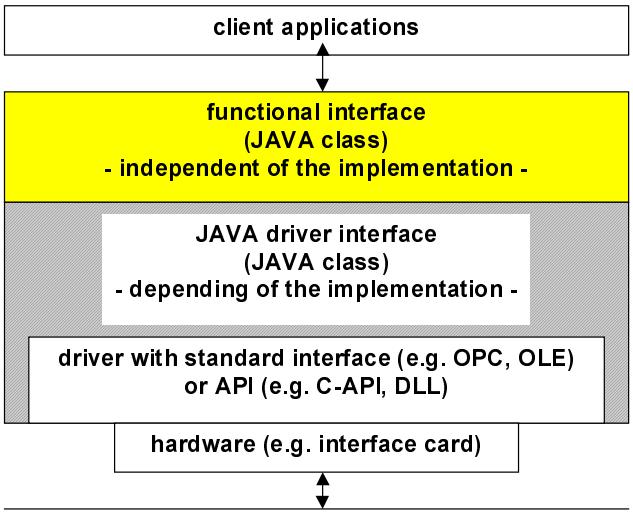

Figure 7. JAVA based data access

EXEMPLARY SOLUTIONS

Exemplary solutions have been implemented, using Windows NT and Windows 95 as a platform for drivers and web servers. The drivers provide OLE Automation interfaces, that can be used by ActiveX objects in browsers. The servers have been implemented whether as

Internet Information Server (IIS), that comes with Windows NT 4.0, or as Personal Web Server running on Windows 95 (Fig. 8).

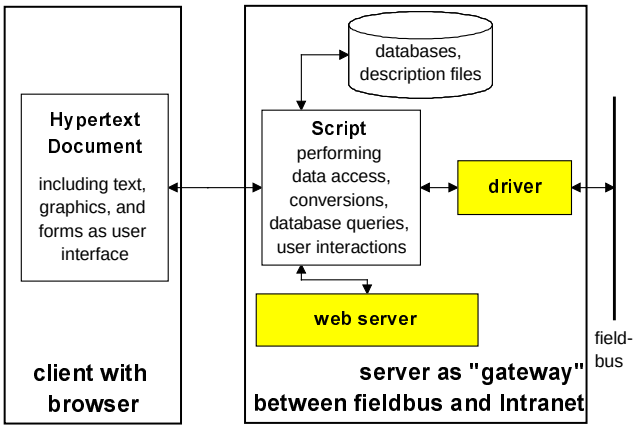

Figure 8. Data access from a hypertext document to a fieldbus

```
<HTML><BODY>
<SCRIPT LANGUAGE=VBScript RUNAT=Server></SCRIPT>
<H1>P-NET access via VIGO and INTERNET</H1><HR>
Select VIGO variable from MibOCX Control:
<object
   classid="clsid:A436C660-1DAB-11CF-B597-
      0020AF38E834"
   ID=mib>
<PARAM NAME="PhysID"></object>
<HR><INPUT TYPE=BUTTON VALUE="Show Types"
NAME="BtnShowTypes"><HR>
Documentation of "UPI" in PDF Format:
<body leftmargin=0 topmargin=0 scroll=no> <embed
width=100% height=50% fullscreen=no
src=docsrv\504008.pdf><HR>
<% set ad = createObject("VIGO")
ad.PhysID = P-NETsrc
a=ad.ExFloat %>
      .
      .
</BODY></HTML>
```

Figure 9. Code fragment of a hypertext document containing scripts and objects

Both servers allow the implementation of server extensions. These extensions provide a platform for additional services, e.g. scripts. For example, Active Server Pages (ASP), that come with Windows 95 Personal Web Server, support VBScript and JScript. While JScript is Microsoft's implementation for Java Script, VBScript is a version of the popular Visual Basic, that is adapted to be used as a scripting language in combination with the server extensions. It is as easy to handle as Visual Basic and combines the features of that language with the popularity of the HTTP protocol. Visual Basic scripts run at the server or the client. The essential feature of both script versions is the support for ActiveX components. Any OLE Automation object can be integrated into a script. Fig. 9 shows a fragment of such a script.

Running at the server, the script creates an instance of the object, performs a data exchange with the driver encapsulated by the object and then returns the value to the script. The script can use the value as an input for special algorithms or can put it directly into the generated hypertext document.

MANAGEMENT OF P-NET SYSTEMS

The P-NET fieldbus installation is connected to a PC using a P-NET card and the software driver VIGO (Virtual Interface using Global Objects) [8]. VIGO itself provides an OLE-2 Automation software interface to the P-NET hardware. This interface can be used to access P-NET data. The VIGO object represents a simple or complex variable, that is physically implemented in a P-NET module. The object provides data access methods with built-in data type information and implicit format conversion functions (OLE data types). In addition, error handling functions are implemented. The P-NET data access is performed using a VIGO object within the VBScript page. The requests are stimulated by user interaction (read request of the VBScript over the network), or can be timer controlled.

In VIGO, management tools are related to a project. The tools are concentrated around the Manager Information Base (MIB), that hosts the global definitions, like channels and modules, as well as the specific automation solution, the project. An Intranet based management system is a part of this group of applications around the MIB (Fig. 10).

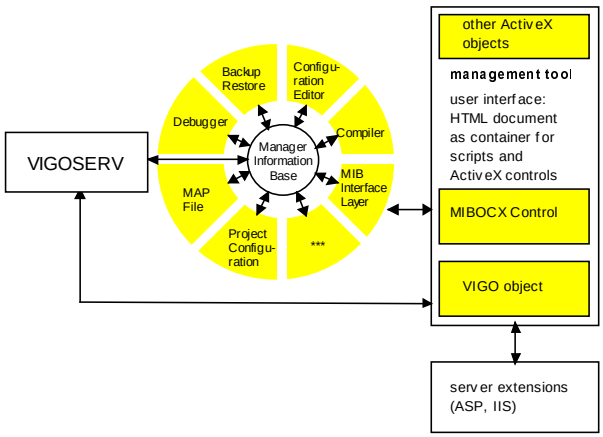

Figure 10. Relationship between the P-NET management tool and the MIB

Management tasks in P-NET cover the definition of projects, the assignment of initial values to softwire numbers, the configuration of channels, the handling of controllers. These tasks can be performed using information from the MIB. The MIB provides an OLE Automation interface, that can be used for data access from tools. ActiveX-components have been created to encapsulate this interface in a way, that it can be used in a browser [9]. Currently the implementation of JAVA classes and applets is in progress.
Based on the controls, compound documents may contain links to other data. Without any special efforts, online documentation is insertable. Using the PDF viewer ActiveX control, these files can be inserted into an HTML document (Fig. 11). Traditionally, also pictures, sounds and other multimedia components are insertable into the compound documents.

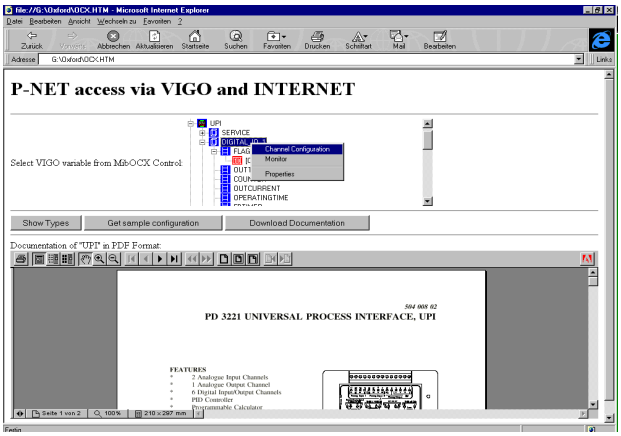

Figure 11. Screenshot of a compound document using ActiveX controls

MANAGEMENT OF A SDS-NETWORK

SDS as a CAN-based protocol provides a hierarchically organized structure of fieldbus components' descriptions [10]. This object oriented library provides the starting point for the definition of ActiveX controls, that represent the SDS devices. While SDS attributes are mapped to variables of the controls, actions and events are linked to appropriate methods of the controls. The data access between controls and the real SDS objects in the components of the network is performed by a SDS driver supporting OLE Automation. The topological view to the SDS network is provided by a treeview control. This control shows the physical SDS components as the first column in the display (top level). An expansion of the view by clicking into a component's icon shows the logical devices in the component, the next level the embedded objects of a logical device and so on. The treeview control only distinguishes from the one used in P-NET by the icons and the by bus specific data. The handling of the control is as easy as handling an explorer window. Hidden from the user, the script pages in the hypertext documents handle the data in a specific way, depending on the bus system. So the user does not need to know any details of the bus systems just for their handling. The topological structure of the SDS management implementation shows Fig. 12.

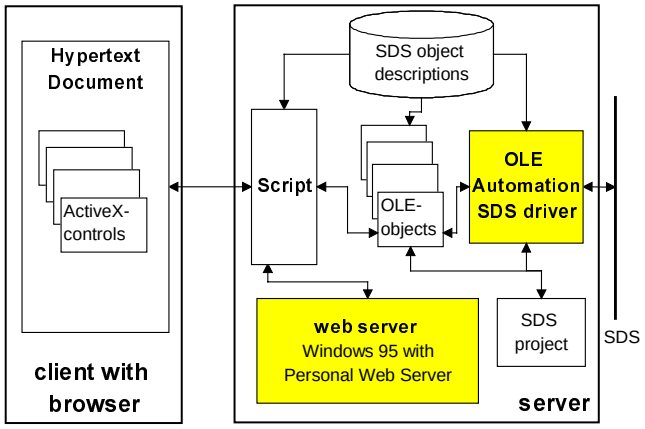

Figure 12. Topological structure of the SDS management implementation

The project specific database for the SDS implementation is set up in text format, in reference to previously used tools. However, its interface is hidden from the user, so he uses the database in the same way as other implementations of a database. The JAVA classes currently under development will be, together with the P-NET related classes, the lower layer of a JAVA implemented, universal fieldbus management solution.

FUTURE TRENDS

The current trends in software development, especially for use in network centric systems, will have a big influence to Intranet based management systems for fieldbusses. For ex-

ample, the integration of fieldbusses into the management of heterogeneous communication networks with directory and naming services, security control and other LAN based services will perform a big step towards a continuous handling of a process information system. The approach of fieldbusses themselves towards LANs will support that development, too. An important prerequisite for a continuous integration is an abstract, function block based modeling of fieldbusses. There are a number of projects featuring that idea, e.g. [11]. The integration of distributed, fieldbus based automation components into PC control solutions and traditionally Distributed Control Systems is another essential trend to - as well as a demand for - network based management solutions. And, JAVA has to be paid attention to in the future. Continuous solutions with embedded JAVA, Ethernet on the plant floor and JAVA applications and applets for control algorithms and user interface may become state of the art during the next years.

CONCLUSION

The introduction of Intranet based management of fieldbusses combines two of the major trends of last years' developments. Intranet integration won't yet provide real-time control systems, but offers solutions for data access and management in process information systems.
The users participate from the integrative features of Internet browsers, allowing a single application to act as a homogeneous shell, a user interface, to compound documents containing heterogeneous data derived from distributed systems. The growing number of reusable ActiveX components with included download and security methods makes it easy for the customers, to create a user specific environment without having to know specific details of the current installation's components. The existing networking environment can be integrated. This will reduce installation costs, and efforts for training and preparation.
The developers' benefits of that integration into Intranets are reductions in software developments. This is dedicated to the use of controls and their development, allowing to reuse previously written code or third-party solutions. The independence of the operating systems, provided by JAVA and the HTML definitions, will reduce the development efforts. The new methods in software support over the net will enhance versioning and license management. In addition, the short contacts to the user will provide improved support and enable the developer to reduce the reaction times to users' demands.
The growing developments in the Internet area and the increasing use of Intranets offer a good starting platform towards well supported, user-friendly, integrative and unique user interfaces to fieldbusses and will help to increase their acceptance.

CONTACT

Dr.-Ing. Martin Wollschlaeger
Institute for Measurement Technology and Electronics (IPE)
Otto-von-Guericke-University Magdeburg
P.O.Box 4120
D - 39016 Magdeburg
Germany

Phone: +49 (391) 67-14653
Fax: +49 (391) 561 63 58
Email: mw@ipe.et.uni-magdeburg.de

Born in 1964, Martin Wollschlaeger studied Electrical Engineering at the Otto-von-Guericke-University Magdeburg from 1983 until 1988. Since 1984 involved in projects on fieldbusses, he received his Ph. D. degree for his theses on communication oriented design of fieldbus based components in 1991. Since 1994 he's been an assistant professor at the Communication Engineering Group of the Institute for Measurement Technology and Electronics (IPE). His research topics cover universal configuration tools for fieldbusses, with focus on the influence of IEC 1131 and Internet technologies. He is involved in the development of specific software tools for P-NET and CAN.

REFERENCES

1. Chappel, D.: Understanding ActiveX and OLE. Microsoft Press, 1996.
2. n.n.: OLE for Process Control. Industry Standard Specification, Version 1.0 OPC

Task Force, August 29th 1996.
3. Chisholm, A.: OPCDATA Control V1.0. Intellution Inc., April 4th 1997.
4. Johansen, J.: P-NET in the RACKS project. 5th International conference on the P-NET Fieldbus system. Oxford, September 11th -12th 1997, proceedings.
5. http://www.omg.org/corba/
6. http://java.sun.com/products/api-overview/index.html
7. http://java.sun.com/products/index.html
8. Cramer, O.: VIGO, a Fieldbus Management System. 4th International conference on the P-NET Fieldbus system. Oporto, May 2nd - 3rd 1996, proceedings.
9. Wollschlaeger, M.: Aspects of the Integration of P-NET into Intranet Technologies. 5th International conference on the P-NET Fieldbus system. Oxford, September 11th - 12th 1997, proceedings.
10. n.n.: SDS specification. Honeywell Inc. Microswitch Division, GS 052 103 ... GS 052 107, 1995.
11. Brownlie, I.: ACORN 1479 - An Integrated Approach to Fieldbus Application Design. FieldComms UK '97, Hinckley, October 27th-29th 1997, proceedings.

DEFINITIONS, ACRONYMS, ABBREVIATIONS

CORBA: **C**ommon **O**bject **R**equest **B**roker **A**rchitecture
DCOM: **D**istributed **C**omponent **O**bject **M**odel
HMI: **H**uman **M**achine **I**nterface
HTML: **H**yper**t**ext **M**arkup **L**anguage
HTTP: **H**yper**t**ext **T**ransfer **P**rotocol
MIB: **M**anager **I**nformation **B**ase
NOAH: **N**etwork **O**riented **A**pplication **H**armonisation
ODBC: **O**pen **D**atabase **C**onnectivity
OLE: **O**bject **L**inking and **E**mbedding
OPC: **O**LE for **P**rocess **C**ontrol
PDF: **P**ortable **D**ocument **F**ormat
PLC: **P**rogrammable **L**oop **C**ontroller
RACKS: **R**eusable **A**pplication Interface for **C**ommunicating Real-Time **K**ernel**s**
SCADA: **S**upervisory **C**ontrol **A**nd **D**ata **A**cquisition
SDS: **S**mart **D**istributed **S**ystem
SQL: **S**tructured **Q**uery **L**anguage
VIGO: **V**irtual **I**nterface using **G**lobal **O**bjects