15th Triennial World Congress, Barcelona, Spain

USER INTERFACES FOR DISTRIBUTED CONTROL SYSTEMS IN CHEMICAL PLANTS - REQUIREMENTS AND SOFTWARE SOLUTIONS

Bodo Peters

ZT ENG PLT (DCS-Planning), Bayer AG
47812 Krefeld-Uerdingen, Germany

Abstract: Distributed control systems, process logical controllers with Supervisory and Data Acquisition Systems are used to control the production process. They uses graphic user interfaces to visualise piping and instrumentation of the plant. They are supported by other systems like simulators, quality management systems and expert systems. These systems give additional information, which help supervising the production process and configuring the automation systems. But it is very difficult to integrate this information into the user interfaces of the automation systems. To overpower this restraints a new interface is needed. For building up this interface the implementation of a common functionbase is required. On one hand this must conform with the information duty of the user and on the other hand it must adduce all system tasks. Modern information techniques like object modelling are used to specify the postulated functionbase. Described in this way, a functionbase must be implemented in common computer systems. *Copyright © 2002 IFAC*

Keywords: User Interface System, Human Machine Interface, Distributed Control Systems, Process Industry

1. Introduction

In the chemical and petrochemical industry all new plants are using Distributed Control Systems (DCS) or Process Logic Controllers (PLC) with Supervisory and Data Acquisition Systems (SCADA-Systems). All these systems are build to visualise the process by showing graphics similar to the piping & instrumentation drawings. In the pictures the measurement points are animated to show the actual process value. The design of the graphic is close to the equipment used in the control panel.
The human machine interface that is build in this way has a lot of restrictions. It is focused on a plant and signal oriented view on the process. Additional information coming from 3rd party systems like simulators, expert systems and management systems can not be integrated into those interface systems. Only a unified user interface in the control room will reduce the work load of the operator and empower him to operate the plant in a good operation point.
An other restriction of the user interface systems used today is that they are unsuitable for online changes, reengineering and documentation of their configuration. These features are required by the process industry because their plants are automated and run different then a manufacturing machine. Most process plants are maintained, enhanced and repaired without shut down. Only small parts of the plant for example a process unit is stopped. The production is still going on using a parallel unit. This units are so small that they have no separate control system. Fore example 10 distillation units are run by 3 controllers of a DCS. The process data is stored in 2 server which also drive 4 operator stations.
Most functions of the controllers are online changeable. Parameters, function blocks and connections can be changed, created and deleted in runtime mode. Only the user interface has to be shut down and restart with new designed interfaces. This is only possible because we are running 2 servers. One can be stopped and reload during the other is still running the old version with to operator stations. This situation is critical because there is only the half power to visualise and interact with the process. To reduce the times this is to be done a good design process with integration of the user was set up in our planning group.
But a real improvement of the user interface system is needed. So only a new model can meet the requirements and involve the user into a continuos development and improvement process of the plant automation.

2. Interface Design in the Daily Work

The design of the graphical user interface is aligned on a plant and measurement point oriented view.

Therefore all graphical objects and configurations are optimised for displaying these kind of presentation.
The most of the staff is used to the stand-alone equipment in the control panel. This practical experience is the starting point of the screen design.
The following steps are made to get a usable interface:

1. A design suggestion is made by a experienced worker of the production, normally a shift leader or master crafts man.
2. His suggestion is developed to a realisable design by the DCS-planning engineer working closely together with the crafts man.
3. The user interface design is realised with the DCS.
4. A screenshot is discussed with the master crafts man and the operators.
5. Last improvements mostly details are build into the user interface design.

The interface designed in this way is now to be used to run the plant unit. A few days after starting the first experience with the new plant unit are made. This could sometimes lead to changes in the user interface or in the hole way of controlling one unit.
At this point the possibility to do online changes on the user interface design would reduce time and costs to optimise the user interface design.

3. Requirements for User Interface Systems

To realise presentations and interactions in this flexible way a proper model is to be designed. It has to be used with the soft- and hardware that are state of the art in computer science. Additional to this the model must support all the features need by automation systems. A good overview about the actual development is given by the great trade fairs like ISA-show or Interkama [Nirschl(1996)], [Brucker, Tauchnitz, Eul(1996)], [Fromme et al(2000)]. Analysis of the new products can be found in some professional journals. The required system technology is [Birkhofer(1999)]:

- Stable continuos operation of the system
- Restart into operable on-state after a power failure
- Based on approved technology – no exotic operation systems or hardware
- Possibility to be integrated into the computer network of the plant with advanced security mechanisms

The graphical user interface system requires beyond it the following:

- Open Source – minimum is that the model is free available.
- Extensibility – upcoming changes must be integrated without loss of stability
- Configuration and set-up must be efficient and error minimised
- Automatic documentation and re-engineering must be part of the system itself

The last part of requirements is only necessary in process industry:

- Systeminterface to PLC and DCS
- Online changeable
- Object oriented like a function block system

These requirements are not meet by any system on the market. To design a system to meet them a detailed analysis of the different use cases is necessary.

4. Use Cases

A lot of use cases were analysed to design the model for a new user interface system. The use cases were separated into three kinds.

- To observe and operate the production process with “classical” and advanced designed interfaces [Ingendahl(1998)], [Brodlie(1992)].
- Engineering of the DCS especially at run time
- Integration of 3rd party date for example process design data [Heim(1996)], [Aspen(2000)], [OSI(2000)]

For all interfaces two topics are to be discussed. On the one hand the behaviour during the design phase and on the other hand the presentation and interaction possibilities during the run time. The screens for the plant operator are the kind of interface which has the highest needs in both topics.
One use case of the field of observation and operation is given below.
The use case describes the information needed to configure a presentation object to display a value on a screen. This value can be a measurement from the process or any other analogue value. Therefore the following information must be retrieved:

- What is to be shown (What kind of information)?
- Where is the data-source (address)?
- Is there a unit or a scale?
- Is beside the numeration a name or text necessary?
- What kind of presentation is available (graphic object library)?
- What size and position of the presentation object on the screen or window is needed?

Beyond it could be necessary to answer these additional questions:

- What is the update period of the process data?
- How is the state of the process data displayed?
- Has the presentation object get an explicit name (identifier)?

Many use cases where made. One conclusion of them is, that all the different questions and information can be sorted. A object oriented modelling of them is to be done next.

5. A Model to Meet the Requirements

After the analysis of the use cases a new model must be designed regarding to the general requirements listed in section 3.
The model is based on generic classes. They build the base from which all other classes can be derived. To build the classes on which the objects depend on the technique of composition is also used.
To describe the details of the model we will start with the general idea of separation and composition of functions and information to build the "inner" model, the "Two Block Model" [Peters(2000)].
The idea of the tow block model is that all the functions and parameters needed to build the presentation of process data can be separated into to blocks.

1. The Presentation-Block (PRE) covers all functions and parameters needed for drawing, displaying and acknowledge of user inputs.
2. The Data-Acquisition-Block (DAB) covers all functions and parameters to retrieve data, prepare them and transform the user inputs into system calls.

The two blocks must be linked and able to exchange data. A first idea of objects that can do it is given in the following figure:

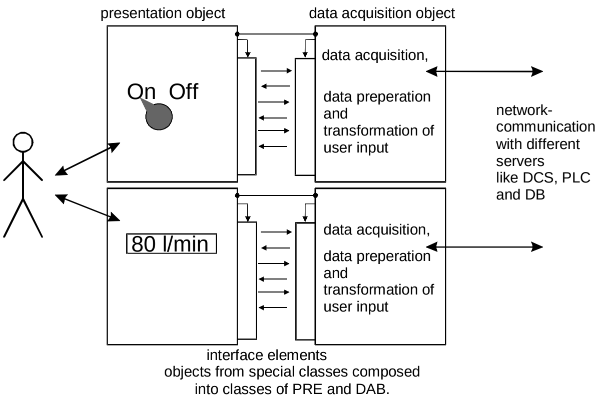

Fig. 1: PRE-DAB-Model

The two objects are linked or connected to do their duty. For data exchange they use objects called interface elements. These objects are composed into the classes of Presentation- and Data-Acquisition-Blocks.
To display a single value two blocks are needed. That seems to be expansive but the blocks can designed and programmed in a way resuming several values.

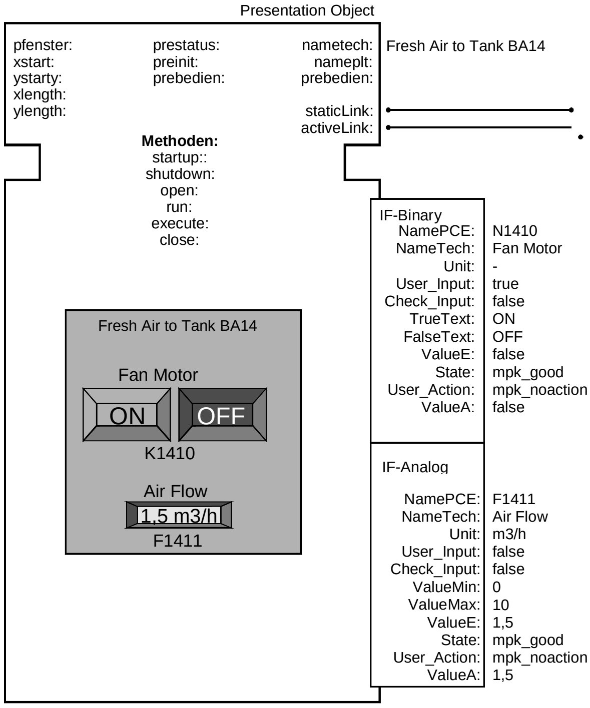

Fig. 2: Example of an Presentation Object

An other benefit is that one DAB-Object can drive several Presentations. So if you want to display the same value in different windows you only have to link all PRE-Objects to one DAB-Object.

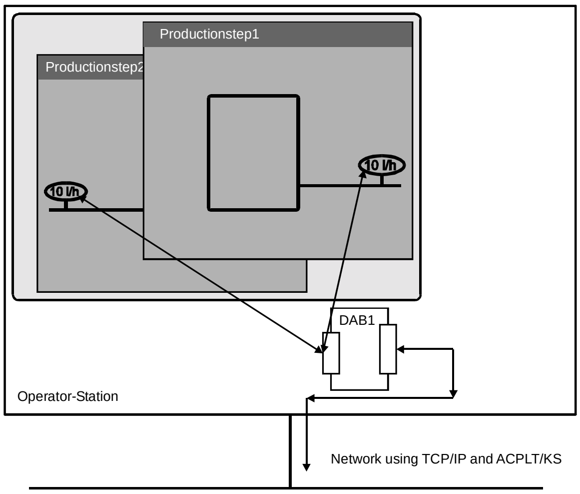

Fig. 3: One DAB powers two PRE's

The Data-Acquisition-Objects which can drive the shown Presentation-Objects are composed by using a derived class of the abstract Dab-Base-Class and objects of the interface elements and of communication elements. The programming of the overload functions of the new DAB-Class is supported by design pattern. A easy way to program functions for transforming user input into system calls and data processing functions to generate useful information out of the raw measured data.
The UML-Notification of the Two-Block-Model is shown in fig. 4.

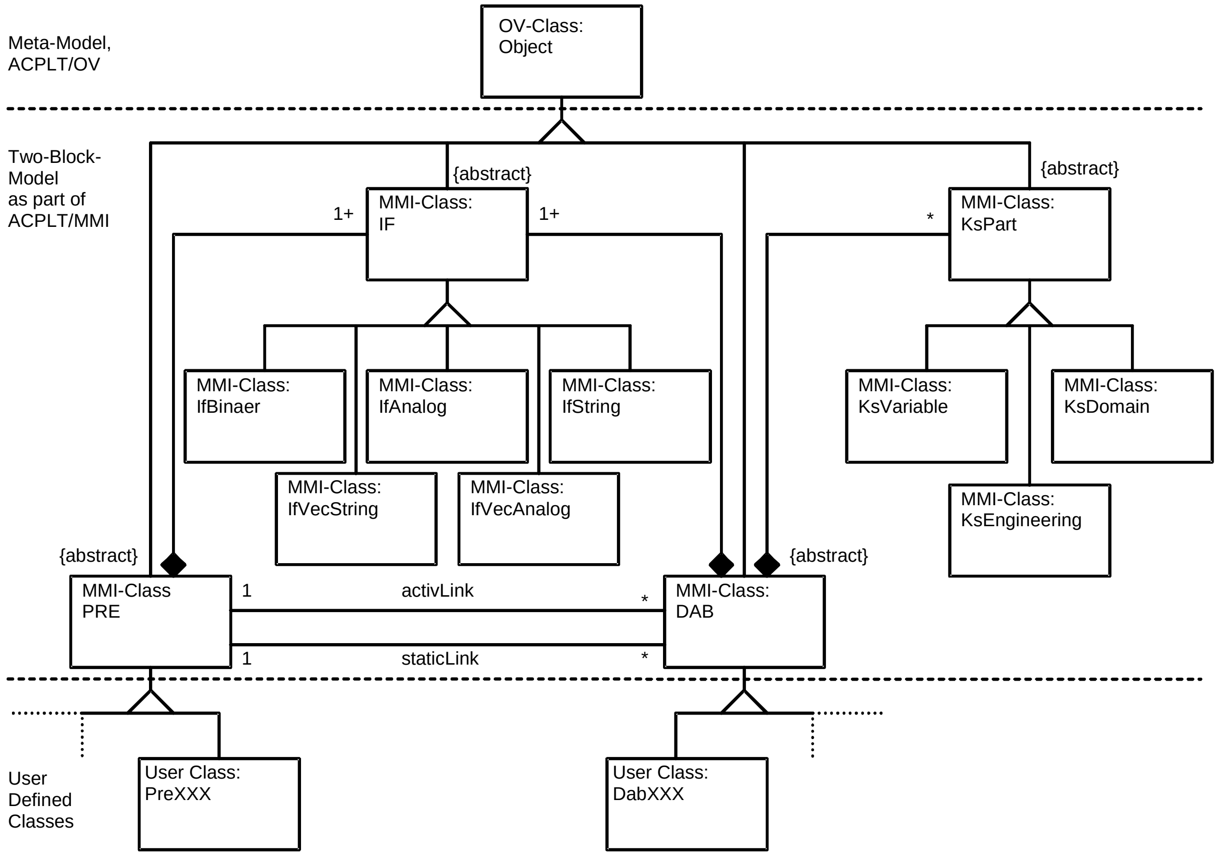

Fig. 4: The Two-Block-Model (UML-notation)

The root for all classes is the Class “Object” of the Object Management System. This system is based on a model build to describe all objects used in process control engineering. The German abbreviation ACPLT/OV is build form the license plate of the city of Aachen (AC), the process control engineering (PLT) and object management.
All other classes are part of the human machine interface ACCPLT/MMI. Four abstract classes are derived from the root class:

- IF — Interface Element Class
- KsPart — Communication System Class
- PRE — Presentation Base Class
- DAB — Data Acquisition Base Class

From the Interface Element Class are again five classes derived. These are not abstract. They have the methods to exchange data with objects of their on class added to a composition of Presentation or Data Acquisition Classes.
Communication elements can only be added to the Data Acquisition Classes. They have the duty to communicate with the DCS or PLC. Three different kinds of them are implemented:

- KsVariable — read/write values (with state / timestamp)
- KsEngineering — create / delete / rename objects and links
- KsDomain — explore structures of objects and classes

User defined classes can be derived from PRE-Class to create Presentation Block Classes. Every new class is a composition of a individual PRE-Class with overwritten functions and added interface elements.
User defined DAB-Classes are derived from the DAB-Class with overwritten functions and added interface elements and communication elements.
To configure the graphical user interface objects of the user defined classes must be created, connected and parameter settings done.
If this object oriented model is abstracted again it shows a combination of two well known models for user interface systems the Seeheim-Model and the MVC Model.

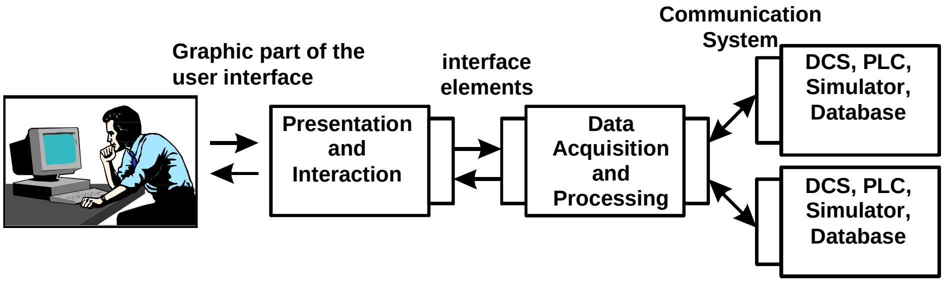

Fig. 5: Model of the man machine interface for process control systems

6. Summary and Outlook

The goal of this research work was to develop a new model as base for the next generation of user interface system in process control systems. Therefore all requirements were collected. Starting with system technology of operator stations over communication systems up to human centred design support a long list of aspects have to be carefully attended. The lifecycle and engineering requirements were mentioned too.

A user interface system fitting to all the duties coming along by interaction with different systems must be well designed. The special demands of process control must be met. Therefore the "two block model" is designed. It gives the possibility to be configured online. New libraries can be loaded at runtime. The programming of new libraries is supported by a design pattern for the most often used methods. Reuse of nearly every program code is done by using an object oriented model.

This model is built on the ACPLT/OV a publish model for object management in process industry. It is as well as the communication model ACPLT/KS [Albrecht, Peters(1998)]available as open source software. The open source system wxWindows [wxWindows(2000)] [Cochran(2000)] is used as GUI-Toolkit. The user interface system called ACPLT/MMI (abbreviation for human machine interface (MMI)) is to be published as open source as well.

More interesting as the realised demo-software is that the model is to be discussed and used for all future graphical user interface systems in distributed control systems.

REFERENCES

[Albrecht, Peters(1998)] Albrecht, H.; Peters, B.:Communication in Process Control, 9th IFAC Symposium on Automation in Minig, Mineral and Metal Processing MMM '98, Colongne 1.-3.Sept. 1998

[Aspen(2000)] Aspen - InfoPlus.21, Process-oriented Information Systems & Applications, WWW-Page, date Februar 2001, adress : http://aspentech.com

[Birkhofer(1999)] Birkhofer, R.: Die Prozeßleit-technik im Spannungsfeld neuer Standards und Technologien, atp 1 / 1999

[Brodlie(1992)] editor Brodlie, K. W.: Scientific Visualization - Techniques and Applications, Springer-Verlag Berlin Heidelberg 1992

[Brucker, Tauchnitz, Eul(1996)] Brucker, A.; Tauchnitz, T.; Eul, J.: Interkama 95: Prozeßleitsysteme, atp 4 / 1996

[Cochran(2000)] Cochran, Sh.: wxWindows 2.2 Offers Cross-Platform Alternative to Java, Dr. Dobb's Journal, ISSN 1044-789X, 8/2000

[Fromme et al(2000)] Fromme, K.-P.; Seitz, M.; Magin, Ch.; Morr, W.: INTERKAMA 1999: Prozeßleitsysteme, atp 42 (2000) 4, S. 17-31

[Heim(1996)] Heim, M.: Konfigurationsräume der Mensch-Prozeß-Kommunikation, Verlag der Augustinus Buchhandlung, 1996

[Ingendahl(1998)] Ingendahl, N.: Konzept zur Analyse der Aufgabe und Auswahl der Präsentation für die Mensch-Prozeß-Kommunikation , Verlag Mainz, Aachen 1998

[IEC61499(1996)] IEC 61499 TC65 WG6: Function Blocks, General Requirements. Working Draft IEC 61499, 1996

[Nirschl(1996)] Nirschl, G.: Interkama 95: Bedien- und Beobachtungstechnik, atp 4 / 1996

[OSI(2000)] OSI Software, Inc., Plant Information System (PI): WWW-Seite, Stand Februar 2001, Adresse: http://www.osisoft.com

[Peters(2000)] Peters, B., Epple, U.: The Two-Block-Model to communicate with processes, 7th IFAC Symposium on Automated Systems Based on Human Skill, Aachen, Germany, Juni 2000

[wxWindows(2000)] The Gui Toolkit, WWW-page, date: april 2000, Adresse: http://wxWindos.org