Integration of CAN-based Networks into the PROFInet Environment

Axel Pöschmann, Lutz Rauchhaupt, Thomas Werner, ifak e. V. Magdeburg

PROFInet is an international standard for Ethernet-based industrial communication systems. It considers the integration of fieldbus systems, such as CANopen, which is well established in the field level of industrial automation.

In the paper the different concepts of the PROFInet model are introduced. In particular the so called migration concept is explained, which deals with the possibility to integrate fieldbus systems into the PROFInet environment using a so called Proxy device. For CANopen the architecture of such a Proxy is described and the relation to the CANopen model elements is discussed. Finally an implementation concept of a CANopen Proxy with static CANopen device assignment is presented.

Ethernet meets CAN

Solutions which combine Ethernet-based communication (mostly also IP-based) and CAN-based communication have been known for a long time (e.g. [1], [2]). However, in factory automation, the open integration of field level networks in factory-wide information networks is demanded by end-users. One approach of integration is the TCP/IP and DCOM based international standard PROFInet for distributed components. Other solutions are known as Ethernet/IP, FF HSE and IDA. An overview of all these systems is given in [6].

The main goal of the PROFInet architecture is to provide a common platform or framework to combine mechanical, electrical, and software components under a common engineering view. Even if PROFInet includes hard real time solutions and soft real time solutions based on Ethernet, the integration of standard fieldbus systems like CAN is an important point, saving investment by integrating the installed base and combining useful technologies. A so called Proxy is used to represent the fieldbus as a whole or even single stations as components in the PROFInet sense. CAN-based automation networks are used world-wide in control systems and inside machines. Therefore, the design of such a Proxy will become very important. The CAN Proxy for PROFInet must combine the configuration, diagnostics and process data facilities of both systems to achieve a transparent and consistent view inside both systems.

PROFInet Model

PROFInet is a modern concept for distributed automation standards. It is based on Ethernet and is able to integrate existing fieldbus systems. Automation systems usually consist of several sub-units, which act as technological modules largely autonomously and coordinate their interactions by means of a manageable number of synchronisation signals, sequence control signals and information exchange signals. These technological modules consist of a combination of mechanical, electrical and software parts, i. e. parts belonging to an intelligent function. The technological modularisation simplifies the re-use of pre-manufactured and pre-tested components. In order to communicate with other components within a distributed system, a component supports a defined interface to the outside world. PROFInet unifies these interfaces using the following concepts.

Engineering Concept

The PROFInet standard contains a manufacturer independent engineering concept. Manufacturer or user specific enhancements of the functionality can be enabled by means of facets. The engineering model can be used to develop own engineering tools, e. g. a connection editor. Using such an editor, components of different vendors can be connected with each other.

PROFInet components are described with XML files which are used within the engineering system. The used XML scheme is defined in the PROFInet standard [15]. An XML file is generated manufacturer specific within a so called componer and is part of the PROFInet device. During the engineering process all communication relations are defined simply by graphical wires. After the system definition with the connection editor, all connections can be loaded into the devices, which belong to the plant network. Furthermore, device-specific data is loaded, if defined. The engineering tool supports an online view to check the states of the devices and all connections. This is the engineering infrastructure which could be used also for CANopen sub-networks.

Runtime Concept

The runtime communication concept of PROFInet is based on the DCOM Wire Protocol and TCP/IP. For real time purposes PROFInet also offers an optimised communication channel (SRT). Components are described with the object model. The data is transferred via the configured communication relations, using the object protocol.

The source code of the PROFInet stack is available for different operating systems. It can be downloaded for free by members of the PROFIBUS user organisation. This software consists of the following layers: DCOM, connection oriented RPC and DCOM application. In Figure 1 the DCOM application is shown containing the runtime objects physical device (PDev), the logical device (LDev), the runtime automation (RT-Auto) and the active control connection object (ACCO).

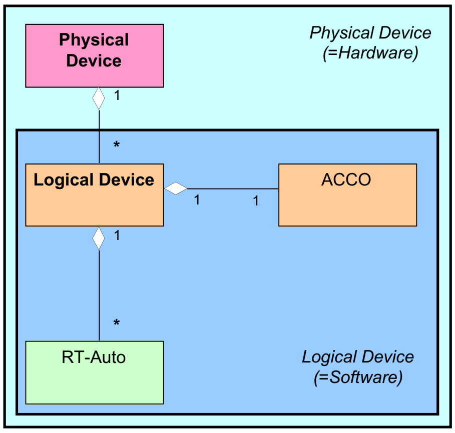

Figure 1: PROFInet – Runtime Object Model [7]

Each PDev represents one physical component and provides its access to an IP network. The implemented software is represented by one or more LDevs. It enables navigation to the further objects. The ACCO handles all connections. The automation functionality is represented by the RT-Auto.

In the case of CANopen the runtime environment is located in a special device, the so called Proxy. The CANopen devices remain unchanged and are nevertheless part of the PROFInet system. The way how this is accomplished is explained in the following chapters.

Migration Concept

An open migration concept enables the integration of devices, connected by fieldbus systems, to PROFInet using a Proxy. There are two main strategies available to integrate a fieldbus based automation sub-system. The first one is to hide the fieldbus system and its devices behind one PROFInet component as depicted in Figure 2. In this case the underlying communication infrastructure is not of interest. From the PROFInet point of view it is not visible. Only the relevant inputs and outputs are made available to connect them with other components.

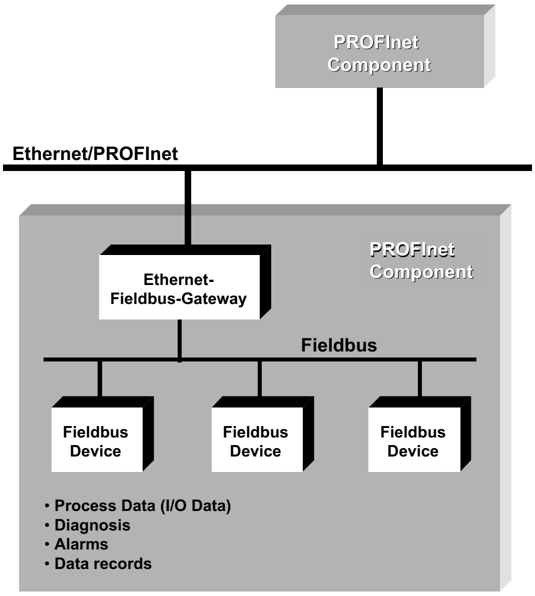

Figure 2: Sub-system as PROFInet component

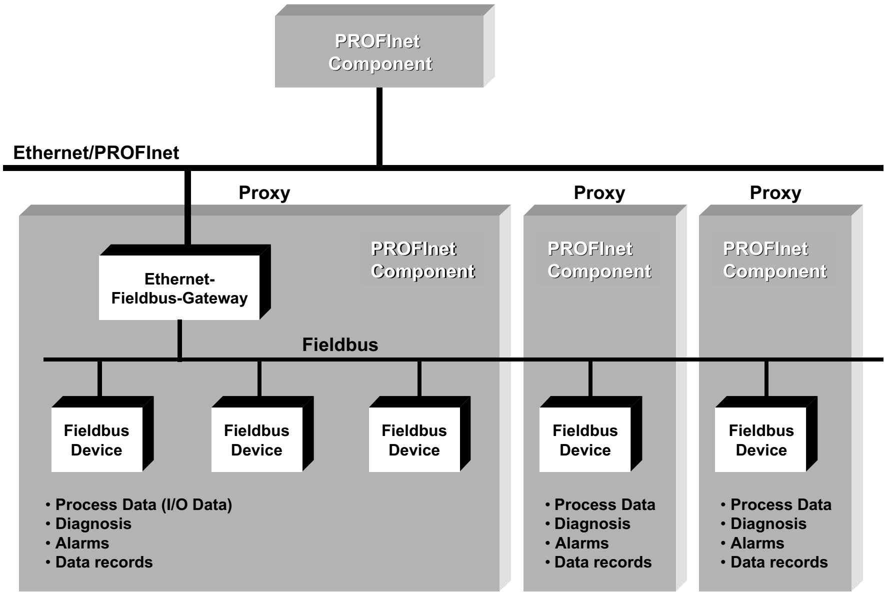

Figure 3: Fieldbus devices as PROFInet component

If it is necessary that also fieldbus devices are visible in the context of PROFInet the Proxy concept must be applied. This is shown in Figure 3.

In the PROFInet standard [7] an example is given in detail using PROFIBUS. Figure 4 shows the appropriate object model. It consists of a PDev that complies the whole PROFIBUS system. Every slave device is represented by one LDev. A special mapping provides access to the cyclic (process) data of the slave devices. The data is available at the interface of the LDevs and can be connected with other LDevs (inside or outside PROFIBUS). Local connections (inside) are detected and can be realised directly by the PROFIBUS master. Furthermore, the PROFIBUS diagnosis and the operating state of assigned slaves must be mapped to available PROFInet mechanisms.

Physically the Proxy contains the PROFIBUS master, the PROFInet stack and an Ethernet controller. The assigned slaves are connected to the Proxy via PROFIBUS.

A similar solution is supposed to be developed for CANopen. In the following chapter different approaches are shown that are possible.

Architecture of a CANopen Proxy

The common architecture of a CANopen Proxy is depicted in Figure 5. It contains the PROFInet software, an operating system, an Ethernet controller, a CANopen stack implementation and a special PROFInet system adaptation, that combines all components. The main tasks of a Proxy are the implementation of local

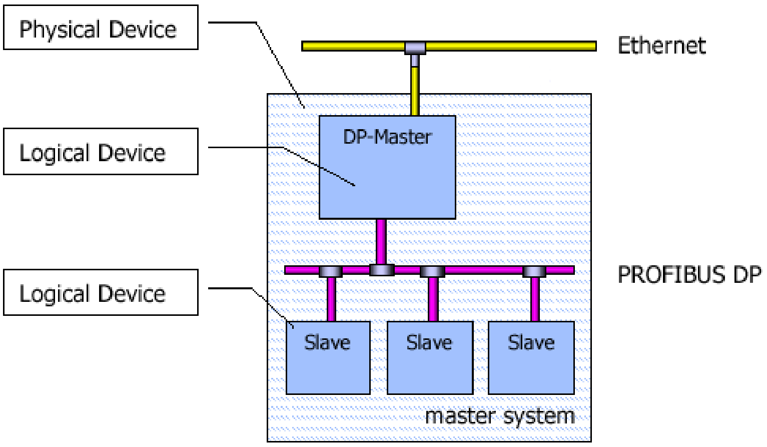

Figure 4: PROFInet - object assignement

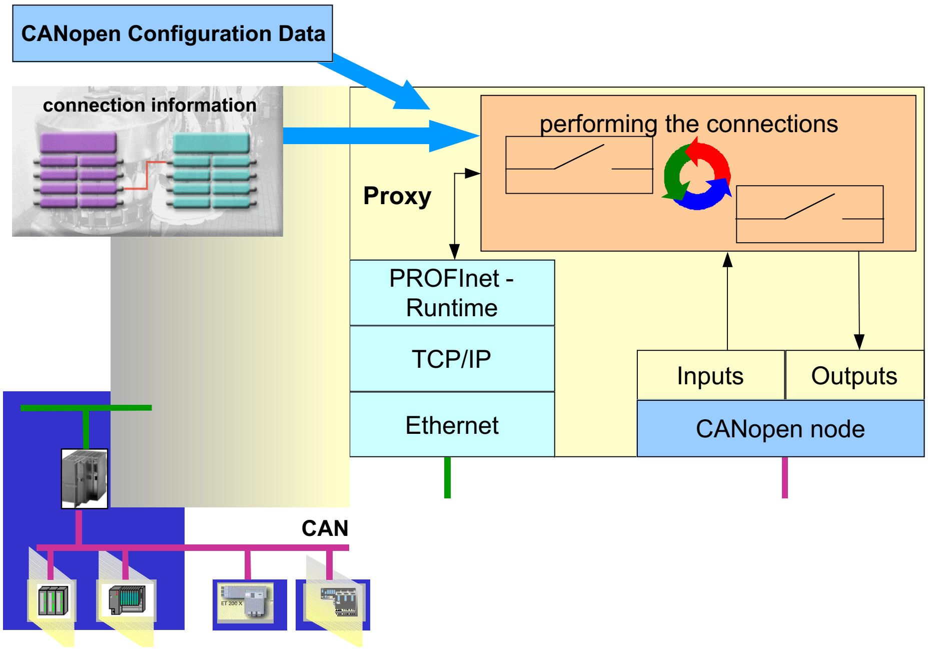

Figure 5: Proxy architecture

connections, the instantiation of objects for every assigned CANopen node as well as the mapping of data types, diagnosis and operating states between CANopen and PROFInet.

A CANopen Proxy needs special configuration information which has to be transferred into the device via a proprietary way, like ftp. It is possible to start this process from the connection editor environment. A special componer generates the necessary XML files and the configuration data. With this data the Proxy has the ability to instantiate the necessary runtime objects (loadable objects).

The previous described migration strategies can be applied as follows.

***Type A**: PROFInet component is a CANopen subsystem*

Building one PROFInet component with a hidden CANopen system is the easiest way. This implementation is not a real Proxy. It is only the creation of a pre-manufactured and pre-tested component, that performs its own automation task.

The real Proxy concept, described in the standard [3], can be separated into the following two concepts. The reason for the difference between these concepts can be found in the engineering requirements. Either fieldbus specific tools shall be used or not.

***Type B**: PROFInet component is a CANopen device (static)*

Every CANopen node is a separate component. From the object model point of view the Proxy contains one PDev and in addition for each CANopen device one LDev, ACCO and RT-Auto. In this case it is not possible to configure the fieldbus from the view of PROFInet. It is a static configuration. Furthermore the CANopen sub-system is not visible within the network view of the connection editor. Reconfiguration can be done with separate CANopen tools and processes.

***Type C**: PROFInet component is a CANopen device (dynamic)*

The modelling is like type B, but in this case there is a total integrated Proxy implementation. That means the CANopen sub-system can be defined within the connection editor and the download process combines the transfer of configuration data and connection data. Therefore it is necessary to create a special facet for the own componer and for the own download mechanism that is used within the connection editor.

Relevant CANopen Model Elements

As described in the previous chapters, several concepts are possible to define a PROFInet Proxy. These concepts concern mainly the visibility of CANopen devices in the PROFInet environment and therewith the configuration of the CANopen subsystem.

Next the relation to the relevant CANopen model elements is explained.

***Type A**: PROFInet component is a CANopen subsystem*

From the perspective of PROFInet, the CANopen system is hidden in this case. Only the contents of the process data objects (PDOs) of the system, or a part of them, are visible as so called properties of a PROFInet component. Therefore, the CANopen subsystem is configured as usual, this means

- by the control application for smaller systems, e.g. using the pre-defined connection set
- by a CANopen configuration tool, which stores the configuration data into the CANopen devices non-volatile during the system commissioning or
- by a CANopen configuration tool, which generates a configuration data set of the system and stores it into a configuration manager.

During runtime the application in the Ethernet/CAN gateway uses services of the CANopen stack as well as of the PROFInet stack to transfer the process data between the CANopen subsystem and other PROFInet components.

Furthermore, the application of the CANopen subsystem has to provide information about exceptional states to the PROFInet system as far as they are related to the process data.

This concept is more a special PROFInet device than a CANopen Proxy for PROFInet.

***Type B**: PROFInet component is a CANopen device (static)*

In this case the CANopen device is part of the PROFInet infrastructure. However, it is still configured using the above mentioned methods. The contents of the configured process data objects is visible as properties of a PROFInet component Therefore, the input and output data of the CANopen device can be mapped to input and output data of other PROFInet devices using the available PROFInet engineering tools. Furthermore, the CANopen devices may be added and removed flexible to the PROFInet system.

The prerequisite is, that the device configuration data is provided by an XML file. The XML file format is specified in the PROFInet standard. It seems to be reasonable to develop a so called componer, which generates these XML files for CANopen devices. The componer could be integrated into existing CANopen configuration tools.

The generated XML files can be imported into the connection editor. Depending on the application, connections between PROFInet components (also CANopen devices behind the Proxy) can be established or removed off-line. After downloading the connection data, the CANopen Proxy is able to transfer process data between CANopen devices and other PROFInet devices. Furthermore, it is necessary that the CANopen devices can be controlled by the overall application. For this purpose, the PROFInet device state machine has to be mapped to the CANopen NMT state machine. In addition, a solution has to be implemented to map the information about exceptional states of the CANopen device into the diagnostic functionality of the PROFInet system.

This concept is an intermediate-step. Specific network information is not managed by the PROFInet engineering tool, but by the CANopen configuration tool. The CANopen specific network information is supposed to be integrated into PROFInet engineering tools, if required. This leads to the next concept.

***Type C**: PROFInet component is a CANopen device (dynamic)*

In the third case the configuration of the entire system, including all CANopen components, is managed by PROFInet tools. This means the CANopen system and device configuration is not done directly by CANopen tools. However, these tools may be integrated into the connection editor, using a so called facet. This approach corresponds to the use of a configuration manager in CANopen systems.

For this approach, an XML file must also be generated for the CANopen Proxy. In addition, preparations, as described for the previous concept, are necessary.

The description of the three concepts has shown that mappings are required from PROFInet to the CANopen.

First of all it concerns the basic data types. A direct link is possible between UNSIGNED8, UNSIGNED16, UNSIGNED32, INTEGER8, INTEGER16, INTEGER32, REAL32 and REAL64 whereas the transfer syntax has to be regarded. For all other basic data types, especially for e.g. UNSIGNED24, BOOLEAN, TIME_OF_DAY and strings,, conversion rules have to be specified. The definition of structures and arrays of basic data types are foreseen in both systems.

Next the mapping of the PDO and SDO services to PROFInet services has to be specified. In particular the mapping of the objects in PDOs is to dissolve.

PROFInet describes a diagnosis mechanism which has to be supported by all PROFInet devices. In the case of CANopen devices the emergency object and the Boot-Up object has to be considered.

Finally it is necessary to map the NMT state machine to the PROFInet device state machine. Namely the state Stopped and the sub-states Reset Application and Reset Communication have no equivalent in PROFInet and must be generated by other means.

Conclusively it can be said that the combination of CANopen and PROFInet is technically possible. For a first step the presented Type B is supposed to be preferred. That is why the following chapter deals with an implementation concept of this variant.

Implementation Concept

A first implementation of a CANopen Proxy is currently being developed in a research project, which is supported by the Forschungsgesellschaft Computer Automation FGCA (Research Society for Computer Automation). In this project small and medium sized companies have the possibility to gain first hand experience with the PROFInet technology and the integration of CAN-based networks.

The implementation concept of the CANopen Proxy is as follows. The hardware platform is an embedded industrial PC solution with x86 architecture, 16Mbyte RAM and 8MByte Flash memory. The board contains an Ethernet controller with 100MBit/s and a CAN controller. The software platform is an embedded Linux complemented by PROFInet software version v1.2 with specific Linux adaptations [4], [5]. Furthermore a CANopen driver is part of the existing software. The PROFInet DCOM application is being enhanced by the Proxy functionality, which includes access to the CANopen driver. Therefore, loadable objects and a mechanism to store and read the configuration data, defined in the componer (file interface), is being implemented. This PROFInet device will be integrated into an existing PROFInet multi-vendor environment at ifak. On the other hand conventional CANopen devices (I/O devices and servo inverters) will be connected to the CANopen Proxy.

Conclusion

Today it is accepted by experts, that heterogeneous communication architectures are indispensable to meet the requirements of future industrial automation systems most efficiently. In the end, it is a matter of saving investments and finding the networks which are best suitable for the different parts of an application.

This paper deals with the PROFInet Proxy approach in the view of CANopen networks. It has been shown that different concepts are possible, depending on how the CANopen network and its devices are used in the relevant part of the application. Some clues are given, on what is to be considered to combine the two technologies CANopen and PROFInet. A more detailed guideline is currently being worked on as part of the above mentioned research project.

In this paper the integration of CAN-based networks into the PROFInet environment was described in the view of the CANopen communication profile. However, these concepts can easily be extended to systems which use specific device profiles e.g. the profile for drives and motion control.

Finally it is strongly recommended to provide a common specification of mapping PROFInet to CANopen to get a standardised way of interfacing. This is also the precondition to be regarded in PROFInet tools, which is necessary in the case of complete integration of CANopen devices as components in PROFInet.

References

[1] K. H. Ang, K. H. Tang, R. T. McLaughlin, B. Levin, "DeviceNet over TCP/IP Gateway", 6th International CAN Conference (iCC' 99), Turin, Italy, Nov. 2nd-4th 1999, Proceedings p. 06-19 to 06-26

[2] G. Gruhler, G. Nusser, D. Bühler, W. Küchlin, "Teleservice of CAN Systems via Internet", 6th International CAN Conference (iCC' 99), Turin, Italy, Nov. 2nd-4th 1999, Proceedings p. 06-02 to 06-09

[3] IEC 61158: “Digital data communication for measurement and control - Fieldbus for use in industrial control systems”, Type 10, IS, IEC 2001

[4] A. Pöschmann, Th. Werner, "PROFInet - Die Linux-Anpassung", Computer & AUTOMATION, Heft 10, 2002

[5] A. Pöschmann, "PROFInet – Die Linux-Protierung für Embedded Systeme“, Embedded Intelligence 2002, Nürnberg, 19.12 - 21.12.2002

[6] A. Pöschmann, Th. Werner; "Ethernet in der Automation", Studie ZVEI, Frankfurt, 2003

[7] PROFIBUS Guideline, “PROFInet Architecture Description and Specification”, Version V 2.0, January 2003, PNO Karlsruhe

Axel Pöschmann
ifak Magdeburg
Steinfeldstrasse 3 (IGZ)
D-39179 Barleben
Phone: +49 39203 810 - 26
Fax: +49 39203 811 00
E-mail: apoe@ifak.fhg.de

Dr. Lutz Rauchhaupt
ifak Magdeburg
Steinfeldstrasse 3 (IGZ)
D-39179 Barleben
Phone: +49 39203 810 - 67
Fax: +49 39203 811 00
E-mail: lra@ifak.fhg.de

Thomas Werner
ifak Magdeburg
Steinfeldstrasse 3 (IGZ)
D-39179 Barleben
Phone: +49 39203 810 - 74
Fax: +49 39203 811 00
E-mail: thw@ifak.fhg.de