Configurability in a Diagnostic Expert System for Paper Machine Dryer Sections

Amyot, J.R.; Gowing, J.D.; Wylie, R.H.; Henzell, R.

Configurability in a Diagnostic Expert System for Paper Machine Dryer Sections

J.R. Amyot,[1] J.D. Gowing,[2] R.H. Wylie,[1] R. Henzell[3]

1 National Research Council of Canada
Ottawa, Ontario, Canada K1A 0R6

2 EDS of Canada Ltd.
1615 Dundas Street East
Whitby, Ontario, Canada L1N 7S6

3 KanEng Industries, Inc.
16847 Hymus Blvd.
Kirkland, Quebec, Canada H9H 3L4

Abstract

A "Paper Drying Expert System" (PDES) prototype is being developed as a diagnostic consultant for troubleshooting dryer sections of paper machines. It is currently undergoing validation in two pulp and paper mills, scheduled to be completed in September 1994. A requirement that the PDES be configurable for many possible dryer configurations is a major consideration. This paper describes the project with a focus on how this requirement is satisfied in the design.

Keywords: diagnosis, troubleshooting, expert system, paper making.

Introduction

A "Paper Drying Expert System" (PDES) prototype is being developed as a diagnostic consultant for troubleshooting dryer sections of paper machines. A requirement that it be configurable for many possible dryer configurations is a major consideration. The purpose of this paper is to describe the project with a focus on how this requirement is satisfied in the design.

The project has five partners, i.e., three pulp and paper companies, one manufacturer, and one software developer who operate under the Software for Integrated Manufacturing Consortium (SIMCON) umbrella. The development is led by the National Research Council of Canada. The project's duration is approximately thirteen months. The scheduled completion is October 1994, after a three-month validation. The prototype is PC-based and initially off-line. Diagnosis proceeds on the basis of departures from baseline operation regarding such parameters as total steam consumption. Given that the energy required to produce steam is costly, significant economic benefits are anticipated.

The PDES has two major modes of operation: a configuration mode and an operational mode. The configuration mode provides the user with the ability to configure the PDES for the paper machine and paper grade of interest. This involves specifying the physical characteristics of the dryer section and a set of baseline operational values (including paper speed, basis weight, trim, dryer pressures, and moisture content of the paper web into and out of the dryer section). A diagnostic session is initiated by accessing the appropriate configuration and baseline data files. The diagnostic portion of the system is programmed in CLIPS, an object-oriented expert system shell from NASA, whereas the remainder is in C++. This selection of software was made on the basis of cost, functionality and ability to be integrated.

NRC 37145

In the operational mode, the user sees a graphical representation of the dryer section. He is presented with some baseline parameter values and asked to enter current values. From the baseline and current values, calculations are performed to determine if the paper machine's dryer section is operating acceptably according to two energy-related performance measures. If the performance is not acceptable, the PDES initiates a dialog with the user to selectively obtain more information with the objective of diagnosing the problem and suggesting remedies. The PDES uses the configuration data to specialize the computations and the diagnostic dialog to the specific dryer section and paper grade under investigation.

There has been much work on the integration of model based and heuristic (or fault based) diagnosis recently. Typical approaches to this problem include switching from one strategy to the other when a dead-end is encountered, using one strategy to confirm, explain or refine results obtained with the other (Lee and Kim, 1993). While practical and appropriate in many cases, such approaches do not relieve the inherent inflexibility and brittleness of the heuristic knowledge. Another approach involves coercing the heuristic knowledge into a model oriented representation, (typically called a *functional* model), so that model based reasoning techniques may be applied (Guida et al., 1993). While elegant, this approach disregards the fact that domain experts think in terms of diagnostic procedures.

The PDES represents a different approach to the integration of model and heuristic based reasoning in diagnosis. The approach is motivated mostly by the requirement that the PDES be configurable for diagnosis of the steam section of any *typical* paper machine. Succinctly, all information about the state of the paper machine is stored in a model, and all computations and tests on system state information are performed by the model. As a result, the information remaining in the diagnostic tree is independent of any particular machine configuration.

Paper Drying Process Description

The paper machine drying process consists of passing the moisture-laden paper web, coming from the press section of the paper machine, over steam-filled rotating cylinders, called *dryers* (or *can*s, or *drums)*. Dryer temperature is controlled by steam pressure. Temperature increases with pressure. Dryers are controlled in *steam groups* to maintain the same pressure (temperature) within a group. There is usually an increase in steam group temperature from the wet end to the dry end. The number of dryers in a group and the number of steam groups in a paper machine dryer section vary between machines. A felt is used with many grades of paper to hold the paper web tightly against the surface of the dryers, to improve heat transfer, and to prevent or reduce wrinkling of the paper as it dries. The dryers are usually driven by electrical motors in groups that correspond to the felts. These *drive groups* do not necessarily correspond to the steam groups or to the differential pressure groups described below.

The steam that is condensed in a dryer (in the process of giving heat to the paper) is evacuated from the dryer and collected as water in various possible configurations of separator tanks. The evacuation of condensate from dryers, via syphons, requires that a differential pressure (DP) be maintained, and it usually results in bi-phase flow of steam and condensate (S&C) involving approximately 20% steam. Therefore, in addition to being controlled in steam groups, dryers are also controlled in *DP groups* that usually correspond to one or more steam groups. The uncondensed steam may be reused in the dryers of a lower-pressure steam group, or it may be reused in the same steam group after having its pressure boosted by means of a thermocompresser. Otherwise, the excess steam is condensed in a condenser, contributing to dryer inefficiency in the form of energy losses. Furthermore, some of the dryers at the wet end are often operated at vacuum pressures (i.e., below atmospheric pressure), with resultant low surface temperatures, to avoid sheet problems. These low pressures are maintained by sending the bi-phase flow directly to a condenser (rather than to a separator tank) in a closed system, with the condensing steam creating a vacuum or negative back pressure. The condensate from the condenser goes to a vacuum receiver before being pumped away, usually back to the boilers. For PDES purposes, a *condensate drainage section* (or simply *condensate section*) can be one of several pre-defined combinations of separator tanks, condensers, and vacuum receiver.

The various configurations of dryers, tanks, condensers and control systems that are encountered in paper machine dryer sections define the configurability requirements of the PDES.

Functionality

The PDES consists of distinct functional entities that either handle or store information.

(a) User Interface

The user interface is a windows based graphical user interface (GUI) that conforms to the *look and feel* of the Microsoft Windows operating system. It provides typical GUI entities such as pull-down menus, pop-up dialog windows, and push-buttons. It can display a simple schematic of the S&C system and will have the ability to display graphs such as the TAPPI (Technical Association of the Pulp and Paper Industry) graphs of drying rate versus steam temperature. It provides help facilities in the form of Windows help files, that explain the various user interface elements and terminology. The user interface supports both British and metric units on input and output, and is available in either English or French.

(b) Configuration

The PDES can manage descriptions of various S&C systems. A user can create a new description or select and modify an existing one. The system guides the user through the specification process, providing defaults where appropriate. The configuration of the S&C system requires detailed information about the individual components and their groupings.

(c) Calculations

Numerical calculations are performed by the PDES to provide a quantitative description of the current operating state of the S&C system. The data required for the calculations consists of previously entered configuration data as well as current operating data provided by the user.

(d) Interpretation

Interpretation is required to convert the numerical description of the S&C state into a symbolic description that is meaningful for diagnostic purposes (e.g., total steam condensed = 60 000 kg/h; therefore, total steam condensed is "too high"). Interpretation involves comparisons of calculated data or current user-supplied operating data, with predetermined *baseline* values. This baseline is usually predetermined by a complete field study of the S&C system. Other sources, such as the historical data base of the system, paper machine dryer literature, and the user, may also be useful in establishing a baseline.

The baseline describes an acceptable state of the S&C system. All comparisons are made to it via user-specified thresholds. The accuracy of determination of baseline values depends on instrumentation accuracy and normal process fluctuations. The determination of threshold values is related to the baseline accuracies, and amounts to *tuning* the sensitivity of the PDES so that meaningful comparisons can be made between the current and baseline operating states. If it is determined that the S&C system is currently operating more efficiently than the baseline state (i.e., the ratio weight of water evaporated from the paper to weight of steam condensed is greater than the baseline value), then the user can reset the baseline to the current operating state. However, If the S&C system is operating below the baseline efficiency, then the result of the interpretation is: (1) total steam condensed is too high, and/or (2) U-factor is too low. Either one of these *initiating symptoms* is sufficient to trigger the diagnostic process.

(e) Diagnosis
The diagnosis is based on a fault tree approach. The knowledge base is composed of symptom and fault nodes connected by edges corresponding to the different outcomes of diagnostic tests associated with the nodes.

The diagnostic system examines the initiating symptoms and hypothesizes on possible faults or problems in the dryer section. It then begins a diagnostic session consisting of a series of system questions and user responses regarding the S&C operating conditions. Some of the observable symptoms that the user may be questioned about are: (i) the paper web is bagging around one or more dryers, (ii) the electrical current in a drive motor is above normal, (iii) a blow down valve is open, (iv) a condensate tank is flooded, and (v) a differential pressure is out of range.

The responses are used to confirm or refute the possible hypotheses. The session terminates with a list of possible faults and corresponding recommendations for their elimination. Some of the possible faults are: (1) steam supply problem, (2) wet end water removal problem, (3) control loop problems such as steam pressure, differential pressure, water level, or moisture control loops, (4) incorrect pressure or DP set points, (5) defective components such as thermocompressers, vacuum pumps, condensate pumps, and syphons, and (6) felt tension problems.

Information Handling System Design

The graphical user interface (GUI) consists of two basic components: (1) a model of the physical dryer section appearing as a schematic diagram on the screen, and (2) the visual objects, such as pull-down menus, pop-up dialog windows, and push-buttons, that allow the user to create or modify the physical model in conformity with a given dryer section.

The physical model is a collection of C++ classes that represent physical components found in dryer sections of paper machines. The main class is the *Dryer class*. One instance of the Dryer class exists for each paper machine and contains all the other sub-components in the dryer section. Since the sub-components are variable in number, they are included as lists in the Dryer object.

The dryer section of paper machines can typically be characterized by four functional components, and there can be a variable number of each per machine. These components, or building blocks, make it possible to configure the PDES for various paper machines. They also allow for grouping the atomic physical components so that analysis and diagnosis can be performed on each group, independently of exactly how the components within the group are connected. Baseline, actual and threshold values of component variables are stored as attribute values in the component objects.

(1) The first component, a *pressure control group* or *steam group*, is a set of dryer cans whose steam pressure is independently controlled. This component is used because it is necessary to: (i) know the steam pressure in each dryer can to calculate the dryer efficiency, and (ii) perform diagnosis on each pressure control loop.

(2) The next component, a *DP group*, is a set of steam groups whose DP is controlled by the same DP controller. This component is used because it is necessary to perform diagnosis on each DP control loop. Several steam groups may be connected to the same DP control valve(s). In such cases, only one of the steam groups acts as the *master* of the DP control loop. This is the steam group whose DP is actually being measured and controlled. The other *slave* steam groups have a DP equal to that of the master steam group plus the difference in steam pressure between the slave and the master.

(3) The next component, a *condensate section*, is a set of steam groups whose condensates flow to the same location and whose total condensate flow rate can be determined. The distinction is made between a DP group and a condensate section because the condensate from several steam groups which have different DP controllers may flow to the same location. A condensate section may contain a separator tank and/or condensers, but it has only one condensate flow rate. The total amount of condensate being condensed by a paper machine's dryer section is needed to determine the efficiency of the dryer section.

(4) The last component, a *drive group*, is a set of dryer cans that are rotated by the same drive mechanism.

The visual objects are sub-classes of the Microsoft Foundation Class Library (MFC). This is because the interface is designed for the Microsoft Windows environment. The MFC uses a document / view architecture. The physical model of the paper machine is stored in a *document class* object, which provides file access functionality. The visual objects are *view* and *dialog class* objects which allow the user to view and modify different areas inside the document. For example, a visual object exists for each component of the Dryer object to allow the user to modify the configuration attributes of each component. Other visual objects exist to allow the user to view and modify the current operating parameters or the baseline values of the paper machine.

The separation of the physical model from the visual objects allows the physical model to be portable to different target platforms (windowing environments) as only the visual objects would need to be converted to the new windowing system.

Diagnostic System Design

Diagnostic reasoning in the PDES can be viewed as a guided tour of the state of the paper machine. At any point during the diagnosis, there is a *current node* and a *current component*. Each diagnostic inference cycle involves a change in one or both of these. The effect is to shift the diagnostic focus to the next most *interesting* aspect of the machine's behaviour until enough knowledge about the machine state has been accumulated to isolate a specific fault. Only when the diagnostic process cannot complete some test (due to missing information) does the system interact with the user.

A clear distinction is made between model based knowledge and diagnostic knowledge.

(a) Model based knowledge. Two types of objects, *components* and *variables*, are used to model the structure and behaviour of the physical system. The principal role of the model in the PDES is to allow general knowledge about paper machines (in the form of a standard set of components) to be assembled into valid descriptions of specific machines. Variables are contained in components, and they encapsulate knowledge about the state of the system. Each variable can manage a number of distinct values. Diagnostic tests make comparisons between these values.

(b) Diagnostic knowledge. Diagnostic heuristics are represented as nodes in a diagnostic tree. Nodes contain no knowledge about the state of the system being diagnosed. The dynamic knowledge that can be associated with a node consists of the diagnostic test results, and whether or not it or its children have been visited. There are four types of node: (1) model traversal node, (2) variable test node, (3) information displaying node, and (4) node for moving about the diagnostic tree based on user input information, with no reference to the model.

The first two types of diagnostic node are of special interest here because they interact with the model.

(1) Model traversal node. Model traversal nodes provide a mechanism for identifying components of a particular type in a particular position. For each component found, the same diagnostic test is performed. Model traversal nodes are essential to the configurability of the PDES. They provide a way of specifying model independent algorithms for shifting the diagnostic focus of attention.

Example 1: Creating an instance of a *model traversal node.*

```
(make-instance (gensym m4-) of model-traversal-node
  (print-string "check for flooded tank")
  (query-string "please select a tank for flooding check")
  (child-node-spec "check tank level")
  (traversal-cmd-list supercomp supercomp subcomp)
  (traversal-class-list SteamSect CondensateSect Tank))
```

This node, named "check for flooded tank", (a) identifies all the tanks in the same condensate section as the current component, and (b) visits each one of these components, invoking the "check tank level" node for each one. Its role is to compile lists of candidate components for further diagnosis. The two lists named "traversal-command-list" and "traversal-class-list" specify a simple algorithm for moving about the model. The first step (specified by the first element of each list: "supercomp" and "SteamSect") says "get all the steam sections (i.e., steam groups) which contain the current component". The second command says "get all the condensate sections which contain (in the sense that they *receive condensate from*) these steam sections" and the last command says "get all the tanks in these condensate sections". For each one of the tanks which are identified by this algorithm, the "check tank level" node is visited.

(2) Variable test node. Variable test nodes provide the basic ability to perform comparisons of values of some variable (in a component of the model), and to use the result of the test to decide which node should be visited next.

Example 2: Creating an instance of a *variable test node.*

```
(make-instance (gensym m1-) of var-test-node
  (print-string "check tank level")
  (test-var-name Level)
  (children ``low tank level: probably OK"
_    ``normal tank level: probably OK"
       ``high tank level: check controller")
  (reference-behaviour actual-value)
  (child-behaviours threshold-value threshold-value threshold-value)
  (child-expected-results LT EQ GT))
```

In this example of a variable test node, the actual value of the *level* variable in the current component (which must be a tank) is compared to its threshold value. The result of this test determines which child node is visited next.

These examples illustrate several features of the diagnostic mechanism relevant to configurability.

(i) The model traversal node provides a general model navigation facility that makes the diagnostic knowledge independent of any particular machine configuration. It is only necessary to build a model of another paper machine dryer section (using the same basic set of components) to reconfigure the PDES to the new machine.

(ii) No system state information is stored in the diagnostic tree. The diagnostic nodes know nothing of the state of the machine being diagnosed. They make references to particular variables (in the current component), specify what tests are to be performed, and what to do (i.e., which node to visit next) upon receipt of the test results. By making sure that the diagnostic tree does not store any system state information, a diagnostic node may be used repeatedly in the course of a single diagnosis (i.e., it may be visited once for each component of the appropriate type). Furthermore, the risk of redundancy and inconsistency in the knowledge base is reduced significantly. Because the model has a well defined structure (isomorphic with the system being diagnosed), it is unlikely that components will be inadvertently duplicated. Because the tests are all performed by reference to the model, the likelihood of redundant (and hence possibly inconsistent) state information creeping into the system is reduced.

(iii) All numerical information is represented internally as intervals, and all arithmetic is performed on intervals. This provides a simple, uniform mechanism for representing both allowable ranges and point values for variables (e.g., the "threshold-value" referred to in example 2 is a range whereas the "actual-value" is typically a point value). A test compares two intervals and determines which of "less than" "equal" or "greater than" is most likely to be true (treating the intervals as representing uniform random variables). Intervals could also be used to represent error or uncertainty, although it has not been done in the current system. Finally, the PDES has been built so that the value representation, test and arithmetic computation mechanisms may all be easily replaced (and in fact the replacement of intervals with fuzzy sets is being considered).

(iv) Tests are localized. All tests defined on the model must compare two values for the same variable. In example 2, the "level" variable in each tank component is tested by comparing the actual-value to the threshold-value. The localization of tests has two effects. First, it eliminates certain types of dependency between the diagnostic nodes and the structure of the model. Second, it enforces a certain discipline on the knowledge engineer in the sense that it is impossible to compare quantities with different physical meanings; the knowledge engineer is forced to introduce a new variable into the model for any value sufficiently important as to warrant a diagnostic test.

Besides configurability, the strict partitioning of physical system knowledge from diagnostic reasoning knowledge has other advantages. The model can be viewed as rather unspecialized knowledge about the current and desired states of the dryer section of a paper machine. As such it could be used for a number of other reasoning tasks. For example it would be a simple matter to replace (or augment) the diagnostic reasoning mechanism (and diagnostic knowledge) with other reasoning mechanisms for such tasks as alarm management or process tuning. These modules could employ completely different mechanisms, sharing only their common use of the model for storage, retrieval, computation and comparison of physical quantities.

Integration of the Configuration, Analysis and Diagnosis Components

The Configuration and Analysis components have been built in C++. The Diagnosis component has been built using CLIPS 6.0. Integration consists of a set of CLIPS function calls for diagnostic model construction and to initiate or continue diagnosis. User interaction during diagnosis is managed by the C++ portion of the system. Messages from CLIPS to C++ pass through a message buffer.

Conclusion

The "Paper Drying Expert System" (PDES) project is a thirteen-month (elapsed time) and three person-year effort to produce a working prototype of a diagnostic consultant expert system for troubleshooting dryer sections of paper machines. We described the physical process, the special requirement that the

PDES be adaptable to various dryer section configurations by paper mill personnel (who are not knowledge engineers), and the design methods used to satisfy this requirement.

The original design evolved significantly during the course of the project. Functionally, it changed in scope and focus. The representation of knowledge and manner of reasoning evolved significantly during implementation. Having explored several alternatives, we are satisfied that the resulting design is theoretically sound and practical. In retrospect, the decision to use two development environments (C++ and CLIPS) was somewhat costly in terms of integration effort.

The strict partitioning of physical system knowledge from diagnostic reasoning knowledge is applicable to many other physical processes. Furthermore, the design of the diagnostic reasoning mechanism is such that it could be readily augmented with other reasoning mechanisms for such tasks as alarm management or process tuning.

References

Chittaro, L., Guida, G., Tasso, C. and E. Toppano; Functional and Teleological Knowledge in the Multimodelling Approach for Reasoning about Physical Systems: A case study in Diagnosis; IEEE Transactions on Systems, Man and Cybernetics; Vol 23, No 6 (Nov/Dec)

Lee, J.M. (1993); An Integration of Heuristic and Model-based Reasoning in Fault Diagnosis; Engineering Applications of Artificial Intelligence; Vol 6, No. 4 (pp. 345-356)

Acknowledgement

Since the focus of this paper is on software design considerations to meet the configurability requirements of the PDES, the authors are all significantly involved in this aspect of the development. The first three authors are members of the software development team; the last is the main domain expert who provided substantial input on interface design. However, we would like to express our gratitude to the people who were involved in defining the configurability requirements: Jim Futcher (consultant) and Paul Henzell (KanEng Industries, Ltd.) for their detailed technical contributions as dryer section domain experts, Rohan Jayatilaka (Stone Consolidated) and John Reinsborough (Abitibi-Price) for their general technical advice as paper makers, Oliver Vadas (Pulp and Paper Research Institute of Canada) for sharing his experience with the development and marketing of the "Pitch Expert" (an expert system for diagnosing pitch problems in Kraft mills), and David Cook (EDS of Canada) for promoting the project and providing advice towards future development and marketability.