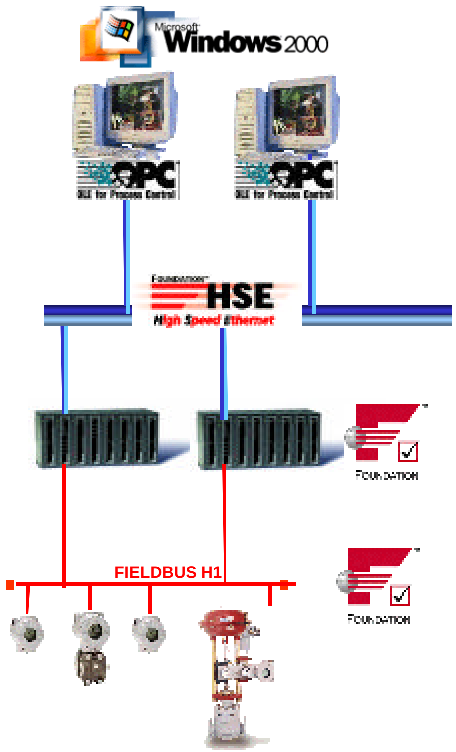
Architecture Description

Applications Engineering Department
International Division

This report briefly presents the key components of Smar System302, which can be found in the typical architecture illustrated in the next figure.



Operation Workstation:
ProcessView
Engineering Workstation:
Syscon (FF Function Blocks Language)
Conf700 (IEC 61131-3 Ladder Logic Configuration)
Maintenance Workstation:
AssetView

Fieldbus HSE Devices
ESD Integration
Package Unit Integration
3rd Control System Integration

DFI302
Linking Device (H1 to HSE)
Link Active Scheduler
FF Discrete I/O Multiplexer
FF Controller
FF Bridge

Fieldbus H1 Devices
Modbus RTU Devices
Modbus TCP/IP Devices
4 – 20 ma Devices
Hart Devices
Discrete I/O

Open System Architecture

Unlimited Scalibility

Power Supply Subsystem

Configuration Languages

Link Active scheduler

Field Devices

Smar will supply field devices with the following features:

- Function block instantiation allowing the selection of up to 20 Function blocks per device, the function blocks type are the following:

  RES – Resource Block
  AI - Analog Input
  PID – PID Control
  AO - Analog Output
  ARTH – Arithmetic
  INTG - Integrator
  ISEL - Input Signal Selector
  CHAR – Characterization
  SPLT - Splitter
  AALM - Analog Alarm
  MAO - Multiple Analog Output
  MDO - Multiple Discrete Output
  MAI - Multiple Analog Input
  MDI - Multiple Discrete Input
  SPG - Setpoint Generator
  TIME – Timer
  LLAG – Lead Lag
  OSDL – Output Signal Selector and Dynamic Limiter
  CT – Constant Block
  DIAG – Diagnostic Transducer Block
  APID – Advanced PID
  DENS – Density Block

- Current consumption

All the devices have a current consumption of 12 mA for transmitters as well as valve positioners minimizing voltage drop and allowing long cable distances. Another benefit of this lower consumption is to allow up to 8 devices in the same barrier when Fieldbus Intrinsic Safety Concept (FISCO) model is used.

- LAS - Link Active Scheduler

All field devices supplied by Smar have link active scheduler capability, which adds a new level of redundancy to the system, since even a in fault situation of the linking devices, the control will continue without any disturb.

- Function Block Execution Time

The function block execution time is informed in the following table.

	FB Type	Block Time(ms)
1	RS	3
2	AO	18
3	AI	22
4	PID	43
5	ARTH	47
6	SPLT	21
7	CHAR	23
8	INTG	26
9	AALM	23
10	ISEL	13
11	SPG	20
12	TIME	13
13	LLAG	22
14	MDI	6
15	MDO	10
16	MAI	6
17	MAO	10
18	OSDL	23
19	DENS	64
20	APID	64
21	DIAG	
22	EPID	43
23	CT	9

HSE – Redundant Linking Devices

Linking Device

A linking device provides the link between the H1 field level networks and the HSE host level network.

Linking Device Redundancy

Hot-standby redundancy is used at several levels, including redundant Fieldbus interfaces. It provides two complete independent communication paths between a Fieldbus H1 network and the workstations ensuring that plant floor data reaches the operation stations even if one interface fails, always providing a window to the process. This way, a single module failure does not prevent the operator from seeing the process. Once again, diagnostics play the important role of fault detection. It enables not only the execution of the bumpless switchover from primary to secondary communication scheduler, but also the warning to the operators of any fault so that the parts can be immediately replaced in order to be ready to enter in service when required.

Controller Redundancy

Although control distributed to the field instruments is one of the keys to high availability, control may also be done centrally in any of the linking devices. The control functions may also be redundant with the two controllers executing identical control strategies. Function blocks executed in one controller are synchronised with those in the other, ensuring a bumpless transition in case of failure.

Separation

A key to fault tolerance is that the primary and backup linking devices must not share backplane or interface modules. They are physically separated with individual processor and power supply to eliminate common causes of problems. If mounted in separated panels, they will not be subjected to the same stress, such as radio interference or power surges. On the other hand, if controllers were mounted next to each other on the same backplane or in the same panel, the cause for the failure of one would most likely affect the second too.

Host Level Network

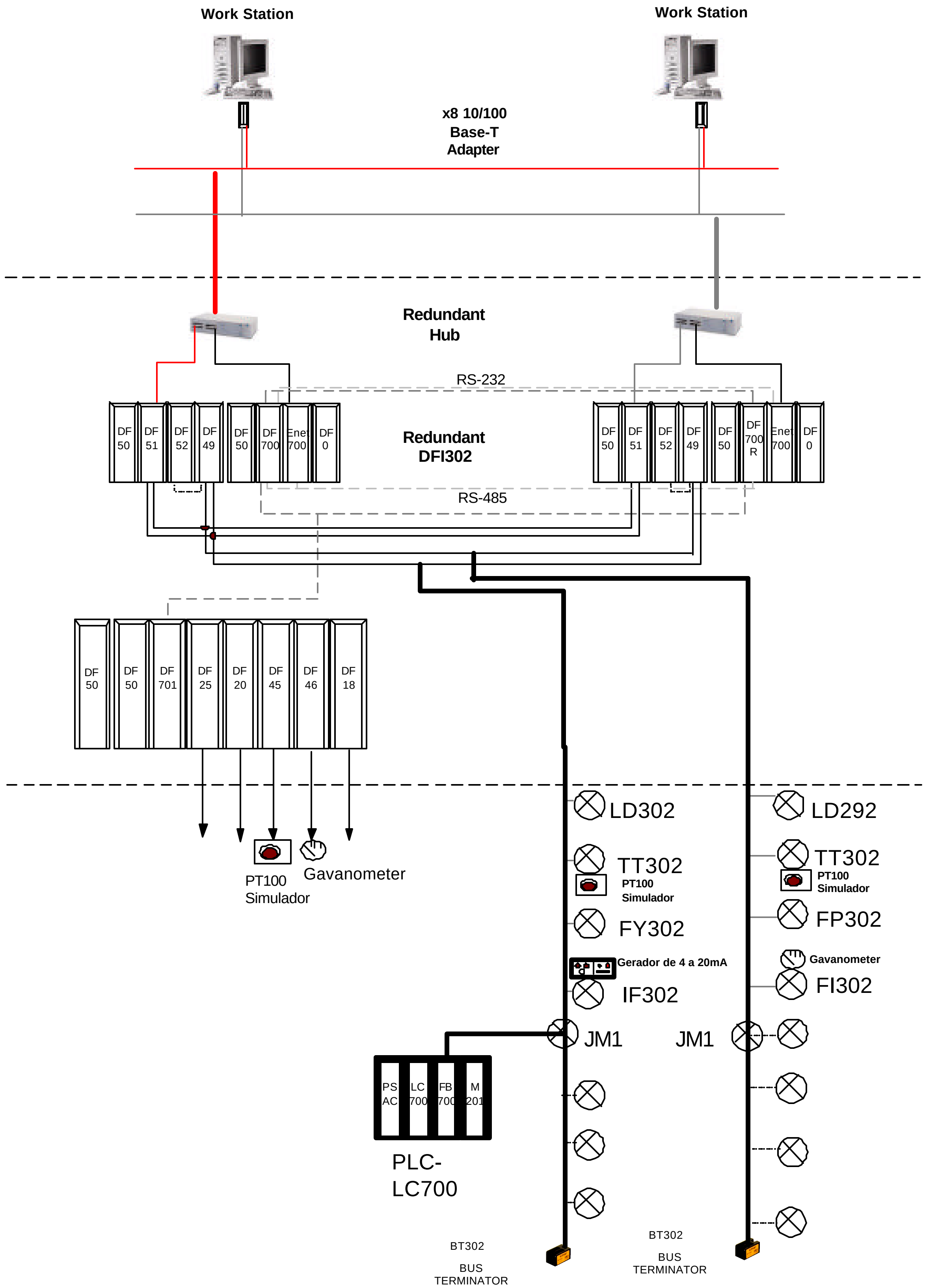
The operator's ability to see the entire plant relies on the host level network. This is why it has to be redundant and consequently fault tolerant. The HSE wiring including hubs has dual redundancy for high availability. In case the primary network fails, the secondary is automatically used eliminating single points of failure. Industrial network hubs with redundant power supply are available. Additional availability can be achieved using a fibre optic ring topology.

As switches and hubs are used, there is only one Ethernet device per segment, limiting the extent of impact in case of wire damage. All redundant Ethernet device pairs and the workstations are connected to both Ethernet buses. The switchover is totally bumpless and transparent.

Redundant Workstations

A system may have several operator workstations where operation, engineering and maintenance can be carried out from the same or independent stations. The workstations are connected to the redundant host level network.
A system typically has two or more workstations. Should any workstation fail, the others are capable of operating the whole system. Workstations may be fitted with dual network cards for redundant communication, and multiple hard disks for redundant data storage, as well as UPS for the power.

Smar will show all the capability described above in action during the Tehran exhibition using the following system architecture.


Work Station
Work Station
x8 10/100
Base-T
Adapter
Redundant
Hub
RS-232
DF 50
DF 51
DF 52
DF 49
DF 50
DF 700
Enet 700
DF 0
Redundant
DFI302
DF 50
DF 51
DF 52
DF 49
DF 50
DF 700 R
Enet 700
DF 0
RS-485
DF 50
DF 50
DF 701
DF 25
DF 20
DF 45
DF 46
DF 18
PT100
Simulador
Gavanometer
LD302
TT302
PT100
Simulador
FY302
Gerador de 4 a 20mA
IF302
JM1
LD292
TT302
PT100
Simulador
FP302
Gavanometer
FI302
JM1
PS AC
LC 700
FB 700
M 201
PLC-
LC700
BT302
BUS
TERMINATOR
BT302
BUS
TERMINATOR

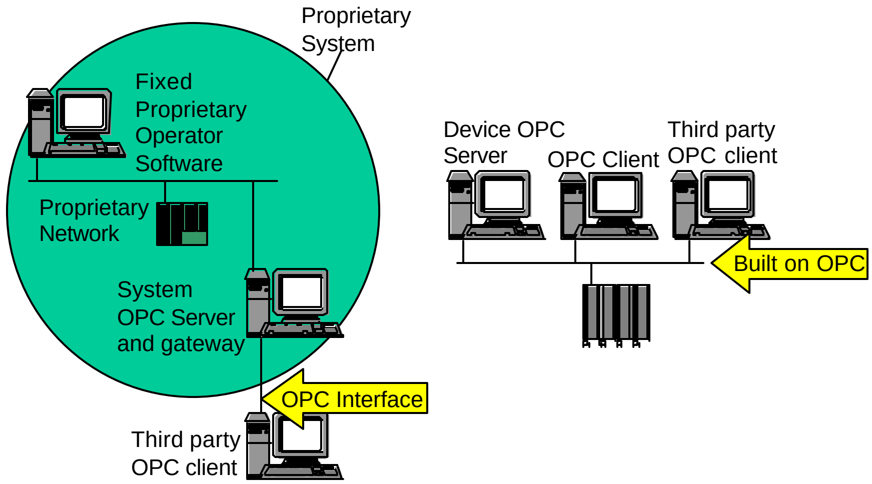
OPC

Until now all control system packages from every system supplier have their own proprietary process visualisation software. Typically this software was written for use only with the one set of hardware. In some cases off-the-shelf software have been customised particularly for one system, losing its flexibility. Actually, it was done that way for a good reason: previously, it was the only way a single integrated database could be achieved. I.e. in order to ensure that once a tag had been configured in one application it also appeared in the others, the software could only work with the single hardware it was dedicated for. Adding in a third party software was either completely impossible or could only be achieved with great expense in both time and money. Again the system supplier was in control of exactly which third party software they would allow. Many times safety reasons were given for not integrating a competitor's product, no matter how reputable the supplier was. Then, users started to demand software that is open yet integrated with hardware like before. Early attempts to pass data between open software applications included use of API calls that requires programming expertise, lots of documentation, and lengthy interactions between parties were involved. Any software revision often means changes. DDE was another attempt but since the specification is not stringent, the format of the data varied from one supplier to the other.

The solution for open real time data access was OPC (OLE for Process Control) client-server technology which was specifically developed for process control. Using OPC software applications can exchange data without custom programming. OPC has several added benefits e.g. it works not only within a computer, but also across networks enabling data to be disseminated throughout the enterprise. OPC also provides a single integrated database where any client application can access data from any server simply by pointing and clicking in an OPC browser.

OPC not only makes it possible to link any two software applications, but also to freely select hardware and display the data in the process visualisation software. OPC greatly simplifies all enterprise IT integration, making it easy to use data for ERP and asset management etc.

OPC technology is no guarantee that the system software will be open. Users should look at how the system implements OPC. Essentially there are two philosophies to which manufacturers to this. One OPC solution is just the same old proprietary hardware and software to which an additional gateway computer with an OPC server has been added. Third party applications can access data from a separate network via the gateway computer. This solution has the drawback that hardware cannot be accessed directly, perhaps limiting some information, and that the gateway computer is a weak link. The system is not built on OPC, it only interfaces to it.

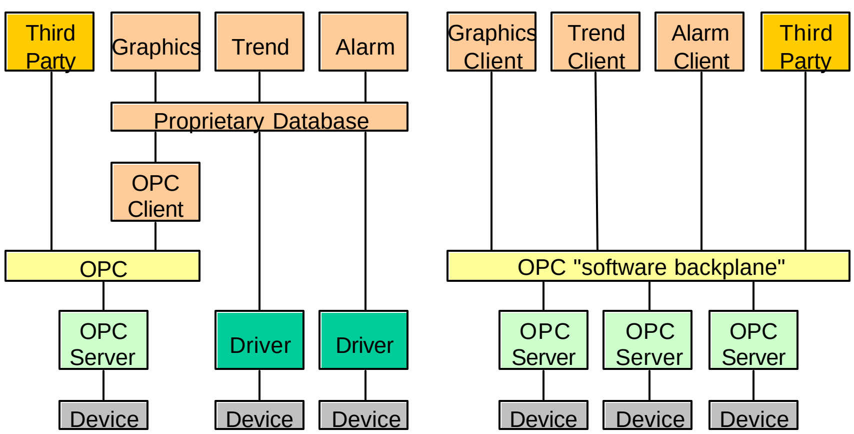


Gateway appended to proprietary system　　Open system built with OPC core

SYSTEM302 (right) is built with OPC native

A better, fully open, way to use OPC is for third party applications to access data as freely as native system applications. SYSTEM302 is built on OPC using this philosophy.

In addition to this, there are even significant differences in how OPC clients treat the OPC data internally. Some software has OPC but the data must still be mapped into and conditioned in a proprietary configuration database, just like for dedicated drivers. For modern plant that accesses many pieces of information from every instrument, the configuration task becomes daunting. The system will not be well integrated as any third party applications are accessing data in a different point than the native applications.



OPC as a driver to a proprietary database　　OPC to the core

The best implementation of OPC is like a "software backplane" where applications just plug in.

For SYSTEM302 any process visualisation software can be chosen, but primarily clients that use OPC to the core are used. All applications plug into the same software backplane, ensuring that all applications see the same data, guaranteeing consistency across the system.

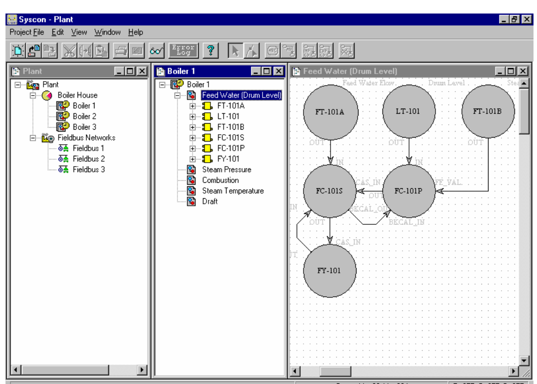
SYSTEM302 is the only packaged control system for which the user can select any OPC client, including the primary process visualisation software. Benefit from the rapidly growing number of OPC client-server applications. In spite of this exceptional degree of openness, the user gets a DCS-style single integrated database, i.e. once a tag is configured it is available everywhere, eliminating retyping, mapping and mistakes.

The OPC and FOUNDATION™ Fieldbus technologies have lots in common. Eventually all field data goes into software of some type, OPC is therefore required to benefit fully from H1 and HSE.

FF Configuration and Programming Tools a Homogenous environment

Until now every DCS manufacturer used their proprietary programming language to build the control strategy. For large plants with systems from different suppliers engineers were forced to learn many languages or had to rely on the supplier at high cost to make modifications to the system. Information and training was only available from the supplier. Now FOUNDATION™ Fieldbus blocks and IEC 61131-3 languages are taking the place of proprietary languages. Engineers can move from one system to another with little or no retraining. Training and information is available from multiple sources. Control systems changing to use the FOUNDATION™ Fieldbus function blocks and IEC 61131-3 languages instead of their own proprietary languages will be one of the major changes affecting control systems over the next few years.

The programming language is the main distinguishing feature of FOUNDATION™ Fieldbus compared to other networks. It is this function block diagram programming language that makes control in the field devices possible, and all the subsequent savings. Since the introduction of H1 Fieldbus a few years ago, thousands of loops have been successfully implemented in the field. The H1 blocks are suited primarily for continuous measurement and regulatory control.



The FOUNDATION ä Fieldbus programming language, a single configurator for multivendors devices.

For instrumentation users who spend too much effort and resources on maintenance, and see too much variability in product quality, instrument management and a proactive maintenance scheme can significantly reduce the cost of ownership for field devices. For this reason a modern control system needs more than just configuration and monitoring software. To manage the maintenance aspects of the control system a modern plant need a system that provides the traditional configuration and monitoring functions, plus rich new functions for field device calibration, diagnostics, identification, materials of construction and setup. Maintenance procedures can be improved through the use of software by leveraging the additional information available in Fieldbus devices. Maintenance suing software is getting ever more important as a new generation of engineers tend to be familiar with computers, not multimeters.

In a proactive maintenance scheme service is performed only on the instruments that need service, and only just before they are about to fail. Resources are not wasted on devices that need no maintenance. Many Fieldbus devices are intelligent enough to diagnose their own faults and many also monitor "leading indicators" of faults meaning additional measurements like ambient conditions and number of operations. This scheme requires a minimum of resources, has the lowest possible cost, and most efficient use of manpower.

The Fieldbus engineering and maintenance software allows you to tap into the information of your instruments and tell you the status of any device at any time, so that you can have a complete overview to picture the status of the entire plant. It is therefore easy to detect a plant situation where several devices may be relatively close to be in need of service. All of these device can then be serviced in one go, requiring only a single stop.