WP7 : Supervision System development Deliverable D 7.1 : Definition of the hierarchical structure of the supervision system

Jérôme Harmand, Djalel Mazouni, Paolo Ratini,
Cristina Verde Rodarte, Dieter Wimberger

26th January 2005

1 Introduction

This document presents the hierarchical structure proposed for the supervision system (called the Plant Operation system in the Deliverable D8.1) developed within the framework of the EOLI project. The supervision systems aims at detecting an "expected" but abnormal situation. In fact, we present hereafter two hierarchical structures of the systems developed since it was decided to validate the EOLI solution on two distinct pilot plants : one at the INRA-LBE, Narbonne, france (pilot plant of partner 2) and the other one at the UNAM, Mexico city, Mexico (pilot plant of partner 6). The idea behind the presentation of these specifications is that we demonstrate that the EOLI system is very generic : in Narbonne, the plant will be equipped with the SPES hardware and software while the plant at UNAM is equipped with a complete different hardware and software platform. Yet, the EOLI solution can be implemented in the same way given that the specifications given here below are followed. This is why only software specifications are given in the section 2 while both software and hardware considerations are treated in the section 3. This deliverable has to be read in connection with Deliverables D7.2, D7.3 and D8.1 in order to better understand the way the POS interacts with the other parts of the EOLI project.

The successful design and implementation of intelligent complex control systems is a grand challenge problem for process control engineers, especially with regard to the incorporation of process fault diagnosis (PFD) [VEN03c]. Industry and academic research alike are seeking patterns and designs that allow the creation of reliable and extensible complex control and supervision systems [HWV03, SAN00]. For the industry, these systems tend to consist more and more of software elements (60-80%) and requirements are:

- Interoperability and Integration
- Adaptability (e.g. cope with changes in the future)
- Ease of deployment

In academic research these requirements are seldom addressed, which sometimes leads to abandonment of successfully implemented prototypes for specific tasks [VEN03c]. Especially with regard to successful incorporation of process fault diagnosis and supervision, the integration of a system is a crucial factor: a completely integrated system allows a more complete access to the system as well as to combine various methods (hybrid approach) and thus leverage their varying capabilities with regard to detection and isolation of problems [VEN03a, VEN03b, VEN03c, DAS00].

The motivation of this work is to create an abstraction and patterns that allow the assembly of a reliable and adaptable complex control system, that incorporates PFD and FDI elements as well as supervision and decision support. Emphasis is put on properties like modularity and extensibility, with the final objective of creating a ”design for change” [SAN00].

The rest of the document is organized as follows.In the next section, the structure of the supervision system to be validated in Narbonne is presented. The section 3.1 presents the system design to be implemented at the UNAM and describes its elements. Section 3.2 presents a possible mapping of the SPES implementation for the EOLI project to the formerly presented abstract architecture. A glossary is appended to explain terms used throughout the document.

2 Detecting and classifying process faults : specifications for the LBE pilot plant

The global structure of the supervision system is given in figure 1.

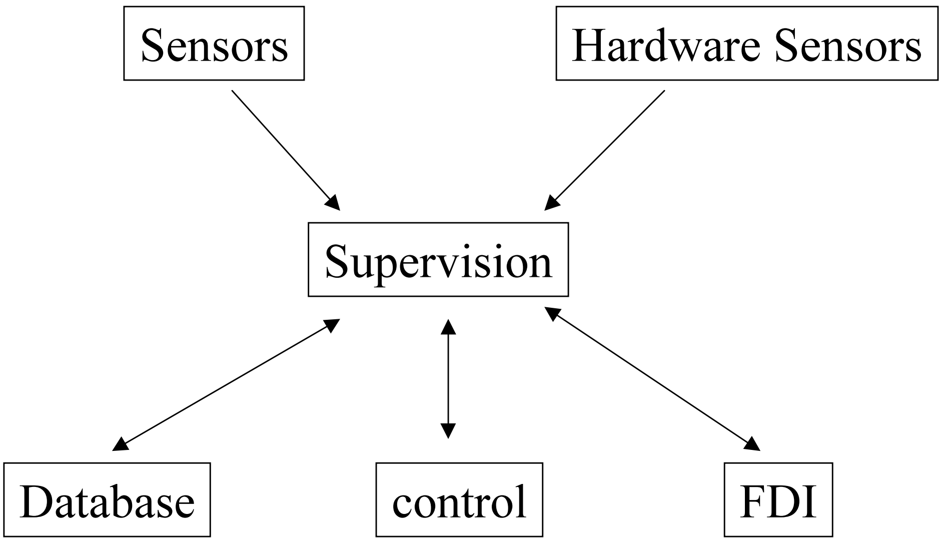

Figure 1: System Structure for the EOLI solution at the LBE

One of the most important point is the way the system accesses to the database of the process. For that purpose and in accordance with SPES, it was agreed to use the data exchange protocol and data format developed within the framework of another European project : the TELEMAC project.

In terms of supervision, we define hereabove two specific configurations to be delivered for the EOLI solution :

- *case 1* The first configuration is dedicated to SBR with poor instrumentation, that is with only a dissolved oxygen probe and/or with redox potential and pH.
- *case 2* The second configuration to be considered is related to an advanced configuration of sensors, as the one proposed within EOLI when UV and/or TITAAN hardware sensors (to have an estimation about the nitrogen) are present together with the classical probes (pH, DO and Redox).

For both configurations, the following supervision structure is developed :

- A model-based process fault detection and classification,
- A data-based process fault detection and classification.

Both solutions are running in parallel and are coupled to a tree-based reasoning system advising the operator in case of problem and/or advising him with respect with the most suitable sequence of operations to be realized. When the information needed to take a decision is not enough, the system may ask the operator to perform a number of off-line measurements while the database is used to compare the actual situation to determine if this situation has already been encountered in the past.

The algorithms developed are detailed in deliverable 7.3 : Knowledge based system able to recognize and classify a given "expected" but abnormal situation.

3 System Design and Elements for the UNAM pilot plant

3.1 System structure

Figure 2 presents an abstract scheme of the complete system architecture.

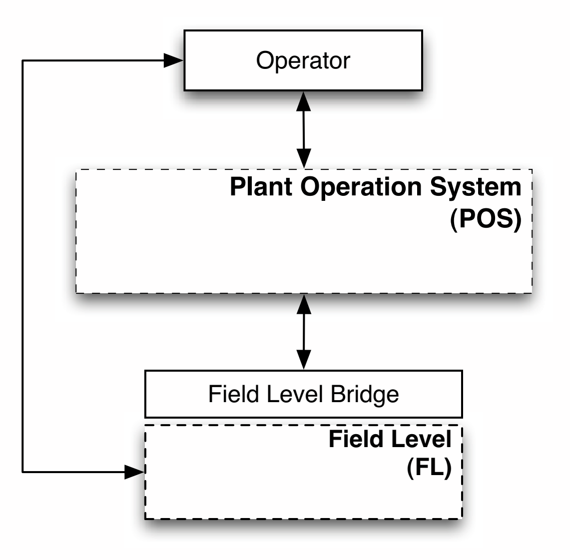

Figure 2: System Structure

The basic division is between the field level (FL) and the plant operation system (POS), which has been introduced because of following reasons:

- Avoid dependencies on a specific field level implementation
- Separation of concerns
- Provide a pattern that can be repeated and re-utilized

3.1.1 Field Level (FL)

The basic motivation for the design of the field level is to achieve total confidence that the process can be controlled at any time. It is expected that the implementation will make use of automatization hardware providing hard realtime[1], although for certain implementations this might not be a requirement. The design should be flexible enough to allow different choices regarding the requirements of the actual realization.

Figure 3 presents the structure of the field level design, which consists of:

- Basic Human-Machine Interface (HMI):
  see Glossary
- I/O System:
  Input/Output system capable of communicating with sensors and actuators. Examples are Distributed I/O, PLC with I/O Modules etc.

[1] a failure in the latency of the system consequently leads to a failure of the system.

- Actuators and Sensors:
  Sensors provide the online data of the process state and actuators allow to control the process by commanding actions.

- Basic Control:
  This element includes basic control loops and logic blocks that are required to maintain operation even if the communication with the higher level is interrupted.

- Low-Level FDI:
  This element includes FDI blocks for sensors, actuators as well as the low level control system. Nowadays sensor and transmitters are implemented with microprocessors enabling the design of intelligent or smart devices. An FDI block for a sensor should realize a diagnosis of the sensor state and it's measurements, providing information complementary to the actual measurement (data quality, abnormal sensor conditions) to a higher decision level. In case of actuators an FDI block should try to identify abnormal instrument conditions based on inherent characteristics and convey them as well to the higher decision level. Additionally low-level control FDI blocks could realize the evaluation of basic system alarms.

Certain building blocks might be omitted, in case they are not required for a specific implementation.

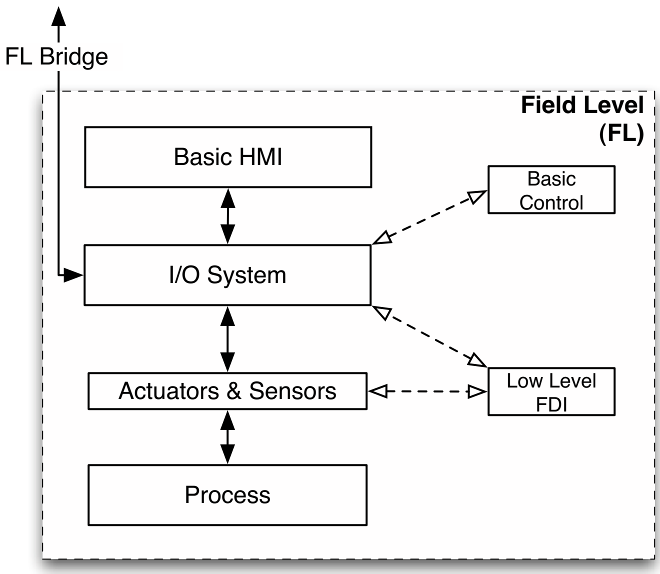

Figure 3: Field Level Structure

An industrial standard implementation of the FL design is demonstrated in Figure 4 (left). The core is a PLC with I/O modules interfacing with the basic HMI as well as the plant. The basic control is realized as a program on the PLC, the low-level FDI could as well be a PLC program or directly coupled with a sensor or actuator, providing extra inputs describing the state of a particular sensor or actuator.

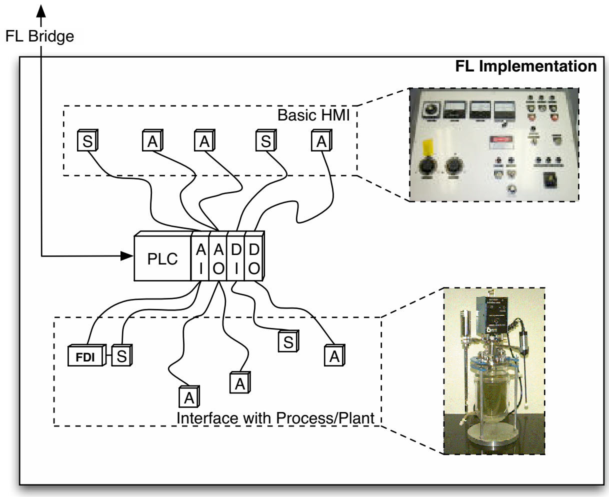

Figure 4: Field Level Implementation Example

3.1.2 Field Level Bridge

The field level bridge provides a bidirectional link between the field level and the POS in a modular way. The idea behind the design (see Figure 5) is to achieve decoupling and to create patterns for reusable bridge elements. Basically the bridge will be composed of three elements:

1. POS Infrastructure Communication:
   This element of the bridge is a POS infrastructure specific implementation, utilizing the infrastructure's mechanisms for communication with other POS elements.

2. Field Level Model:
   This element of the bridge should represent an abstract model of the field level environment instead of relying on raw data channels for sensors and actuators. An example would be a model of a tank with aeration that can be turned on or off, rather then just having a digital channel DO 04 which is "somehow" mapped to the aerator. Similarly the model should provide the temperature of the bulk media content, instead of having to work with an analog channel AI 01 that provides an integer value representing 12.3 mA and by transformation the actual temperature.

3. Field Level Communication:
   This element of the bridge provides a field level specific implementation of the communication with the I/O System. An example could be the implementation of a serial communication protocol over a RS232 cable (see 3.2), as well as Modbus or any other type of field bus communication protocol (which might even require special hardware). Well done, the abstraction should shield from low level details, so that the implementations for different protocols might easily be reused.

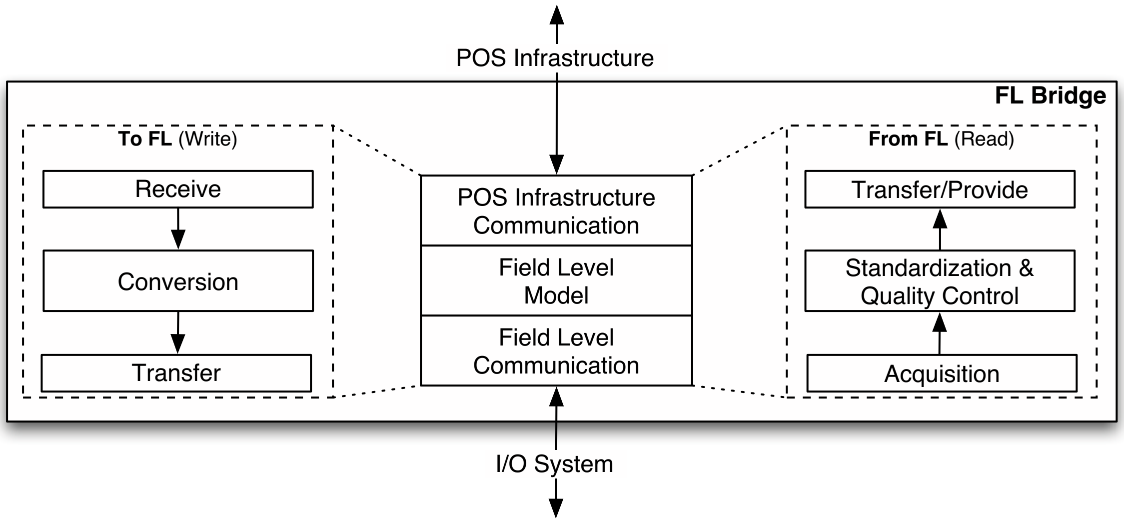

Figure 5: Bridge Structure

A possible implementation of this bridge could be realized on a small embedded system node (distributed system implementation), as well as it could be directly integrated into a single node.

3.1.3 Plant Operation System (POS)

The basic operation of a plant/process is a principal requirement, but today it is much desired to have an automatic system that:

- tries to maintain the process within conditions defined by design specifications
- tries to detect and identify failures (i.e. when the process is no longer within design conditions)
- tries to help the operator to correct an abnormal condition or to correct it automatically through reconfiguration
- provides data and information (actual as well as historical) about the process to the operator

The elements that provide this functionality are basically the different functional aspects of a complex control system:

1. Human-Machine Interface:
   see Glossary
2. Supervision and Decision Support:
   This element should use the information available from a fault diagnosis system to check and monitor the process under regulatory process control, with the objective of assisting a human operator to manage the process [VEN03c] on a time-frame smaller than planning.

The idea is to enable the system to cope with natural variability in processes (due to raw material variability and unsteady environmental condition), that often make the closed world assumption for regulatory control strategies invalid.

3. Advanced Control:
   This element includes control blocks that are leveraging resource intensive methods (heuristic and/or capable of adaptation or learning).

4. Fault Detection and Isolation (FDI) and Process Fault Diagnosis (PFD):
   This element should provide a diagnosis of the process from a higher level perspective. It's task is to evaluate when the process leaves the conditions defined by design specifications and provide the Supervision and Decision Support with information that identifies and describes an abnormal condition.

The basic motivation for the design of the POS architecture is to achieve integration between all these elements in a way that allows the system to evolve over time ("design for change" [SAN00]). Basic design goals are thus modularity and extensibility. Although it is not required or necessary to implement the POS as a distributed system, it would probably be the best solution to ensure the flexibility for future changes. No assumption is made about the system latency, basically it will depend on the specific realization (e.g. hard or soft realtime, as well as "best effort"). Figure 6 presents the structure of the POS design.

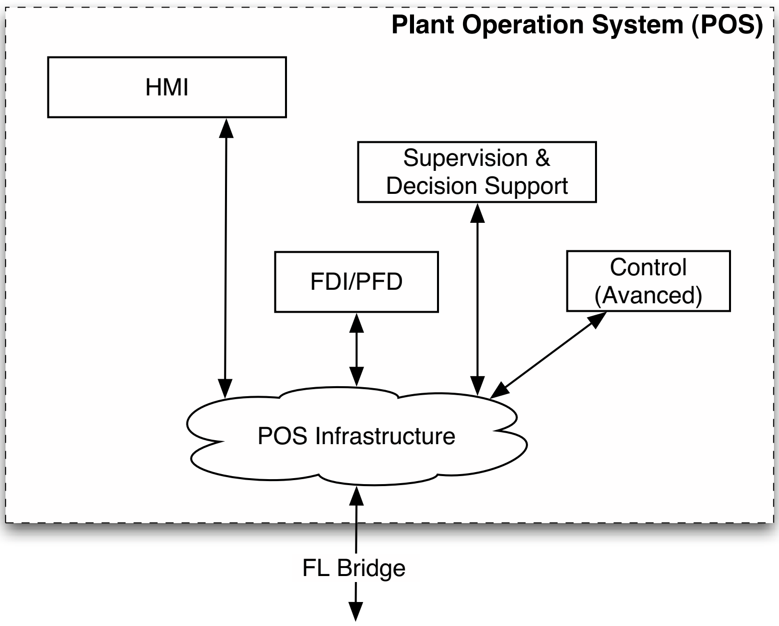

Figure 6: POS Structure

The POS infrastructure is the part of the POS that provides:

- Communication between elements:
  The details of network or interprocess communication should be hidden from the actual component implementation and interoperability between heterogeneous components and platforms should be possible.

- Dynamic reconfiguration and extensibility mechanisms

- Basic services like for example data storage

3.2 Example: SPES Implementation for the EOLI Project

3.2.1 Field Level and Bridge

Figure 7 (right) shows how the SPES proposal for the EOLI project maps to the field level abstraction. The I/O System is realized as embedded master board with RF links to actuators and sensors (each with it's own slave board). A simple PLC connected via an RS232 serial link provides basic control.

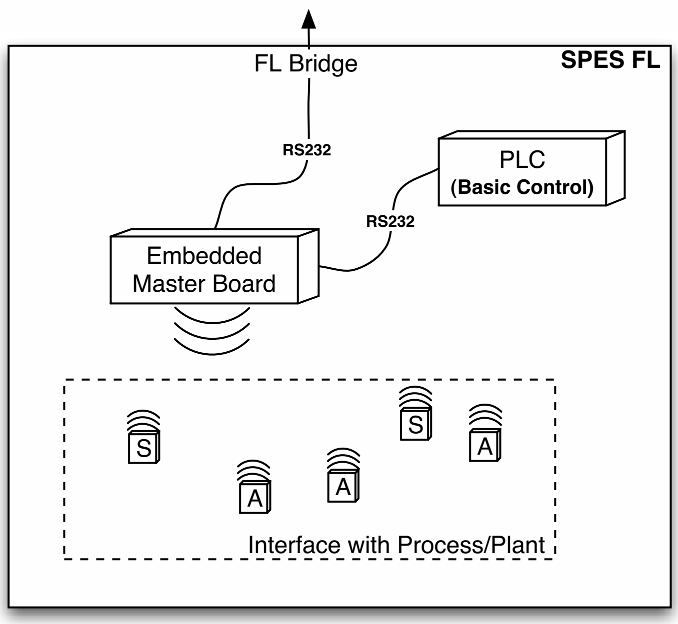

Figure 7: SPES Field Level

Field level communication is realized via another RS232 serial link between the host PC and the embedded master board using a proprietary protocol. Some software component running on the PC has to provide the FL bridge functionality. Although a model has not yet been proposed, it is very likely that there will be some kind of mapping or interface that helps the various components to read from/write to specific field level sensors and actuators.

3.2.2 POS

The SPES implementation can also be mapped to the POS abstraction, although everything will be running on a single node. This means that the system has certain natural limitations with regards to the proposed "design for change" (i.e. properties like extensibility and dynamic reconfiguration). However, the POS infrastructure is reflected by the following elements:

1. Communication between components:
   Windows Messages and global variables

2. Data Storage:
   Database interface and a MySQL RDBMS

Control, FDI and Supervision are to be realized as compiled Matlab modules running within a VC++ host application, that will provide the HMI (GUI) and FL bridge at the same time.

A Glossary

Bus-Coupler Processing unit of a modular Distributed I/O system, capable of handling the communication with various I/O modules as well as with the other nodes on the network.

Distributed-I/O An I/O system capable of communicating with sensors and actors, as well as with other nodes on a network, allowing them to remotely read inputs or write outputs.

HMI The human-machine interface (HMI) is where people and technology meet. In can be as simple as a button on a control board, or nowadays as complex as an operator console application. However, in any case it should be well designed and fit the user's mental map of the task he or she wishes to carry out. [IEC03]

I/O Input/Output

I/O-System Input/Output system capable of communicating with sensors and actuators. Examples are Distributed I/O, PLC with I/O Modules etc.

PLC Programmable Logic Control

Remote-I/O see Distributed I/O

References

[VEN03c] Venkatasubramanian V. et.al.: "A review of process fault detection and diagnosis - Part III: Process history based methods", Computers and Chemical Engineering, pp. 327-346, (27) 2003.

[SAN00] Sanz T. et. al.: "Software for Complex Controllers", in "Control for Complex Systems", Springer Verlag, pp.143-164, 2000.

[HWV03] Heck, Bonnie S., Wills, Linda M., Vachtsevanos, George J.: ”Software Technology for Implementing Reusable, Distributed Control Systems”, IEEE Control Systems Magazine, pp.21-35, Feburary 2003.

[VEN03a] V. Venkatasubramanian, R. Rengaswamy, K. Yin, and S. N. Kavuri, "A review of process fault detection and diagnosis: Part I: Quantitative model-based methods," Computers & Chemical Engineering, vol. 27, pp. 293-311, 2003.

[VEN03b] V. Venkatasubramanian, R. Rengaswamy, and S. N. Kavuri, "A review of process fault detection and diagnosis: Part II: Qualitative models and search strategies," Computers & Chemical Engineering, vol. 27, pp. 313-326, 2003.

[DAS00] S. Dash and V. Venkatasubramanian, "Challenges in the industrial applications of fault diagnostic systems," Computers & Chemical Engineering, vol. 24, pp. 785-791, 2000.

[IEC03] International Engineering Consortium (IEC): ”The Human-Machine Interface (HMI) Tutorial”, 2003, 28 Nov. 2003 <http://www.iec.org/online/tutorials/hmi>.