Transaction Safe Field Devices
Base of System Wide Data Consistency

Thorsten Szczepanski*, René Simon, Sören Scharf*****

** ifak system GmbH, Oststr. 18, 39114 Magdeburg, Germany, e-mail: tsz@ifak-system.com*
*** HS Harz, Friedrichstr. 57-59, 38855 Wernigerode, Germany, e-mail: rsimon@hs-harz.de*
**** ifak system GmbH, Oststr. 18, 39114 Magdeburg, Germany, e-mail: ssc@ifak-system.com*

Abstract: Parallel accesses to the data of a field device by more than one user, e.g. a programmable controller and a commissioning tool, may cause inconsistencies, which may result in a functional misbehavior of the overall (automation) system. The solution to this problem must start within the field device. It is proposed here to integrate a transaction manager directly into the field device. The transaction manager controls all internal and external accesses to the field device data and ensures the data consistency within the device. These transaction safe field devices are the necessary base for an (automation) system wide data consistency.

Keywords: Field Device / Multiple Users / Parallel Accesses / Data Consistency / Transaction Safety

1. STATE OF THE ART

Field devices are used within automation systems in order to measure and / or to manipulate process values. Field devices are connected with other field devices and higher levels of the automation system through communication interfaces. A field device consists of an internal data base, one or more process interfaces and optionally of human machine interfaces, internal signal processing and persistent memory. The interfaces, the signal processing and the persistent memory access the internal data base, i.e. they read and write its data. Other field device elements, not integrated from a data point of view, are e.g. the power supply, mechanical elements, etc.

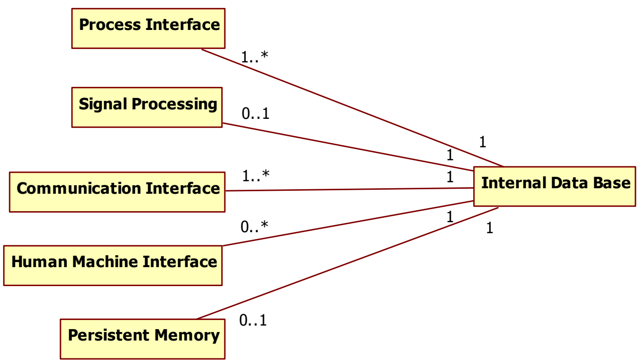

Fig. 1. Conventional field device.

The internal data base contains the process image with process variables, parameterization and configuration parameters and the running program code. The process interfaces measure process values and manipulate control variables. The signal processing, implemented e.g. by function blocks, processes the process values. The communication interfaces (field buses, industrial Ethernet) provides for the communication with other devices. A local display or switches may serve as human machine interfaces. One or more microprocessors realize the overall system field device (Fig. 1). Usually, field devices have very limited resources (memory, processing capacity) and cannot use a (real-time) operating system.

Field devices are used in all areas of automation, including process engineering (chemical, power plants) and manufacturing (car production). They include sensors (pressure, temperature, level, etc.) and actuators (valves, inverters, motors, etc.). The data and functions of field devices in automation systems are used by other field devices and other automation components. The latter are e.g. programmable logic controllers (PLCs), distributed control systems (DCSs), commissioning tools, maintenance tools, Enterprise Resource Planning (ERP) systems or Manufacturing Execution Systems (MES). Simon, R. (2002).

Some problems may occur while using field devices. These problems are caused by the interaction of both external and internal users. A user accesses the internal data base of the field device via the elements of the field device. Internal users are the described elements of the field device. External users are other automation components and other field devices, which are connected with the field device via communication interfaces. The automation system itself, persons operating it or the process to be automated may trigger the user accesses.

The problems may be classified as follows:

- inconsistencies through simultaneous reading / writing of data of the internal data base by more than one user, e.g.:

    - cyclic and acyclic communication
    - local display and communication interface
    - more than one communication (e.g. Ethernet and fieldbus)
    - any other combinations
- inconsistencies through reading / writing of logically related data of the internal data base by one user in several steps (acyclic in several telegrams or partly cyclic, partly acyclic), e.g.:
    - measured value and unit
    - lower and upper limit and unit
    - measured value and state
    - parameter set
- missing possibility to go back to an initial state during sequential execution of several write / read operations, when needed by the user, e.g.:
    - re-configurations
    - downloads (e.g. of several function blocks or of all parameters of a field device)
    - uploads (e.g. of all parameters of one or more field device)
    - downloads into several field devices
- missing consistency of the internal data base of a field device with:
    - data bases of other field devices
    - data bases of automation components (e.g. controller, DCS)

While solving practical automation tasks using field devices the above described problems may occur in any combination. These problems may result in a severe functional misbehavior of the overall (automation) system. Solving the described problems becomes more urgent because field devices are no longer used isolated or in 1:1 communication relationships, but are more and more supporting the communication with several users. This is further accelerated by their ongoing integration into higher levels of automation (e.g. for Asset Management).

2. TRANSACION SAFE FIELD DEVICE

Simultaneous and competing accesses of several users to the internal data may cause a number of errors such as lost changes, dirty reads, non-repeatable reads or phantoms. These problems are known in information science (implementation of data base management systems) and can be avoided by using transactions. A transaction consists of a series of accesses (read / write) to the internal data base, which are either all successful or all unsuccessful. A partial execution is not allowed. A transaction transfers the internal data base from one consistent state to another consistent state. A typical example in field devices is the necessary consistency between the value of a measured variable and its unit. A field device which is measuring a weight of 2 kg and providing it for further processing is not allowed to deliver (temporary) 2 g during a re-parameterization to the unit gram, but correctly 2000 g. Furthermore, also the processing (e.g. by the controller) must use not only the new value but also the new unit, i.e. it must also be "re-parameterized". Transactions may fail due to hardware faults, power failures, communication errors, programming errors or unrealizable conditions. A transaction must fulfill the requirements atomicity (only complete execution), consistency (of the internal data base after execution), isolation (from other transactions running in parallel) and durability (saving of the results). A transaction is managed by a transaction protocol. Geisler, F. (2006).

Therefore it is necessary to develop a field device for automation systems which prevents conflicts during parallel accesses to its internal data base. The basic solution must start within the field device because solely external measures are often intentionally or accidentally skirted. It is proposed here to integrate a transaction manager directly into the field device, which manages locks, transactions and / or users. The transaction manager controls all internal and external accesses to the internal data base of the field device using transactions (Fig. 2). A transaction ensures the consistency of the internal data base over a non-interruptible sequence of accesses to its data.

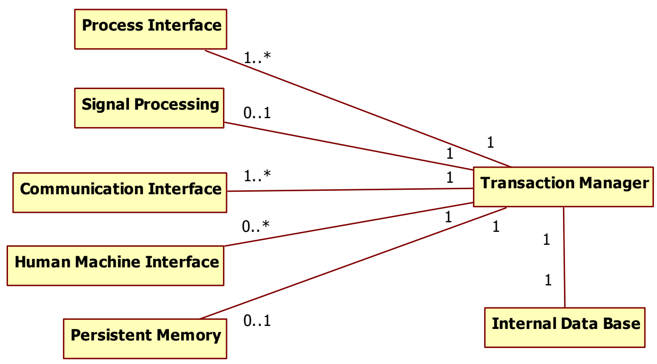

Fig. 2. Transaction safe field device.

A transaction manager provides several functions to manage users, locks and transactions. The user management includes e.g. the creation and deletion of users, the assignment of rights to each user and the assurance of the observance of these rights. The lock management includes e.g. the request and release of locks, the assignment of locks to data of the internal data base, the definition of locks (e.g. exclusive locks for reading and writing or shared locks for reading) and the assurance of the observance of the locks. The transaction management includes e.g. the start, the rollback (dismiss temporary changes), the commit (save of temporary changes), the assignment of locks, the access to data of the internal data base and the assurance of at least two-phase transactions. The implementation of these requirements must consider the special constraints in field devices (embedded, small footprint).