ANL/RA/CP-87983

Full paper to be presented at the Nuclear Plant Instrumentation, Control and Human-Machine Interface Technologies, University Park, Pennsylvania, May 6-9, 1996.

RECEIVED

MAR 27 1996

OSTI

Flexible Human Machine Interface for Process Diagnostics *

by

J. Reifman
G. E. Graham
T. Y. C. Wei
Argonne National Laboratory
9700 South Cass Avenue
Argonne, IL 60439

K. R. Brown
R. Y. Chin
ComEd
125 S. Clark Street
Chicago, IL 60603

The submitted manuscript has been authored by a contractor of the U. S. Government under contract No. W-31-109-ENG-38. Accordingly, the U. S. Government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or allow others to do so, for U. S. Government purposes.

MASTER

*Work supported by the U.S. Department of Energy, Energy Research Programs under Contract W-31-09-ENG-38.

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

PK

FLEXIBLE HUMAN MACHINE INTERFACE FOR PROCESS DIAGNOSTICS

Jaques Reifman^{*}
Glenn E. Graham
Thomas Y. C. Wei

Argonne National Laboratory
9700 South Cass Ave.
Argonne, Illinois 60439, USA
^{*}E-mail: jreifman@anl.gov

Kevin R. Brown
Ronald Y. Chin[†]

Commonwealth Edison Company
125 South Clark Street
Chicago, Illinois 60603, USA
[†]E-mail: g#chin@ccmail.ceco.com

ABSTRACT

A flexible human machine interface to design and display graphical and textual process diagnostic information is presented. The system operates on different computer hardware platforms, including PCs under MS Windows and UNIX Workstations under X-Windows, in a client-server architecture. The interface system is customized for specific process applications in a graphical user interface development environment by overlaying the image of the process piping and instrumentation diagram with display objects that are highlighted in color during diagnostic display. Customization of the system is presented for Commonwealth Edison's Braidwood Chemical and Volume Control System with transients simulated by a full-scale operator-training simulator and diagnosed by a computer-based system.

1. INTRODUCTION

Argonne National Laboratory (ANL) and Commonwealth Research Corporation (CRC), the research arm of Commonwealth Edison Company (ComEd), are collaborating on a DOE-sponsored Cooperative Research and Development Agreement (CRADA) project to perform feasibility studies on a novel process diagnostic system. The computer-based diagnostic system being developed, PRODIAG,¹⁻⁴ uses artificial intelligence (AI) techniques to assist nuclear power plant operators during off-normal plant conditions. When a process component malfunctions, PRODIAG hypothesizes faulty component candidates based on analyses of the disturbances produced by the malfunctioning component.

In contrast to existing advisory diagnostic systems that directly map transient symptoms into component faults based on an event-oriented approach, PRODIAG incorporates fundamental physical principles into the diagnostics through

a function- and component characteristic-based approach. Fundamental physical principles in the form of conservation equations^{2,3} and thermal-hydraulic (T-H) characteristics of the process components^{1,4} are used to emulate the reasoning process of system analysts. This unique approach results in a diagnostic system that is (1) capable of identifying *unanticipated* events, (2) for the most part is plant- and system-independent and therefore portable to other T-H processes, (3) is easier to verify and validate, and (4) is suitable for systems with limited instrumentation.

PRODIAG combines two AI techniques, expert systems (ESs) and artificial neural networks (NNs), in a two-level hierarchical structure to successively narrow down the diagnostic focus of the system at each level. At the first or top-level, a physics-based ES with knowledge of the overall process, including the detection of macroscopic imbalances in the conservation of mass, energy, and momentum inventories and the design function of the process components, performs plant-level diagnostics. By correlating the type of the detected imbalances, e.g., mass, energy, momentum, with the design function of generic components, e.g., open valves and pumps are designed to provide a source of momentum, the ES makes hypotheses about the possible faulty components. For example, when a momentum imbalance is detected, components designed to provide a source of momentum are hypothesized.

At the second level, a hierarchical structure of NNs with increasing discriminatory power at each level is then used to successively discriminate among the hypothesized faulty components based on the T-H characteristics of the components. The NNs first discriminate among classes of generic components which perform a common function, e.g., open valves versus pumps, and then among specific components within a given class, e.g., gate valve 1 versus globe valve 2.

As part of the ANL/CRC CRADA project, a Graphical Diagnostic Interface (GDI) system was also developed to provide the process operator with the diagnosis inferred by PRODIAG. GDI is a flexible system to design and display textual and graphical process diagnostic information. Like PRODIAG, GDI is a process-independent tool that is customized to the specific T-H process monitored by the diagnostic system and operates on a full range of computer hardware platforms in a client-server architecture. GDI can also be used with different diagnostic tools other than PRODIAG.

In this paper we discuss the main features of the GDI system. We present the customization and application of the system to the diagnosis of component malfunctions in ComEd's PWR Braidwood Chemical and Volume Control System (CVCS). Component malfunctions are simulated with the CVCS full-scale operator-training simulator and diagnosed by the ES portion of PRODIAG.

II. SYSTEM OVERVIEW

The GDI system is written in C and is developed for multiple windowing environments by using a set of libraries which support windows on multiple hardware platforms. XVT⁵ libraries provide cross-platform functionality for graphical user interface (GUI) standards such as menu pulldowns, list boxes, file open, dialog boxes, and scroll bars, while ViewDirector⁶ image decompression libraries are used to display, zoom, and scroll graphics. The system is designed to operate on a client-server architecture where GDI could be run on a PC under MS Windows or X-Windows terminal, and PRODIAG could be run across a network of workstations. In the current system architecture, both GDI and PRODIAG (written in Prolog⁷) run on a UNIX platform under X-Windows (NCDware) with SunOS 4.1.3 operating system on a color monitor.

The GDI system displays sensor values and diagnostic information inferred by PRODIAG through two windows: a textual window and a graphical window. In the textual window, diagnostic information is provided through written statements, such as pipe break between valve cv-121 and reactor coolant pump filter, that specify the hypothesized faulty components. In the graphical window, the hypothesized faulty components are highlighted in the image display representing the piping and instrumentation diagram (P&ID) of the process being diagnosed. The highlighting consists of changing the color from black to red of the faulty components specified in the textual window. The values of process sensors are also displayed in this window.

PRODIAG and GDI interact through a small ASCII interface file. The file is written by PRODIAG at every

diagnostic time step which is then processed and deleted by GDI. The interface file contains diagnostic information that is used to update the textual window and refresh the graphical window with the current diagnosis. Text messages are passed to the textual window and attribute values of dynamic objects (discussed in the next section) are passed to the graphical window. The attribute values are used to identify which components of the P&ID should be highlighted and to update the values of the sensors. The interface file is also used to synchronize sensor readings with the diagnosis performed by PRODIAG and displayed by GDI.

The GDI system has two modes of operation: design and display. The design mode of operation is used to customize the system to the P&ID application of the specific T-H process monitored by PRODIAG. Once the graphical interface has been customized, GDI is switched to the display mode of operation where it is then used as the interface between the diagnosis performed by PRODIAG and the process operator. Under the display mode of operation, the system is completely automated. In the next section, we discuss the customization of the P&ID for a specific application of the system.

III. GRAPHICS CUSTOMIZATION

Under the design mode of operation, the user customizes the graphical window display for the specific P&ID application by overlaying a black-and-white image of the P&ID with display objects. The first step of the customization is to import the black-and-white background image of the process P&ID into the working hardware platform to serve as the raw background image onto which objects are overlaid. If the P&ID image is drawn on paper, it can be imported by scanning the image and storing it as a compressed bit-map TIFG4 file. TIFG4 is one of many formats used to compress bit-map files. If the image is generated by a computer aided design program stored as a .DWG file, it can be directly imported by transforming the .DWG file into a raster file and saving it as a TIFF (Tagged Image File Format) file. Figure 1 illustrates the raw background P&ID image used to customize the graphical window display for the Braidwood CVCS.

With the background image imported into the working platform, the user can then customize the display image by creating display objects that overlay the raw background image of the P&ID. Objects are created in a GUI development environment by first using point-and-click mouse actions to shape and size the display object and then by using dialog boxes to specify the object ID or name and associated attributes. There are two types of objects: static and dynamic. There are three types of static objects, strings,

lines, and boxes, whose attributes, e.g., color, are defined during the creation of the object and remain fixed during the diagnostics. Static objects are used to add missing information to the raw background image. For instance, strings are used for labeling components, while lines and boxes are used for representing missing pipes and tanks, respectively.

There are four types of dynamic objects: node, value, line, and time, whose attributes, e.g., color and numerical value, are allowed to change during the diagnostics in order to highlight the hypothesized faulty components and display the changing values of the process sensors. For instance, dynamic objects such as *lines* are overlaid on the raw background P&ID piping to illustrate through a change of colors, e.g., from transparent color to red, that a pipe break is a hypothesized fault.

Node objects are represented by rectangular boxes overlaid onto process components other than pipes, e.g., valves, pumps, filters, that are used to highlight their faulty state. Color is the only attribute of dynamic nodes. The color of a node can change from transparent, i.e., not visible, at normal process conditions to red, blue, black,

green, cyan, magenta, or yellow, during off-normal conditions to alert the process operator. *Value* objects are created to display numerical values of sensor readings. *Value* objects have two attributes: numerical value and color. Numerical values change during the transient to indicate the changing process state and are highlighted by changing its color. For instance, the color of a sensor reading can change from black, indicating that the signal value is within an expected envelope, to green or yellow, to indicate a signal value above or below, respectively, of the expected signal value.

Line objects are horizontal or vertical segments overlaid onto the process piping that are used to indicate blockages and breaks in the piping. *Lines* are overlaid onto the raw background P&ID piping with transparent color, i.e., not visible, to indicate normal process conditions. During diagnostics, the color of the *line* objects may then change to red or other color to indicate a piping blockage or break and alert the process operator. Color is the only attribute of dynamic lines. *Time* objects are used to indicate the transient time and have only one attribute, a positive numerical value.

Figure 1. Black-and-White Raw Image of a Piping and Instrumentation Diagram Used as the Background for Customization

Figure 2. Graphical Diagnostic Display After the Background Image Has Been Overlaid With Static and Dynamic Objects

The result of customizing the raw background image of the CVCS in Figure 1 with static and dynamic objects is illustrated in Figures 2 and 3. Figure 2 shows the image display used as the graphical interface (the graphical window) for PRODIAG. When an operational transient occurs, this image indicates it graphically by highlighting in red the faulty components hypothesized by PRODIAG. This image is also used to indicate increasing and decreasing trends in the sensor readings by changing the color of the numerical value of the sensor to green and yellow, respectively. In Figure 2, the numerical values of the sensors have been normalized relative to their expected values.

The image shown in Figure 3 illustrates the locations of the twenty⁴ possible component malfunctions simulated by the Braidwood simulator that were used to test PRODIAG. These locations are marked by four-character labels located inside rectangles, with the labels indicating the type of the malfunction. For instance, CV01 indicates the failure of pump A. Only static objects were used to customize this image. The user may switch or tunnel between the images in Figures 2 and 3 by clicking the small square located in the middle of the two figures. GDI allows for tunneling among multiple images that can be used to indicate more detailed information not illustrated in the main graphical interface.

BRAIDWOOD CHEMICAL AND VOLUME CONTROL SYSTEM TRANSIENT LOCATIONS

Display Mode

IV. PRODIAG AND GDI INTERFACE

The interface file contains the attribute values of the dynamic objects and the textual diagnostic information generated by PRODIAG. In the interface file, each dynamic object is identified by a unique name or ID that coincides with the ID given to the object during its creation in the design mode of the GDI operation. Adjacent to each object ID are the associated attribute values that are instantiated to reflect the diagnosis at the current time step. Table I shows the structure of the interface file. The first column of the table presents the four types of dynamic objects: time, value, node, and line, of the graphical window. The second column illustrates with a few examples each one of the four dynamic object types. The examples are followed by the

word TEXT, which is used to flag the beginning of the text information sent to the textual window. The text information, e.g., There are no malfunctions in the system, is followed by the word EOF, which is used to indicate that GDI should update the two windows.

Table I. Structure of the Interface File

Dynamic Object	Examples
Time	5.0
Value	cold_leg3:temp:out yellow 0.985 cont_sump:level:l1 black 1.000 h2_tank:flow:out black 1.000
Node	cold_leg3 white v_8389a white s_reg_htx red
Line	junc_13:rv_sw_htx white rv_sw_htx:junc_16 white unc_16:junc_12 red
	TEXT There are no malfunctions in the system EOF

There is only one entry to the dynamic object *time*, which indicates the current transient time, e.g., 5.0 seconds, through a positive numerical value. There are as many *value* objects as sensors in the P&ID. *Value* objects are identified by their name or ID, e.g., cold_leg3:temp:out, cont_sump:level:l1, which must coincide with the IDs used in the customization of the background image. *Value* objects have two attributes, a color, e.g., yellow, black, and a numerical value, e.g., 0.985, 1.000. *Node* objects are also identified by their ID, e.g., cold_leg3, v_8389a, and color is their only attribute. There are as many *node* objects as components (other than pipes) in the P&ID. Like *node* objects, *line* objects are defined by their ID, e.g., junc_13:rv_sw_htx, and color is their only attribute. There are as many *line* objects as pipes connecting adjacent components in the P&ID.

V. GRAPHICS DISPLAY

The GDI system was used as the human machine interface in the diagnoses of ninety-seven simulated CVCS transient events inferred by PRODIAG.⁴ The transient events were simulated with the full-scale operator-training Braidwood simulator. In addition to being used as the human machine interface for simulations of transient event diagnostics, we have also found the system to be very useful during the developmental stage and debugging of the PRODIAG diagnostic system. GDI can also be used with different diagnostic tools other than PRODIAG.

Figure 4 shows the graphical display window and textual window of GDI at 5.0 seconds into one of the CVCS simulated transient events. In the graphical window, the customized background image is represented by the black lines forming the P&ID of the CVCS. Annunciation of the process anomalies is represented by highlighting the deviating sensor readings and hypothesized faulty components. In Figure 4, sensors FT-121, PI-120, TE-127, PI-140A, SI filter DP, PI-140B, and FE-145 are highlighted indicating that their values are either above or below their expected value. In a color monitor, the numerical values of these sensors would appear in green or yellow indicating increasing or decreasing trends, respectively. The sensors are represented by dynamic *value* objects.

The figure also indicates three faulty components hypothesized by PRODIAG, valve cv-121, SI filter, and regenerative heat exchanger, through the drawing of rectangular boxes surrounding these components. The boxes represent dynamic *node* objects. The pipe segments connecting these three components are also highlighted indicating a possible piping break. *Line* dynamic objects overlaying the background P&ID piping were used for this purpose. In a color monitor, the hypothesized process components and piping are indicated in red.

The user can indicate these components as well as the deviating sensor values in any of the following colors: red, blue, black, green, cyan, magenta, and yellow, through the instantiation of the corresponding attribute values in the interface file. The user may also resize, scroll, or zoom the display image of the graphical window through the View pull-down menu located on the upper left corner of Figure 4, or by point-click-and-drag mouse actions. The other pull-down menus, *File*, *Edit*, *Design*, *Align*, *CRADA*, and *Comment*, are used to open and close files, exit GDI, edit objects, toggle between design and display mode of operations, align objects during customization, refresh the two interface windows, and send comments to the software designer.

Figure 4. Diagnostic Display of the Graphical Window and Textual Window at 5.0 Seconds Into the Transient

Figure 4 also shows the textual window used to explicitly indicate the hypothesized faulty components highlighted in the graphical window. For instance, it indicates that at 5.0 seconds the process anomaly is caused by either a pipe break between junction 7 and the tube-side of the regenerative heat exchanger, or between valve cv-121 and the reactor coolant pump (or SI) filter. The user may also resize and scroll the textual window through point-click-and-drag mouse actions. Unlike the diagnostic information displayed in the graphical window which can not be reviewed once refreshed, the information in the textual window is appended to the diagnosis of the previous time steps allowing the user to review the entire diagnostic history.

VI. SUMMARY AND CONCLUSIONS

In this paper we describe a flexible human machine interface to design graphical displays and display process diagnostic information. The interface system is customized for specific process applications in a GUI development environment by overlaying display objects onto images of existing process P&IDs. This procedure allows for quick development of graphical interfaces that are easy and simple to design and modify without the need to write software programs

ACKNOWLEDGMENTS

This work was supported by the U.S. Department of Energy, Energy Research-Laboratory Technology Transfer Program, under contract W-31-109-ENG-38.

REFERENCES

1. J. Reifman, T. Y. C. Wei, R. G. Abboud and T. M. Chasensky, "Cooperative Research and Development for Artificial Intelligence Based Reactor Diagnostic System," *Proc. American Power Conference*, Chicago, Illinois, April 25-27, 1994, p. 365, Illinois Institute of Technology, Chicago, Illinois (1994).
2. J. Reifman and T. Y. C. Wei, "Systematic Construction of Qualitative Physics-Based Rules for Process Diagnostics," *Proc. 7th Portuguese Conference on Artificial Intelligence*, C. Pinto-Ferreira and N. J. Mamede, Eds., Funchal, Madeira Island, Portugal, October 3-6, 1995, p. 311, Springer-Verlag, Berlin (1995).
3. J. Reifman, T. Y. C. Wei, R. G. Abboud and T. M. Chasensky, "Function-Based Approach to Plant Diagnosis," *Trans. Am. Nucl. Soc.*, **73**, 126 (1995).
4. J. Reifman, T. Y. C. Wei, J. E. Vitela, C. A. Applequist, and T. M. Chasensky, "PRODIAG - A Hybrid Artificial Intelligence Based Reactor Diagnostic System for Process Faults," to be presented at the 4th International Conference on Nuclear Engineering, New Orleans, Louisiana, March 10-14, 1996.
5. XVT Software Incorporated, Release 3, Boulder, Colorado (1993).
6. TMS Incorporated, Version 2.0, Mountain View, California (1993).
7. Quintus Corporation, Release 3, Palo Alto, California (1991).

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.