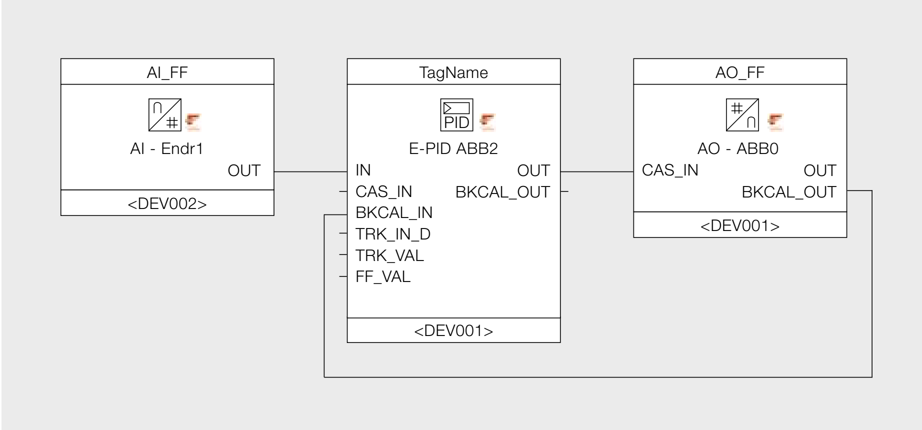
The fieldbus outside the field

Soft FF reduces commissioning effort by simulating Foundation Fieldbus

MARIO HOERNICKE, PHILLIP WEEMES, HEINO HANKING – Computer simulation is a valuable tool whose importance is growing in all areas of engineering. In industrial control, simulation is an effective way to minimize effort for test preparation, provide imitations of comprehensive systems under test, enable the systems to be properly dynamically tested on the shop floor and, hence, to reduce changes to device applications during the commissioning effort and so ultimately reduce plant downtime. Simulation is a standard part of the factory acceptance test (FAT) of control systems, and is also used for operator training. For fieldbuses and fieldbus systems such as Foundation Fieldbus (FF), however, this simulation ability is lacking. The FAT can be performed for sample loops but not for complete installed solutions with their complex interactions and distributed control functionality. Because FF devices can execute control loops, the FAT technique used is error prone and does not guarantee correct functional behavior of the entire system. To address this, the development of an FF simulator was launched. The technology is making it possible to dynamically test control-in-the-field applications for FF during FAT. This article looks at the technical challenges that had to be solved as well as the architecture and resulting technology of the Soft FF.

1 Typical use case of control-in-the-field, AI-PID-AO loop



The ARC Advisory Group has said, "Fieldbus has made its way into the culture of process automation and ARC expects growth will virtually skyrocket during the next several years" [1].

The increasing complexity of plants creates a need to distribute control code in order to gain a control balance without overloading centralized control units. This can be achieved using Foundation Fieldbus (FF) [2]. However, this approach introduces problems regarding the factory acceptance test (FAT). Unfortunately, FF networks can only be partly tested during FAT because usually the hardware is not present at that point. The configuration is thus commissioned without the certainty that the FF networks are properly engineered and parameterized. Hence, most of the effort in testing FF networks, especially the control-in-the-field applications, occurs during commissioning.

Title picture
Process plants use myriad control components. The fieldbus enables the exchange of information between these.

Commissioning hours are more expensive than hours spent during FAT, leading to higher total project costs (see the "factor 10 rule" [6]). Furthermore, the untested parts of FF applications decrease confidence in the solutions and thus induce a high cost of poor quality (COPQ). The solution implemented in Soft FF was to simulate FF devices or networks for dynamic FATs to imitate the behavior of real hardware.

Soft FF provides an open architecture that can seamlessly integrate enhanced or customized fieldbus types.

Challenges and requirements

As a field control system (FCS) [5], Foundation Fieldbus is not a controller-centric fieldbus system, but can execute control strategies directly on devices. Hence, the latter cannot be treated as simple peripheral devices that usually only provide and consume process values. FF devices are able to execute control strategies. Instead of several loops typically being executed on the same device, as would occur in controller-centric systems, each device performs its own function blocks.

Interconnecting the function blocks of several devices is a way to achieve a powerful control strategy. The application is formed of interconnections between defined and standardized function blocks. The development of control strategies can thus be described as the configuration and interconnection of powerful separate functions.

Typical loops consist of an analog input block, a PID (proportional-integral-derivative) block and an analog output block → 1. The sensor executes the analog input block and the actuator executes the PID and the analog output block.

As mentioned earlier, the function blocks of FF are standardized and specified by Foundation Fieldbus [3, 4]. The function blocks consist of several sub-blocks, each with a given functionality, such as filtering, linearization or alarm handling. The specification also defines parameters for I/O connection, but also for the fieldbus' configuration. Furthermore, the behavior of the blocks is specified, but not the required algorithm. Therefore, every vendor can use its proprietary programming language and style for the implementation, but needs to fulfill the defined behavior.

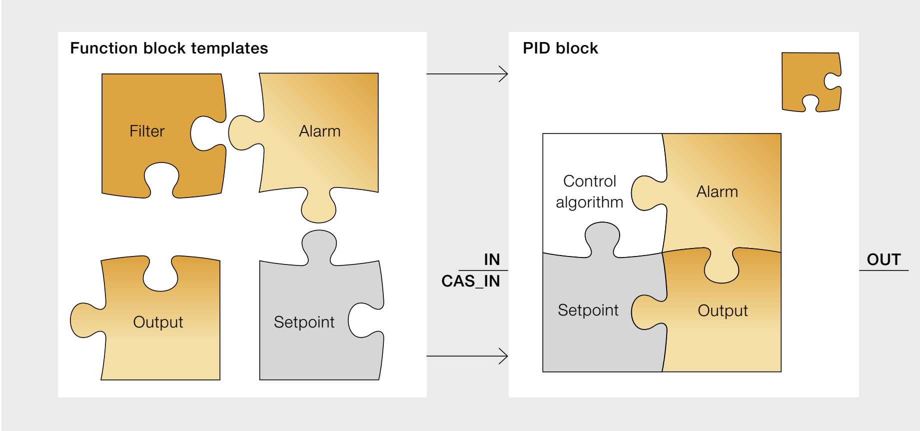
Challenge 1: Fieldbus development

Although the function block design within FF is based on a standard, there is room for interpretation. In most cases, vendors implement the standard functionality in order to be FF compliant, but realize advanced add-ons. Such fieldbuses therefore implement the standard behavior as a subset of an advanced behavior. The function block remains exchangeable against similar blocks from other vendors, but with restriction to the standard functionality.

2 Excerpt of the common blocks of standard FF fieldbuses

FB \ Block	Simulate	Simualte_D	Convert	Cutoff	Filter	Output	Alarm	Mode	Optional invert	Output track	Setpoint	Bias & gain	Selection
AI	x		x	x	x	x	x	x					
DI		x			x	x	x	x	x				
ML					x	x	x	x		x			
BG						x		x		x	x	x	
CS						x		x					x
PD					x	x	x	x		x	x		
PID					x	x	x	x		x	x		

3 Function block templates and their usage



For the development of the Soft FF, this was one of the main challenges. It led to three major requirements regarding fieldbus integration and implementation:

– Function blocks, which had to be integrated, needed to be implemented according to the standard in order to execute the generally used standard functionality. Since each of the function blocks is quite complex, a modular approach had to be found to reuse as much as possible.
– Soft FF was required to provide an open architecture in order to seamlessly integrate enhanced or customized fieldbus types, without reengineering the software of the Soft FF core.
– Enhanced fieldbuses, which are not integrated into the simulator, needed to be mapped down to the standard fieldbuses in order to have basic interoperability features available for the simulation.

Challenge 2: Communication interfaces

Additional challenges that had to be solved were the communication interfaces of the Soft FF. The Soft FF needs to communicate in a manner similar to the real subnet. This means it requires a connection to the engineering tool, the human-machine interface (HMI) and other simulators. In order to establish communication during simulation, two requirements had to be met:

– Communication to the engineering tool needed to be established by using the high-speed Ethernet (HSE) protocol. Since the simulator must be executable on a standard PC, H1 (two-wire interface) communication could not be established and therefore only the HSE communication could be used.
– Providing the OPC [1] server namespace was required. The Soft FF must be able to fill the OPC servers namespace with process and fieldbus parameters in order to be able to connect to the operator graphics. This also enables the Soft FF to be connected to several different types of HMIs.

Challenge 3: Configuration

The third challenge that had to be resolved was the configuration of the simulator itself. The following major requirements applied:

– Since the simulation only works on the HSE level and does not provide the H1 protocols that are required for downloading configurations to devices, a way had to be found to configure the simulator from the information present in the engineering tool.
– The topology of the subnet is not known until the configuration of the simulator has been effected. This means that before the application can be simulated, the topology must be rebuilt.
– An intermediate file is required because the simulator must be able to perform independently. Hence, this intermediate file can be copied to other PCs in order to configure the Soft FF there.

Technical solution

The implemented technology is largely shaped by the architecture of the Soft FF core and the architecture for the fieldbus.

Function-block

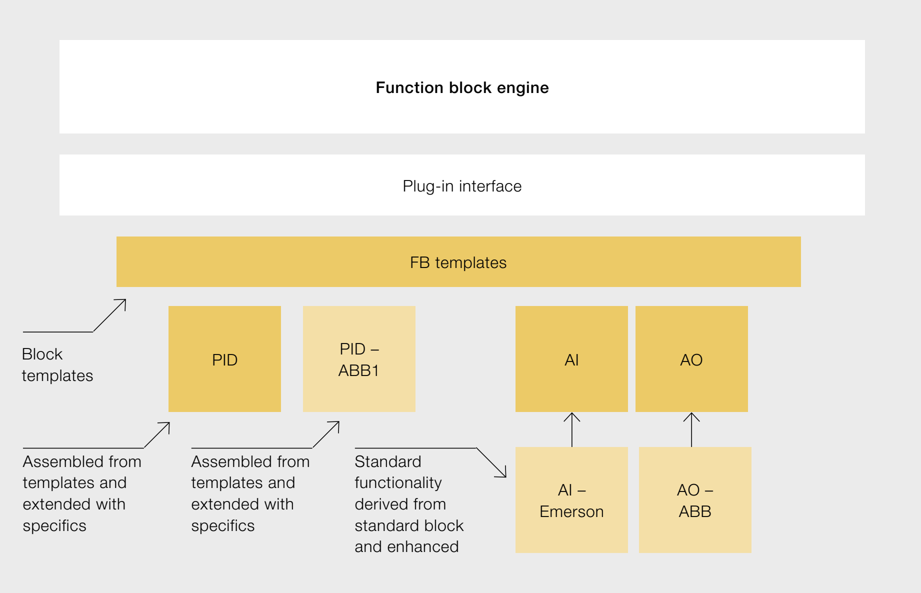
Development of the standard function blocks follows a principle that is known from IT: The template method is applied (according to GoF:325 [7]). This method

The development of control strategies can be described as the configuration and interconnection of powerful separate functions.

Footnote

1 OPC (OLE for Process Control) is a standard specification maintained by the OPC Foundation.

4 Function block engine of the Soft FF



Commissioning hours are more expensive than hours spent during factory acceptance testing, leading to higher total project costs.

provides code snippets for the internal blocks, which are used by several fieldbuses. Analyses showed that some internal blocks are used more than once within the function blocks → 2. Therefore their functionality can be generalized.

Development of function blocks according to the standard (but also of vendor-specific function blocks) can be implemented based on the templates, and therefore basic characteristics of the blocks can be used out-of-the-box → 3. The parts provided by the templates and snippets can be used like a puzzle and are afterwards enhanced with the function-block-specific control algorithms.

Using templates enables the function blocks to reuse the required functionality for the generic parts. Additionally, implementing the specific interfaces and inheriting the template classes makes the fieldbus discoverable by the Soft FF. Each function block that is implemented in this way can be used inside the Soft FF. Vendor-specific and enhanced fieldbuses can be derived from the standard fieldbuses. The standard functionality is provided and only the enhanced functionality and additional parameters need to be added → 4. A seamless integration of function blocks through implementation of the template classes, or deriving them from an integrated fieldbus, makes the Soft FF open for future expansion – and vendor-specific fieldbuses.

For the mapping of unknown fieldbus types, a parameter specified in the FF specification can be used. Each fieldbus type has a defined profile number that identifies the comprised standard functionality. Based on the profile, the corresponding standard fieldbus type can be identified and therefore, the type can be mapped to the standard fieldbus.

Evaluation of the PID control algorithm

In order to demonstrate that the simulated function blocks conform to the behavior described in the standard, the simulated PID block was compared with the PID function block from Endress+Hauser's temperature transmitter (TMT165). For this purpose, the PID controller was set up with a simple feedback loop between its input and output (a control process with a gain of one and without timing constraints). The control constants were configured to ensure stable closed-loop control.

The resulting closed-loop control was simulated and executed on the real device. In different states, the values at the output pin and the corresponding set point were measured over several iterations. The measurement used OPC for the real device and proprietary software to store the values into a file for the simulated part.

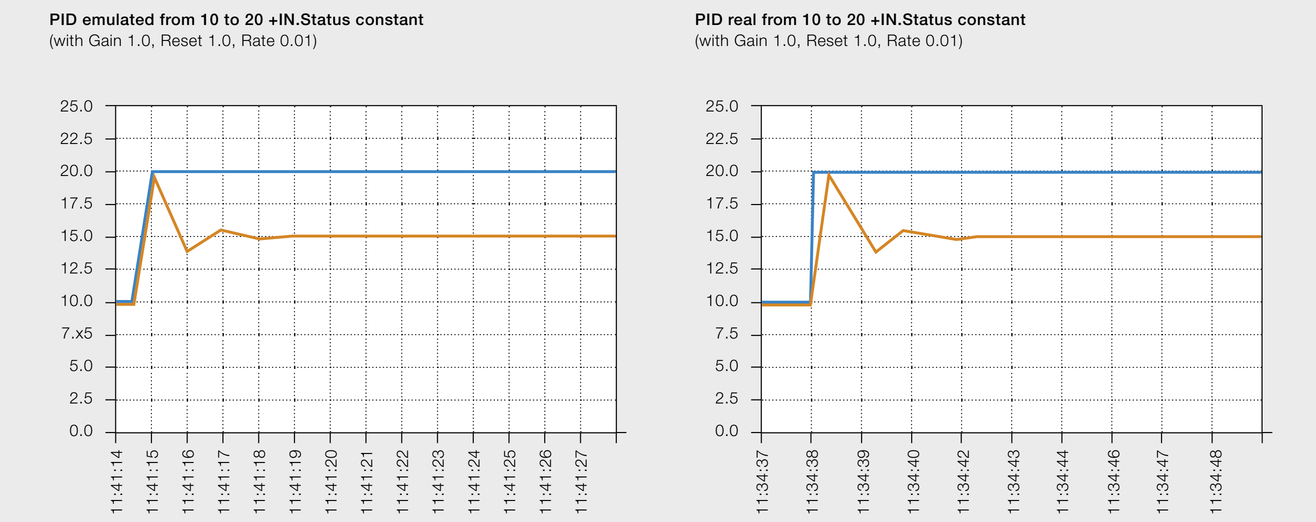
A step response from a stable state at set point 10 to a new set point of 20 is shown → 5. The status at the input pin has been configured to be constant, which means that the integrative part of the PID control is disabled. Therefore, the output of the PID will never reach the set point but another stable value that depends on other constants. The results show an amazing similarity between the simulation and the real PID.

Simulation core architecture

Simulation of FF includes the communication to the engineering tools and the process control system. Thus, the second aspect of the Soft FF is the architecture and the technology of the Soft FF core that finally executes the function blocks and establishes the communication to the engineering tools and the process control environment.

Firstly of all, the Soft FF core needs to be configured in order to obtain information about the HSE subnet. A soft controller is usually configured by downloading the compilation of the IEC61131-3 code. In

5 Example test of the PID function block (IN.Status = constant)



the case of the Soft FF this kind of configuration is hard to manage because the device topology is not known before the configuration takes place, but is required for performing a proper download. Therefore, an XML needs to be designed that represents the topology of the FF network.

Since FF devices are not freely programmable, but are developed by instantiating and parameterizing standardized function blocks, the application can easily be represented in XML. The XML can be generated from different tools, rendering, the FF simulator configuration interface independent of the engineering tool.

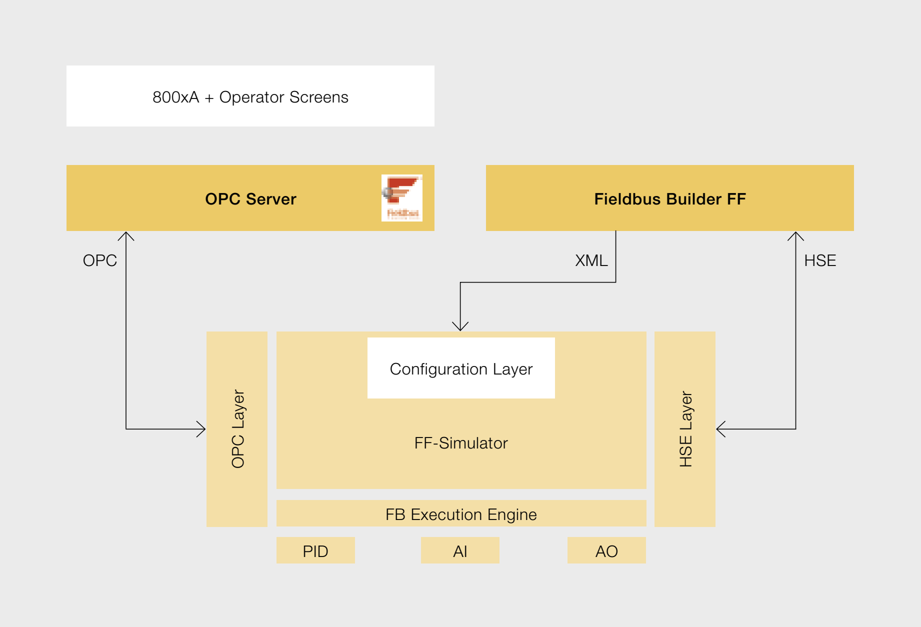
When the Soft FF is properly configured, it provides an executable image of the applications from the subnet. During execution, it must by capable of communicating with the engineering tools and the process control environment. Although the main purpose is not the simulation of the network behavior, two communication protocols have been implemented:

- HSE communication: HSE communication is implemented in a separate layer. It is partly dependent on the engineering tool for the simulator, since some protocol specifics are not simulated, therefore, it has to be replaced if another tool is used for online access. Nevertheless, the software is designed to provide an exchangeable interface for HSE communication.
- OPC communication: OPC communication is independent of HSE communication. The HSE and H1 devices usually provide the process values to the OPC server. In the simulation, this is not possible because real devices are not present. Since the OPC server is also able to grab the values directly from the HSE layer, it is not independent of it in Soft FF. The HSE layer has to provide the process values, alarms and events in order to ensure OPC communication. Therefore, the OPC layer is a special layer that can be treated as separate from the rest, but that must be changed when the HSE layer is changed.

The increasing complexity of plants creates a need to distribute control code in order to gain a control balance without overloading centralized units.

The behavior of blocks is specified, but not the required algorithm. Every vendor can use its proprietary programming language and style, but needs to fulfill the defined behavior.

6 Interfaces of the Soft FF



Soft FF will be the fuel that permits Fieldbus to lift even higher into the orbit of process industries.

The scenario for the System 800xA environment is shown in ➔ 6. The HSE layer is implemented to fit the needs of the Fieldbus Builder FF, and the OPC layer is implemented to feed the OPC server FF with the process, alarm and event values. Therefore, Soft FF technology is tightly integrated into the System 800xA environment. Additionally, the configuration layer and the function block execution layer, which is also used for dis-covering the function block plug-ins.

An effective solution

FCS simulation is a strong necessity for testing during engineering and FAT. Approaching FF applications in a manner similar to IEC 61131-3 applications on controllers, by means of a Soft FF, increases confidence in the engineered solutions and reduces COPQ and commissioning hours for FF projects. Additionally, the FATs can be applied to the FF solutions in their entirety in order to ensure correct functional behavior of the interoperable fieldbuses, as well as the parameters of the fieldbuses, for the communication to controllers and cross-communication between the devices and H1 segments.

Soft FF is easy to handle, open to integrating standard, vendor-specific and user-defined fieldbuses and provides the required connectivity for engineering and FAT and a variety of engineering, process control and operation tools. Soft FF will be the fuel that permits Fieldbus to lift even higher into the orbit of process industries.

Mario Hoernicke
ABB Corporate Research
Ladenburg, Germany
mario.hoernicke@de.abb.com

Phillip Weemes
ABB Process Automation,
Oil, Gas and Petrochemicals
Oslo, Norway
phillip.weemes@no.abb.com

Heino Hanking
ABB Automation GmbH
Cosult[IT] Solutions
Minden, Germany
heino.hanking@de.abb.com

References

[1] Shah, H.(2007). Automation Systems for Process Industries Worldwide Outlook. ARC Advisory Group.
[2] O'Brien, L., Hill, D. (2007-02). Foundation Fieldbus Provides Automation Infrastructure for Operational Excellence. ARC Advisory Group, White Paper.
[3] Fieldbus Foundation (2008). FOUNDATION Specification Function Block Application Process Part 1, Revision: FS 1.8.
[4] Fieldbus Foundation, (2008). FOUNDATION Specification Function Block Application Process Part 2, Revision: FS 1.8.
[5] Berge, J. (2004). Fieldbus for Process Control: Engineering, Operation and Maintenance, ISA.
[6] Heimberg, R. (October 2011). Die Digitale Fabrik erhält Einzug in die Prozessindustrie – visuelle Entscheidungssyste-me im Anlagenbau. Digital Plant Kongress, Würzburg.
[7] Gamma, E., Helm, R., Johnson, R., Vlissides, J. (1995). Design Patterns. Elements of Reusable Object-Oriented Software. Addison Wesley.