

A Model-Based Graphical User-Interface for Process Control Systems in Manufacturing

X.J. Li^{a,b}, T. Schlegel^a, M. Rotard^b, T. Ertl^b

^a *Fraunhofer Institute for Industrial Engineering, Nobelstr. 12, 70569 Stuttgart, Germany*

^b *Institute for Visualization and Interactive Systems, University of Stuttgart, Universitaetsstr. 38,
70569 Stuttgart, Germany*

Abstract

The communication of humans with manufacturing processes – respectively their representation in software – is an integrated cooperation. It includes information acquisition via sensors, information processing by control systems and feedback information. This supports control personnel in supervising and operating the processes using information captured in real time. This paper proposes an approach of building an innovative graphical user-interface for intelligent process control systems based on the analysis of various requirements for process control of today's manufacturing. Information and presentation models are used for transforming real-time production data into a dynamic and easy-to-use graphical user interface using XML technology.

Keywords: INT-MANUS, process control, presentation model, XML, transformation, user interface

1. Introduction

The manufacturing systems of today consist of various processes from planning to transport. Technical production processes are initialized by the human and accordingly must be monitored and controlled by the human. The human-process communication of manufacturing is an integrated cooperation that works together with information acquisition via sensors, information processing by control system and information feedback which supports control personnel to manipulate the processes with the captured information in real time [1].

Today's manufacturing requires real-time monitoring and control of the production process through innovative mechatronics. The EU project INT-MANUS (Intelligent Networked Manufacturing System) addresses these problems by developing a

Smart-Connected-Control (SCC) Platform that will allow decentralized control of the production process in a new way. One important part of such an SCC platform is a user-friendly interface that visualizes and interacts with complex data of production processes. The user interface will be embedded into a display wall [2] that enables the supervisors to monitor the real-time process values and on their PDAs handled by the control personnel [3].

This paper proposes an approach of building an innovative graphical user-interface for intelligent process control systems based on the analysis of various requirements for process control of modern manufacturing. In order to distinctly model the functions of a user-interface the tasks of this work are divided into the information model [4] and the presentation model. In the information model, the

process values are modeled according to data structures and data types. The presentation model describes the graphical objects that present the data to the user in various manners. The bridge to connect the two function models are interactive elements. The objective of this work is to build information and presentation models which are used for transforming real-time production data into a dynamic and easy-to-use graphical user interface using XML technology.

2. Determining needs of process control in different problem domains

The analysis of requirements for a process control system is the foundation of building models for human-machine interfaces. The restriction of the requirements is the basis of choosing graphical user interface technologies. Process control engineering encompasses all technical means that assist humans to control a process according to previously captured requirements. In automation systems the processes are controlled by monitoring the process values and allow full access to measurement and configuration of execution data. In this work three problem domains in process control systems are discussed: real-time monitoring and control, statistical process control, and phase model of production processes.

2.1 Real-time monitoring and control

One of the important functions of a user-interface for process control systems is to display current process information in real-time because the control personnel must acquire the process values for process management.

General information on a production process, such as the order of the products, the model of the target product, the number of workers, etc. is essential information required by all production processes.

Besides this general information the real-time process values include also the inner information of partial processes and atom elements, for example, the running state of a machine. For a certain value the exact address from where a value is coming is important as well as attributes and type of the value. For further usage in deciding what kind of (interactive) element is used, it should also be considered, whether a value is controllable or not.

For reliable process control the ability of response to exceptions and errors is important and

necessary in case a value exceeds or falls below a limit. If the control person is just alarmed when the value is already out of the threshold, it will be too late to avoid dangerous errors. Therefore, the alert range of a value should be definable in model in order to warn the control person when the value approaches the threshold value.

Table 1 shows an example for a complex data type "machine temperature", which has a valid range from 18°C to 27°C. Lower than 15°C and higher than 32°C the alert will be started.

Table 1
An example of complex data type - Temperature

	Unit	Control	Max	Warning Max	Min	Warning Min	Current
Data Type	string	bool	float	float	float	float	float
Example	°C	false	32	27	15	18	25

2.2 Statistical process control

For the quality control of a manufacturing system not only the current process values but also the trend and history or statistics of the values are of interest to the control personnel. In other words, what is happening currently and what has happened previously should be analyzed by the system. In the database for a production process the process values during a certain time should be stored in an archive.

Normally the history and trend of a process are displayed in a diagram or a chart with an axis for value and an axis for time. The following questions should be considered for the statistical process control:

- How long should be the history reported?
- How many values should be compared in a diagram?
- How many curves should be displayed at the same time?
- How should the coordinates be scaled?
- Has the Diagram more than one abscissa or ordinate?
- How is the history of a binary value presented such as the running state of a machine?
- What is the sample rate or sensors refresh frequency of the statistical diagram per minute, per hour or per day.

Diagram representations for engineering data like histograms, check sheets, scatter diagrams, and control charts and more ways to visualize the process values are described and illustrated in [5].

2.3 Phase Model

A manufacturing process system can be described in different models. The model for the workflow of a process is called phase model [6]. Here, a process can be decomposed into more process elements and each process element can again be hierarchically composed of other process elements. A process can exist individually or serve as successor or predecessor of another process.

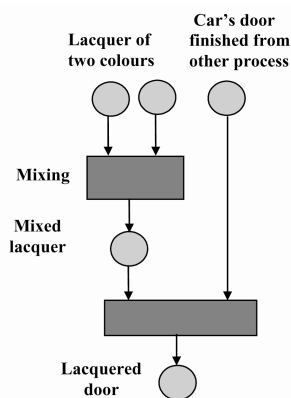


Fig. 1. An example of phase model

A simple process contains process elements, input product as well as semi-finished and end products from process elements and other output elements. Fig.1 shows a simple phase model of the process for lacquering the door of car. The door of car is an output product produced from other processes and acts in this process as input product. The first process element (Mixing) accomplishes mixing of the raw materials. The output product from this process element works together with another input product (door) through the second process element (Lacquering).

3. Information models of process control systems

Manufacturing a process is an integration of products, machines, and process elements. We attempted to model our user-interface for manufacturing systems step by step from simplicity to complexity. At first the models for the simplest entities such as products and machines are built with their attributes, each entity must have a unique identifier in the whole production process in order that it can be easily referenced by the other models

without confusion. Based on the models of all entities it should be considered how the entities are combined for a complex function. Fig. 2 shows an entity model of a machine. In this work, the information model is built on *Product Model*, *Machine Model* and *Process Model* that will be explained later.

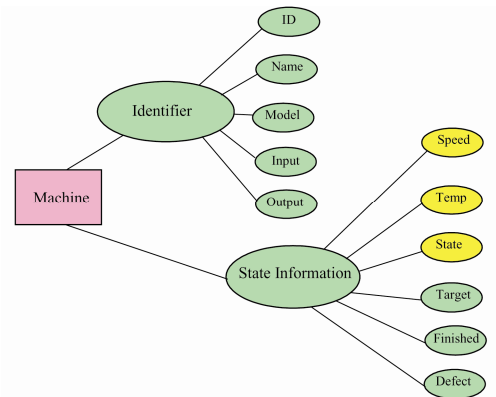


Fig. 2. An entity model of a machine

Identifier. In order to let a process control system run trouble-free it must be avoided that a component of the system has different meanings or a notation corresponds to more than one component. In this case the ambiguous relationship of system components will lead to confused operations and false control. Therefore, the uniqueness of all components in the control system is necessary.

To make a product unique in the control system, the product model should contain the information of both the inner properties of the product such as name and unit and the exterior properties such as its position in a process. For a machine, the function of the machine, the product which the machine produces, and the position of a machine in a process are considered for a unique reference in the whole system. Similar to product and machine, a unique id of a process denotes a sub-process, which works for a certain processing phase and has definitive predecessor and subsequence.

Product model. A product model can be described in a class. For each special product an instance of this class or model is created. One product model can correspond to more than one product with variable parameters which are distinguished by unique ids. Because each id corresponds to one product for a certain usage the control person can monitor the states of all products by choosing the product id.

Machine Model. To model a tool machine the

configuration and state information of the machine should be considered. The configuration of a machine is determined by the usage and functions of the machine.

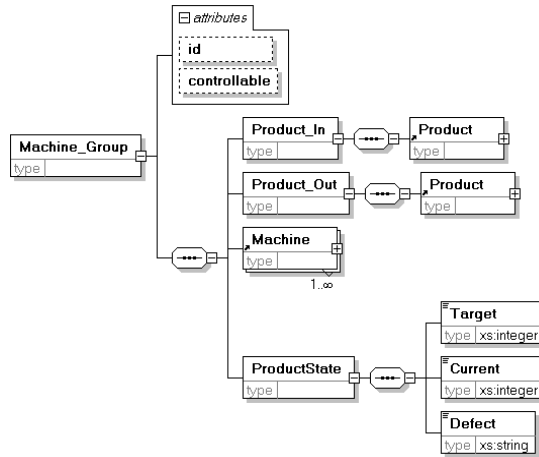


Fig. 3. Model of machine group in XML Schema

For different functions the information is divided into two blocks, the inner information and the external information. The inner information of a machine exists free of context, that is, the information is independent of other machines and the position in a process. The inner information is mostly defined by the machine manufacturer, limited by sensors and specification.

The external information of a machine is context sensitive that depends on cooperating machines and the function of the machine in a process. The external information determines the logical relationship between the machines and other cooperative compositions.

In practical production processes it is possible that some machines are designed for multiple functions or more than one machine is used for one task/purpose. Such machines with complex functions are built based on the model of single machine with single function. Fig.3 shows a model of a machine group that is built with referenced models of single machine and products.

Process model. A process element model can be built by combining the models of product and machines with additional process information. A process consists of process elements and products. These are connected according to their functions and their positions in the process – referenced to their unique identifiers.

4. Modeling with XML Schema

In this work we adopted XML Schema to build the information model by defining the data of process values with their data types in XML Schema which can be treated as a meta-model for the farther modeling. In XML Schema an arbitrary data type can be easily defined and referenced anytime when required. Besides, the XML Schema diagram enables the developer to have an entire concept of the model.

Step 1: Building models from atom elements. Each atom element is defined in a separate model. Complex data types are defined by combination of simple data types and other complex data types.

Step 2: Associating models of atom elements. The process model is built stepwise by associating product models, machine models and process element model. Each atom model can be referenced repeatedly.

Step 3: Application-specific models. Based on the models defined in XML Schema the data of process values can be specified in XML format corresponding to different tasks.

5. Presentation Model

The presentation model describes which graphical elements will be presented to the users. According to the functions of a user interface for process control we divided the presentation model into two major parts: navigation model and process information display.

In the navigation window the graphical navigation flow chart, in which each process element is represented by an icon, is used to visualize the production phase for quick switch between processes [7]. The sequence and relationship of components in the flow chart should correspond to the real process phase model.

In the information window various minor user interface elements such as process bar, control slider, diagram, etc. are employed to display the process information and receive user commands. The selection of suitable minor user interface elements is driven by the information models described in the previous sections.

6. Transformation

To find a consistent and appropriate representation for information elements, it must be considered how the data can be transformed into a presentation object. The concept of the two-block

function model [8] is adopted for transforming data elements to graphical elements. With the two-block function model the task of a user-interface is divided into two blocks, one is the information model and the other is the presentation model, both are already described in the previous sections. The bridge to connect the two function model is interactive functions that determine how the data is transformed into graphical elements.

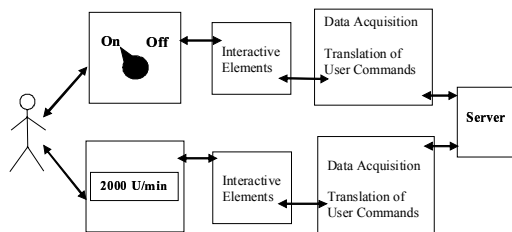


Fig. 4. Transformation of data and user commands

After the instances of the information model have been created, the data types of the concrete process values influence directly their representation by the presentation objects. Interactive elements play an important role as management of data model and presentation objects. The data and the commands from users are translated by the interactive elements such that suitable presentation objects in the presentation model can be chosen.

Fig. 4 shows the approach of transforming information structures and data into presentation objects. The interactive elements translate the process values that have been acquired from a central server and user commands via user interface. The suitable presentation objects are chosen according to the data types and user commands.

In our scenario, we have identified a two-dimensional graphical presentation as the best suited user interface form for a process control system, especially a control centre. In this work we declare the method of transformation with example of the web application.

The left XML code in Fig.5 describes an instance of the temperature model. Because the XML elements are stored a DOM tree structure by an XML parser, all elements can be traversed. First the element “temperature” is found and the value of the node attribute “controllable” is read for determining whether the element – here temperature – is controllable or not. If “controllable” is false, the predefined design pattern for a non-controllable (non-interactive) temperature element is called from resource. Similarly, the values of children nodes

“MAX”, “MIN” and “Actual” are interpreted and assigned to corresponding parameters.

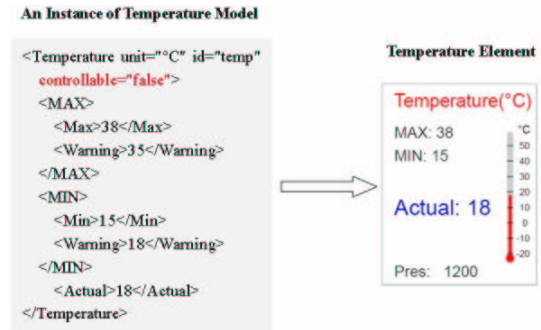


Fig. 5. Presentation object from XML document

Another method for transforming the information model into presentation objects for a web-based interface is using XSL Transformation (XSLT) [9]. XSL Transformations is a template language expressed in XML syntax. XSL was developed to add style to an XML document and XSLT was designed to be more general and to allow the transformation of documents into any documents of XML type like XHTML or and Scalable Vector Graphics (SVG) [10]. With different XSL Transformation an information model can be transformed to user-interfaces for different platforms and specific users.

7. Generation of Model-based User Interface

In our case, the application control will be done using a web interface, so we have chosen SVG [11, 12] as the target for the user interface generation. This vector-based representation has the advantage of being an XML format and allowing for lossless zooming, needed when scaling the control centre interface to a PDA-based version with conformity to user expectation – using the same elements. The specification SVG Tiny provides a solution for the generation of user interface consisting of SVG components for the PDAs and Mobiles. The DOM mechanism of SVG enables full access of data and various event handlers.

For all purposes in a process control system, the interfaces can be generated for different platforms. For desktop-oriented graphical interfaces the *Batik SVG Toolkit* [13], a Java based toolkit could be used. For web interfaces today’s web browsers can render SVG natively or need a special plug-in.

8. Conclusions

Building a dynamic user-interface for production control centers it is important to consider that the essential of process control engineering is information exchange and visualization as well as the communication between human controllers and the production system. The basis for a model-based, dynamic user-interface is therefore the information models. To build an information infrastructure we chose to define the data of process values with their data types in XML Schema. Based on this XML Schema document, the defined models with adjacent data can be specialized in XML format corresponding to different tasks. The models built in this work prove that XML Schema can effectively define a meta-model of information structures even with complex data types. In our approach the generic information model plays the role as a meta-model consisting of a set of first-class modeling artifacts allowing for keeping it small and invariant. Compared to the meta-model the application-specific models are the lower layer of abstraction, they do not introduce new artifacts but are depicted using the meta-model artifacts.

In order to better model and manage the user interface for information exchange, the functions of data exchange through user interface are separated into data acquisition and data representation. Both parts can be modeled individually and connected by interactive elements. The separation of functions benefits code separation and code reuse. Presentation models rely on the previous built information models, according to specific task only limited presentation objects are required. The data types in the meta-model give us an early perception of which presentation objects may be possibly used.

Acknowledgments

This work is related to INT-MANUS project, funded by the European Commission. Fraunhofer IAO is a member of I*PROMS network of excellence, supported by the European Commission.

References:

- [1] Peters, B.; Epple, U.: A generic object model to build the human-process interface. 6th IFAC Symposium on Automated Systems Based on Human Skill, September 1997.
- [2] Wallace, G.: Tools and Applications for Large-Scale Display Walls, IEEE Computer Graphics and Applications, vol. 25, no. 4, pp. 24-33, August 2005.
- [3] Farella, E.; Brunelli, D.; Bonfigli, M. E.; Benini, L.; Gaiani, M.; Ricco, B.: Using Palmtop Computers and Immersive Virtual Reality for Cooperative archeological analysis: the Appian Way case study, In International Conference on Virtual Systems and Multimedia (VSMM) Gyeongju, Korea, 2002.
- [4] Albrecht, H.: On Meta-Modeling for Communication in Operational Process Control Engineering, VDI Verlag, 2002.
- [5] Engineering Statistics online Handbook <http://www.itl.nist.gov/div898/handbook/>, last update, September 2005.
- [6] Lauber, J.: Methode zur funktionalen Beschreibung und Analyse von Produktionsprozessen als Basis zur Realisierung leittechnischer Lösungen, Verlag Mainz, 1996.
- [7] Constantine, L.L.; Lockwood, L.A.D.: Software for Use: A Practical Guide to the Models and Methods of Usage-Centered Design, Addison-Wesley Professional, 1999.
- [8] Peters, B.; Epple, U.: The Two-Block-Model to communicate with processes, 7th IFAC Symposium on Automated Systems Based on Human Skill, June 2000.
- [9] XSL Transformations (XSLT) Version 1.0 Specification, W3C Recommendation, November 1999.
- [10] Froumentin, M.; Hardy, V.: Using XSLT and SVG together: a survey of case studies, SVG Open Conference 2002, July 2002.
- [11] Fettes, A.; Mansfield, P.: SVG-Based User Interface Framework, SVG Open Conference 2004, September 2004.
- [12] Scalable Vector Graphics (SVG) 1.1 Specification, W3C Recommendation, January 2003.
- [13] The Apache Software Foundation: Batik SVG Toolkit, <http://xmlgraphics.apache.org/batik/>, May 2006.