The Expanded Role
of
Software
in a
Control System

Joe Damico

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.

MASTER

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

for
RECEIVED

MAR 08 1994

OSTI

Disposal of hazardous wastes such as chemical waste and military munitions present a serious environmental challenge. A new technology, Supercritical Water Oxidation (SCWO), provides a safer, cleaner process for disposing of these and other wastes. Sandia National Laboratories built a SCWO reactor, called the Engineering Evaluation Reactor (EER) to study the SCWO process. Designed as a flexible testbed, the EER required a control system that would accommodate a variety of experimental instruments, configurations and control schemes. Due to safety concerns, the design required that the control system provide remote operation of the reactor. In the recent past, such a control system would be custom built from hardware components. The EER control system implemented many hardware functions in software and proved more flexible than a more hardware oriented system. This paper describes the design and implementation of the EER control system and discusses the advantages of a software based control system.

The EER control system, a virtual instrument, incorporates all reactor functions into a graphical user interface. The operator remotely controls the reactor using a computer monitor and a mouse. A screen copy of the user interface appears on page 2 of the Appendix. The user interface software is easily modified to reflect changes to the reactor. In more hardware oriented designs, the system controls, like switches and transducer readouts, are mounted in rack panels. The rack panels can present an intuitive user interface, but modifications to the experiment require control panel modifications. Often, the control system interface does not keep up with experiment changes and becomes non-intuitive and unsafe. Considering the constant changes planned for the EER, we implemented the control system using software as much as possible.

The Engineering Evaluation Reactor

The EER reactor, shown schematically on page 1 of the Appendix, combines the technologies of high pressure systems, combustion science, and chemistry. In

the supercritical state, hazardous wastes, acting as fuel, dissolve easily into water and react with oxidizers producing innocuous combustion products. These products include water, inorganic salts, carbon dioxide, molecular nitrogen and other gases depending on the waste. The EER uses three high pressure liquid pumps that supply the flow reactor with water, oxidant, and waste product. The reaction takes place in a 40 foot section of high pressure tubing that, when heated and pressurized, brings water to its supercritical state. Heating elements wrapped on the reactor tubing heat the reactor flow. A pressure regulator keeps the reactor pressure to within the supercritical range and sampling valves allow collection of effluent for analysis. The control system combines software, electronics, and electrical power and integrates the other technologies into a working reactor. The control system hardware forms the interface between the EER and the control system software.

The Control System Hardware

The control system hardware runs on a Macintosh Quadra 950 computer. Four interface boards provide data acquisition and system control. The Direct Memory Access (DMA) board provides timing control for the system and DMA for the data acquisition board. Data acquisition is handled by one 8 channel Digital-to-Analog Converter board and an external chassis that multiplexes the 8 channels into 96 data channels. An analog output board provides six channels of voltage or current control signals. A 32 channel Digital Input-Output board provides On-Off signals to the valves and pump controllers.

The control system hardware includes three chassis. The Temperature Controller chassis contains sixteen independent temperature control loops that interface to the computer with an RS-485 link. These control loops regulate the power delivered to the reactor heaters to control the liquid temperature in the reactor.

The Analog Output Chassis breaks out the voltage and current control signals from the Analog Output board and powers the 4-to-20 milliamp loops. The voltage control channels are individually configured to the appropriate output range, while the current outputs provide 4 to 20 milliamp signals. One voltage control channel controls the system pressure and three current channels control the high pressure pumps.

The Digital Input/Output chassis breaks out the output connector from the I/O board into general purpose connectors. The Digital Input/Output Chassis demonstrates the simplicity of a software oriented system. The chassis converts one connector type into another; inside the chassis one sees only wires running from one connector to the others. In a hardware oriented system, this chassis would contain relays interlocking various reactor functions. These hardwired relays limit the system flexibility. In the EER system, reactor interlocks occur in software using Boolean functions, eliminating the need for interlock relays and allowing easy modifications of the interlocks.

The Control System Software

The EER control system was developed using a graphical programming language, LabVIEW. LabVIEW programs are coded as icon based functional block diagrams. An example of LabVIEW code for the software interlocks discussed above appears on Appendix page 3. The control system program consists of 144 subroutines combined into 7 processes that share data through global variables. Other than sharing data, the processes execute independent of one another. The main process collects data from the reactor stores it in a global array for the other processes to use. One process checks and displays the status of the system controls and transducers. Other processes control the reactor's valves and setpoints by interpreting mouse clicks into commands for the interface boards. The processes share CPU cycles based on priorities. The processes that control the setpoints and valves have highest priorities, while the display related processes have lower

priorities. The assignment of priorities ensures timely reaction to operator commands while allowing reasonable update of the reactor status displays.

The user interface, presented as a series of menus, buttons, and pop up windows, controls reactor operation and control system configuration, like selection of channel and controller types. The reactor controls and status displays overlay a schematic of the EER. Valves open and close and pumps turn on and off with mouse clicks on their schematic representation. Changes appear on the main control monitor instantly through shared global variables. The user interface allows the operator to easily control the reactor by turning mouse clicks on a monitor into operation of reactor components.

The software controls the reactor in a mode known as SCADA, for Supervisory Control And Data Acquisition. Using the interface boards or the RS-485 link, the computer communicates a user selected setpoint value to dedicated controllers, like the 16 temperature control loops, and the dedicated controller controls that parameter to reach the desired setpoint. The computer confirms the setpoint change from acquired data and displays it for the user. Dedicated controllers control the temperature, pump rates and system pressure while the computer concentrates on data acquisition and the user interface. The SCADA control system allows much flexibility; the system allows the use of different controllers with little or no changes to the software.

The flexibility of this software based control system proved helpful during the testing of this system. Software changes solved several hardware problems. One major problem, easily solved in software, involved creating a duplicate set of reactor controls. The users of the system decided that they needed duplicate controls in the reactor cell for system testing. The original system specifications required only one set of controls, remotely located in the control room. Building the additional control station with hardware would have required significant time and expense. The control system software enabled easy implementation of the extra controls. The

addition of a touch screen monitor and one days programming effort, to cut and paste the main controls, provided the needed controls. The cost of the touch screen and the programming time involved are a fraction of the costs to do implement the extra controls with hardware. In fact, due to cost, this second set of controls would not have been implemented in hardware. The addition of the duplicate controls demonstrates that the EER controls can be reproduced and operated outside the control room. Taking this a step further, a network connection to the EER control system computer allows reactor control from anywhere on the network.

S u m m a r y

The EER control system demonstrates the power of a software based system. The user operates an intuitive graphical interface allowing control of this complex system with a mouse. In the EER control system, software replaces many hardware components reducing the cost and increasing the flexibility of the system. Implementing a hardware component, like an on-off valve control switch, in software allows that switch to be controlled from nearly anywhere. Software interlocks for that switch modify easily to adapt to changing experimental setups. We realized many of the advantages of expanding the role of software in this system as problems arose. The flexibility and power of software helped solve many problems and enabled solutions not possible in hardware. We expect the software to continue enabling solutions for our changing system.

The EER User Interface

Sample LabVIEW Code: Software Interlocks

This subroutine takes the Control Boolean Array In as input and outputs the Control Boolean Array Out. The routine prevents turn on of any pumps when all sample valves are closed. This prevents overpressure from 'dead heading' of the system. The Control Boolean Array Out is written to the DIO Board. Status of pump controls read from DIO Board

DATE

FILMED

5/2/94

END