Support for Integrated Value-Based Maintenance Planning

John C. Kunz, Yan Jin, Raymond E. Levitt, Shouh-Dauh Lin, and Paul M. Teicholz, Stanford University

THE INTELLIGENT REAL-TIME MAINTENANCE MANAGEMENT SYSTEM HELPS PROCESS-PLANT ENGINEERS AND OWNERS PERFORM VALUE-BASED PLANT MAINTENANCE. WITH IRTMM, THEY CAN INSPECT SUBSYSTEMS, IDENTIFY COMPONENT OPERATING PARAMETERS, AND REVIEW AND MAKE NOTES REGARDING COMPONENT PERFORMANCE OR OPERATIONAL AND MAINTENANCE HISTORY.

THE MAINTENANCE OBJECTIVE FOR any manufacturing or power plant is that plant systems should always be available to support plant function, without ever limiting plant production. More precisely, the cost of any maintenance activity should be less than the expected marginal value of production enabled by the planned activity. In planning maintenance to meet this time-varying objective, a plant's operating engineers and owner need to share information about current plant component status and the business situation.

Supporting this objective is difficult. Assessing the risk posed by an observed noncritical problem to future production is challenging. There are multiple goals, goals change and conflict, and indicator data are almost never completely reliable or adequate. Furthermore, the problem has multiple aspects: interpretation of observed data, problem diagnosis, repair and maintenance planning, and business evaluation of different repair and maintenance options. Finally, interpreting available engineering and business data demands good judgment, and the ability to define the value of maintenance requires a clear business policy.

As this article shows, our Intelligent Real-Time Maintenance Management (IRTMM) system helps process-plant owners perform value-based plant maintenance—not simply periodic maintenance or repairs following a breakdown, but maintenance motivated by engineering or business concerns. Implemented using object-oriented software with associated objects; displays; and systems-interface utilities, rules, and methods, the system performs three coupled functions—situation assessment, planning, and value analysis—each implemented as an independent software module. The modules share a symbolic plant model that describes plant components, their attributes, and their connectivity. IRTMM has performed successfully on test cases from power company utilities and from a large process plant.

Three-module design

The IRTMM system provides integrated subsystems for three aspects of the maintenance and repair planning problem:

- *Situation assessment.* The SA module interprets observed data as normal or abnormal, and diagnoses causes and effects of plant equipment problems. It also analyzes system performance to identify indications for condition-based maintenance.
- *Planning.* Given a set of problems to repair, provided by the SA or by a user who is considering maintenance, the

0885-9000/96/$4.00 © 1996 IEEE

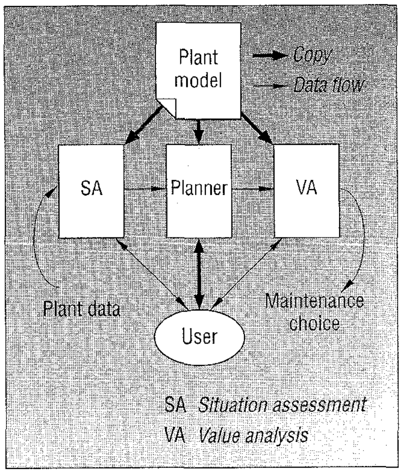

Figure 1. IRTMM architecture. Each module has a copy of the generic plant model. When it receives control from the user, a module requests data from its data source. The querying module receives and stores the requested data, specifically the attribute values of instance objects.

planner builds plans of the activities needed to repair a diagnosed problem. It can also merge related plans for the same or different components that can be performed during the same plant outage.

- *Value analysis.* The VA module identifies the monetary costs and the predicted benefits of performing every selected repair plan at different times. Using predicted power demand and plant operating and maintenance cost data, this subsystem assesses the net unit operating costs associated with performing plans at different possible times.

The IRTMM system provides interactive analyses to facilitate engineering decision making, not automate it. Given data from a data-acquisition system, the system identifies candidate problem causes and predicted effects. The user selects one or more components to analyze in more detail. After reviewing the system-generated plans and value analysis, the user selects the one or more components for repair, indicating both the desired repair activity and the planned repair time, after considering the system-generated options.

Designed to reside on a computer network, IRTMM can receive component status information from an on-line data-acquisition system or any available diagnostic expert systems; staff- and equipment-availability information from a computerized maintenance management system (CMMS); and projected product demand, cost, and selling price data from a business database. Users can log recommended work in the CMMS.

Shared plant model

At the heart of the system is a shared symbolic plant model, used by the three IRTMM modules. The model explicitly represents the form, function, and behavior of the plant systems, components, and processes. *Form* describes the layout and part composition of components and systems. If the application were extended to support other purposes, the form model would also represent component features, dimensions and tolerances, and materials. *Function* describes design intent for the component or system—for example, for pumping—and specifies methods to compute the simulated values of component output parameters, given values of input parameters. *Behavior* describes the possible, measured, and simulated values of parameters, including states (operating, startup, or failed), engineering parameters such as vibration and temperature, and relationships.

IRTMM processing control involves shifting the system's focus of attention from the SA to the planner to the VA modules (see Figure 1). Each module requests data from its information source.

Figure 2 shows the most important entities represented in the shared plant model; characterizes their roles as form, function, and behavior; and identifies the module that uses each model constituent. The different IRTMM modules use the model constituents, as shown by x's in the rightmost three columns.

The model explicitly represents each parameter. Each has a value, which is often numeric. Numeric parameters also have a state. A parameter's state attribute value is normally high, normal, low, or off. The SA and planner use the model as a source of the plant's components and their connectivity, engineering functions, and possible behaviors.

Situation assessment

The SA system diagnoses plant equipment problems. Given specialized input data from instrumentation (pressures or flow rates, for example) and possibly from specialized expert diagnostic systems (such as vibration analysis), the SA provides a systems diagnosis. It identifies potential root causes and effects of component problems, where some causes and effects might be in the component with a problem and others in subcomponents or connected systems.

The SA uses a combination of methods to perform this assessment: model-based diagnosis (MBD) to identify the details of a large class of possible problems, heuristic classification to identify the presence of a set of idiosyncratic problems, and case-based reasoning (CBR) to compare observed data with previously identified cases identified by the MBD technique. The SA uses the shared symbolic plant model of the plant systems, components, and parameters. Users interact with the system using interactive process and instrumentation diagrams (P&IDs) for different selected subsystems, identify component operating parameters, and review and make notes regarding component performance or operational and maintenance history.

The SA provides a systematic monitoring and component diagnosis capability for facility equipment and systems. Henny Sipma implemented the first versions of the SA in the BB1 software environment.[1]

SA purposes. The SA includes the following capabilities:

- *Check reported data for consistency.* The SA reports alarm conditions for data that is out of the expected range or inconsistent with other measured data. This process checks data value with respect to context-dependent limits to classify data as normal, high, low, or artifact. Plant-monitoring systems now largely perform the out-of-expected-range test, but the SA includes it for those cases in which a plant-monitoring system does not do the check.
- *Hypothesize possible component faults.* For processes that are out of statistical control, the SA identifies candidate causes and reports evidence for and against hypothesized faults. It reports causes, effects, supporting data, missing data, and recommended actions.
- *Show a shared, annotated P&ID to plant staff.* Any IRTMM user can bring up a subsystem P&ID and view the following information provided by any SA user: measured data currently in the plant model system (such as temperatures), staff comments (for example, to let one user store the fact that a component was repaired and is now back on line and let another user discover the fact), and inferred data (such as when the computer concludes that a component is unreliable and should not be used except in emergencies).
- *Use a uniform environment (the shared*

plant model) to describe the current facility, including components, their roles, and their connectivity. The plant model stores measured data values, fault possibilities, and component history as concluded by the SA or reported interactively by staff.

The SA system operates in either the *periodic* or *demand* mode. In the periodic mode, the system queries the monitoring system every period (for example, every hour) for the current status of all measured parameters, assesses each component's diagnostic status, and displays a summary status assessment of all monitored subsystems, including trends when available. Plant maintenance supervisors, the principal periodic users, review the situation assessment at least once a week and prepare work orders as appropriate. The SA operates as a maintenance advisor, not a control system.

In the demand mode, when initiated by a user or a monitored event reported by the data-acquisition system, the SA queries the monitoring system for the current status of components in a selected subsystem, assesses the diagnostic status of that subsystem, and displays analysis results on an annotated subsystem P&ID. When requested by a user, it queries or updates the work-management system history file. Individual plant maintenance staff and management, the principal demand users, review outstanding work orders; review component status, history, trends, and comments; plan maintenance activities; and log comments about their actions, observations, and conclusions.

Form		SA	Planner	VA
	Components (pumps, valves)			
	- id	x	x	x
	- Connectivity	x		
	- Operating, failure modes	x		
	- Parts		x	
	- Functional process role	x		
	- Parameters	x		
Function				
	Activities: Methods to			
	- Find early start, late finish		x	
	- Build, merge activities		x	
	Components: Methods to			
	- Diagnose problems	x		
	- Isolate (clear) a component		x	
	- Analyze choices, chances, value			x
	Processes (pumping): Method to			
	- Propagate behavior	x		
	Tasks (repair pump): Method to			
	- Estimate costs, benefits			
Behavior				
	Actions: elaboration, refinement		x	
	- Chances			x
	Activities			
	- Start time, finish time, duration		x	x
	- Successors, predecessors		x	
	Components (pumps, valves)			
	- Hypothesized faults	x		
	- Mean time to failure, variance			x
	- Normal, expected repair costs, durations			x
	- Normal repair duration			x
	- Selected faults to repair	x	x	x
	- Status	x		
	Parameters (outlet temperature)			
	- Expected, simulated values	x		
	Plants (Unit 1)			
	- Demand, price, cost profiles			x
	Resources: Actions they can perform			x

Figure 2. Entities defined in the shared plant model. Shading indicates data items shared among two or more modules. The plant model represents numerous classes of plant components, including filters, headers, heat exchangers, instruments, pipes, process equipment, pumps, and valves.

SA reasoning. Knowledge systems now routinely do diagnostic reasoning using MBD, heuristic classification, and CBR. The SA uses a combination of these.

Model-based diagnosis. MBD involves qualitative simulation of system behavior.[2] First, a user (or an algorithm) sets up the plant model, selecting components to represent in the system, their connectivity, their states (such as on, off, open, or closed), and their assumed behaviors (for example, as-designed or leaking). Next, the user injects a change into a system model—for example, by closing a valve or starting a leak—and runs a simulation to propagate the change through the system to determine the complete system behavior, given the assumed system parameter states. Finally, the user compares the simulated and observed data.

With consistent data, users assume that the physical system and model states are consistent. If the observed and simulated data are inconsistent, users vary their assumptions about the model until simulated and observed data become consistent. Any abnormalities in the model state are sufficient to explain abnormalities in the observed system.

The SA uses MBD to analyze the behavior of a process plant's process cooling water (PCW) system. The method generally applies to situations in which relatively simple rules describe the propagation of behavior from one system component to another.

Heuristic classification. This method is the basis for classic expert systems.[3] It abstracts evidence in the form of measured data and relates it to a predefined potential problem. Heuristic classification matches the problem with a solution and refines the solution. The SA uses this method to analyze situations having idiosyncratic causal behavior—for example, the causes and effects of bearing failure and shaft imbalance.

Case-based reasoning. CBR is the basis for some diagnostic systems and many recent help-desk applications.[4] An expert creates a set of cases, each including descriptions of a situation (a case) and an associated statement of a cause and suggested repair. Thus, the CBR method aggregates the heuristic classification steps. Diagnosis involves simple matching of observed data with the data of each case. This method reports relevant cases

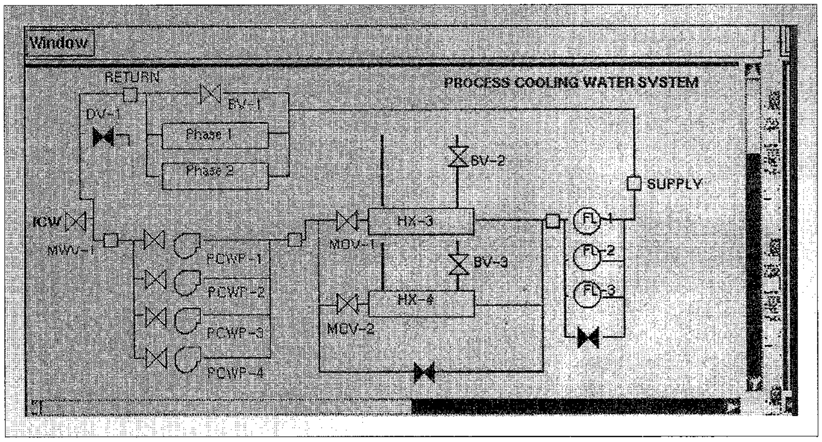

Figure 3. Active P&ID for a subsystem. A user gets component status information by selecting any component icon with a mouse. The SA infers the candidate causes and potential effects of any observed or simulated parameter abnormality. By selecting any icon that shows in red because it is abnormal, the user can view the information listed in Figure 4.

as appearing similar to the observed situation.

Integrating MBD and CBR. MBD uses no notion of fault. Systems simply have behavior. Observed and simulated behaviors either do or do not compare. Separate from the MBD technique, a user might choose to modify a system (physical or model) to change its behavior. In the SA, the MBD technique generates a set of cases. The user then annotates each case with a recommended change and an expected outcome if the change is applied. The SA diagnostic procedure then matches observed data with case data and reports the cases that best match the observation, the evidence for and against each case, the recommended action, and the expected outcome.

Reasoning procedures. The SA model performs the following reasoning procedures:

- *Set up a test case.* The SA lets a user introduce a problem into the model and record both a description of the problem and the appropriate repair. This setup method then identifies consequences of the introduced problem (by invoking the `Propagate!` method) and collects these secondary parameter values into the case record (by invoking the `Snapshot!` method).
- *Propagate a component's qualitative behavior to its downstream components* (`Propagate_Behavior!`). For example, the SA checks the component-inlets state parameter and determines the component discharge state parameters. For a normally functioning component, the propagation method sets a component's discharge state parameters to be the same as the states of corresponding inlet parameters—for example, high, normal, or low temperature for the inlet gets propagated to high, normal, or low outlet temperature. Heating components, for instance, set the discharge temperature to high if fluid inlet is high.
- *Record subsystem snapshot values* (`Snapshot!`). The SA records a user's parameter state assignments and the associated states of all abnormal parameters after behavior propagation. The user can (and should) check that the case makes sense and also annotate the case with suggested repair actions and the expected result if the suggested action is taken. The SA adds the snapshot to a library of cases.
- *Perform system situation assessment* (`Diagnose!`). The SA compares a set of observed parameter data with the cases in the library and reports cases having values closest to the observed set of parameter status observations.

We developed and tested the SA with test cases from industrial plant operators: a bent shaft in the main boiler feedwater pump, leaks in a boiler and a process component, an inadvertently closed valve that reduced chilling capacity of a chilled water system, and blockage of a filter in a process flow system.

Figure 3 shows part of the SA module's graphic user interface. The interface is an interactive P&ID that records and reports user notes, component history, and measures of success; shows parameters of any selected component; and invokes a diagnostic routine to diagnose problems that can cause any observed parameter abnormality.

SA testing. Figure 4 shows two test cases for the SA module. In Test Case 1, the system identifies two possible causes for the observed low PCW discharge (return) pressure. Compared to the selected explanation, the other explanation has less significant evidence in favor and more evidence against it. Thus, between the possible causes shown in the figure, the first is more likely and should be confirmed first.

In Text Case 2, the system has cases that allow it to identify two possible causes for the observed high process-supply water temperature. Compared to the selected explanation, the other explanation has less significant evidence in favor and more evidence against it. Thus, between the possible causes shown in the figure, the first is more likely and should be confirmed first.

Planner

The engineering planning process involves generating work procedures that workers and machines should follow to achieve an engineering goal and that planners need to estimate costs and manage projects. From a planning viewpoint, the engineering planning problem is likely to be complex, because it is highly contextual and might involve many types of objects, actions, and resources.

Early AI planners showed that computers could identify plan activities and infer activity precedence.[5] While such early Strips-style planners evolved by the mid 1970s, they remain ill-suited to most engineering domains because of the limited activity representation these planners adopted.[6] General-purpose planners in the tradition of Strips (Stanford Research Institute Problem Solver) still have not found broad use for generating realistic engineering plans. Conversely, narrowly scoped expert planning systems work for specific domains but have little applicability to even slightly different domains.

The engineering planning research we describe here attempts to address the shortcomings of overly general or overly specific planning approaches by modeling the behavior of plan elements. Oarplan (Object-Action Resource Planner), a model-based planner, defines activities in an engineering plan by their constituent objects, actions, and resources.[7] That is, an activity specifies an object-action-resources set, as well as traditional attributes of times (start and end) and relationships (for example, successors). It generates a plan by reasoning about objects, actions, and resources of a specific engineering domain.

Oarplan uses generic models of objects (pumps and valves) contained in the shared plant model. In addition, it uses models of actions (for example, close or remove) and resources (plumbers and cranes).

Planning in Oarplan involves activity generation and activity ordering. Oarplan planning is hierarchical because it recursively generates activities at lower planning levels by elaborating those at higher levels, based on object-action-resource models. Activity ordering proceeds after all required activities are generated and is based on satisfying constraints represented in the form of relations among components and methods. A final plan results after Oarplan performs all elaboration and examines and satisfies all constraints.

Planner purposes. The planner can clear a component, generate activities based on elaboration and refinement of a top-level activity, order planned activities, and merge plans that are mergeable.

Planner reasoning. The planner uses a model-based approach to engineering planning. Planning explicitly considers the form and function of plant components. The clearance procedure identifies fluid inlet valves to close and discharge and drain valves to open for isolating a component and safeguarding workers. If a valve is disabled, the procedure searches upstream from the component for other valves to close or open.

Activity generation builds on planning knowledge of how actions elaborate. Actions can have relationships that specify how they elaborate into more detailed actions; objects can have relationships that specify how they decompose into parts. For example, the activity-generation procedure recursively generates detailed subactivities to elaborate a given top-level activity (see Figure 5).

In this example, the diagnosis was that the pump required maintenance to repair a bent shaft, and a planner chose to send the shaft out for repair (rather than replace the pump or repair the shaft in-house.) In the model, `UNIT_3_MBFP_Shaft` is part of `UNIT_3_MBFP_Pump`, and the repair action elaborates into five more detailed actions. The generation procedure takes a given top-level task, `(Repair UNIT_3_MBFP_Pump Some-Resources)` and elaborates an activity (Figure 5). By exploiting the component composition constraints with a relatively simple reasoning algorithm, the planner hierarchically generates required activities for achieving a given project goal. The final plan includes the leaf activities that cannot be expanded further, shown as boxes in the figure.

Activity ordering now has a simple procedure that places and orders these activities in a way that constructively satisfies the parallel or sequential elaboration constraints. The plan-merge procedure will offer to merge two plans if they have activities that work on components within the same clearance boundary or if some activities are shared—if they have the same object, action, and resources. Merging plans is desirable when users can perform a merged plan faster than several independent plans and when they can perform opportunistic maintenance relatively inexpensively while also performing another required maintenance activity.

Planner testing. The SA reported that the main boiler feedwater pump (MBFP) required maintenance because vibration indicated a bent shaft problem. Figure 5 shows the generated repair activity tree. The plan itself is a linearized list of boxed activities.

Test case 1
The process plant monitoring system reported a set of alarms, including a low alarm for the process return water pressure, a process parameter of the process cooling water (PCW) system.

Problem: Low Phase-1 PCW return pressure
Possible causes:
- Water leak in Phase-1 process equipment;
- Filter degraded

For water leak in Phase-1 process equipment,

Evidence +:
Phase-1 (PCW) discharge pressure low
HX3 (Heat-exchanger 3) discharge flow rate low
HX3 (Heat-exchanger 3) discharge temperature low
HX4 (Heat-exchanger 4) discharge flow rate low
HX4 (Heat-exchanger 4) discharge temperature low
Supply header discharge temperature low
BV1 (balance valve) position partially closed

Evidence –: None

Recommended action: Walk down the Phase-1 area of the plant, confirm leak location, and repair.

Expected result of action: Repair leak

Test case 2
For the PCW system, the plant monitoring system reported a high alarm for the process supply water temperature. The SA analysis of the test case is summarized below.

Problem: High process supply header water temperature
Possible causes:
- MOV 1 closed (hand-operated HX3 inlet valve)
- MOV 2 closed (hand-operated HX4 inlet valve)

For MOV 1 closed,
Evidence +:
High HX4 discharge temperature
High supply header discharge temperature
High return header discharge temperature
HX3 outlet flow off
High HX4 outlet flow

Evidence –: None
Recommended action: Check MOV 1 valve position and close if improperly opened.
Expected result of action: Process supply header water temperature should return to normal within 30 minutes.

Figure 4. Two SA module test cases.

Value analysis

Decision analysis recognizes that making choices means taking chances. Given a choice of time to maintain a component, the failing component might break at any time prior to scheduled maintenance, with likelihood given by a prescribed probability distribution, or it might survive to the scheduled maintenance. The VA uses a decision-theoretic approach—decision analysis—to analyze the expected

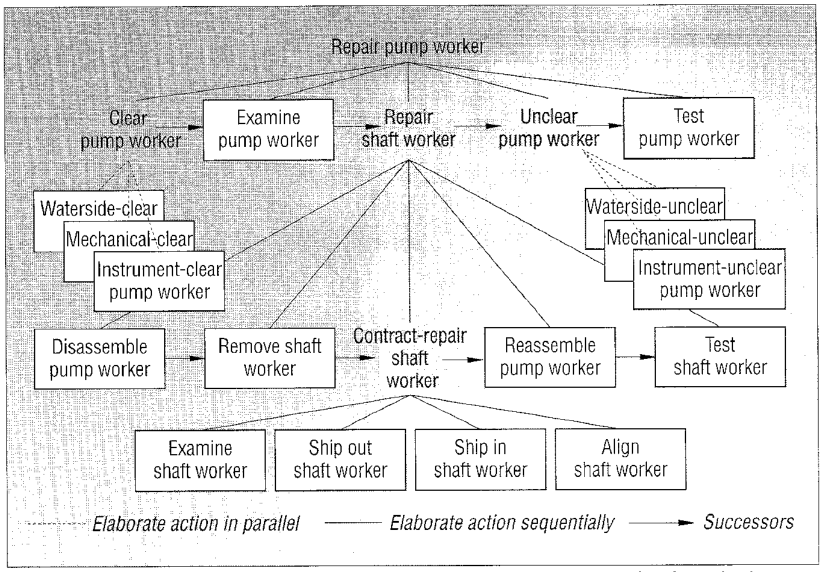

Figure 5. Oarplan plan activity tree. Oarplan builds activities with object-action-resources triples. The top-level repair activity elaborates into a set of more detailed activities, based on the composition of the pump and the elaboration of the repair action. Similarly, the repair shaft activity elaborates into a set of more detailed activities based on the elaboration of the repair action. Boxes indicate activities that are included in the final plan.

value of each choice on the basis of each possible chance's likelihood and the value of its outcomes.

Figure 6 shows how decision analysis computes a choice's expected value as the sum of the values of the chance outcomes, weighted by the probability of their occurrence. For a study period that is q units long, there are $q + 1$ choice alternatives: schedule repair at the ith hour, $i = 1, \ldots, q$; or defer repair until after the end of the study period. Each choice has some number of chance outcomes. For the ith-hour repair alternative, for example, we can represent the possible chance outcomes by the tree shown in Figure 6.

The size of the decision space depends on the size of both the study period and the data's grain size: a seven-year study period must consider seven outcome possibilities if the problem assumes an annual data grain size. It considers 84 outcomes if the problem assumes data precision of one month. Each decision's impact is measured by the expected value of the objective function, weighted by the probability of occurrence associated with each outcome. The "best" decision is thus the one that has the optimal expected value.

VA purposes. For each maintenance plan produced by the planner and selected by the user, the VA identifies several timing choices: perform the maintenance or repair as soon as possible, at the next period of low demand that is long enough to do the activity, or at the next scheduled outage that is long enough; or defer maintenance until after the end of the study period. Then it computes the expected costs and net value for a given repair plan at each time option.

The VA considers only one failing component at a time, so only one maintenance activity at a time passes from the planner for study by the VA. The VA assumes that the failing component is critical to the system. If the component fails unexpectedly or is under planned repair, the VA assumes that the system goes down. If repaired or maintained, a component will not fail again within the study period by the same mechanism. The only random aspect of the analysis is the failing component's life. Unplanned repair begins immediately following component failure, but the VA is built with the assumption that the time to perform unplanned repair includes both a wait time to assemble parts and crew plus actual repair time.

Components deteriorate only during operation, so when a component is idle, its performance does not deteriorate. Once repaired, a component returns to an as-good condition and will not fail again within the study period. The VA assumes that the component failure always occurs at the beginning of each hour. For example, the component failing at the kth hour means that the breakdown happens at the beginning of the kth hour. To simplify the analysis, the VA computes the benefit of a choice relative to a reference value, called the *baseline*. The simplest baseline assumes performing no maintenance activity during a study period and no unexpected failure.

The VA assumes that a failing component's life distribution is Weibull, the most widely used parametric family of failure distributions. The Weibull distribution takes the form:

$P\{T < t\} = \mathrm{F}(t) = 1 - e^{\lambda t^{\alpha}}$
= Probability that a component fails at time T

where

T = failing component's life
λ = scale parameter
α = shape parameter

The VA gets the estimate's mean-time-to-failure and standard deviation from the model and converts these parameters to the two Weibull distribution parameters.

VA reasoning. For this discussion, we take the time unit to be one hour and the study period to be n hours. The benefit of a chance B is a value associated with the chance outcome computed from the following three parts:

- cost savings S of operational mode relative to the baseline operational costs, during a time period;
- cost of downtime C_{downtime} due to buying replacement power from other sources during a time period. This cost will vary depending on duration—whether the repair is planned or unplanned.
- cost of repair C_{repair}, either planned or unplanned, during a time period.

We can compute a chance occurrence's expected benefit by summing the costs, weighted by the probability of failure at each time period, to determine the expected net benefit of a choice:

Benefit of a change

$$= \sum_{\text{time from now} = 1}^{\text{end of study period}} (S - C_{\text{downtime}} - C_{\text{repair}})$$

Suppose the component fails at the kth hour. The total cost of failure would be:

Total cost if the component fails at the kth hour equals costs due to deterioration from the first to the $(k-1)$th hour (for example, the cost of deration, or running the component at less than its design rating capacting, and marginal costs of operating an aged component—cost of production loss)
+ costs due to shutdown (for repair) from the kth to the $(k + t_r-1)$th hour, including the cost of replacement power while the plant is down;

+ costs of unplanned repair labor and materials

where t_r = repair time.

The total expected repair cost at the kth hour also considers the time value of money.

The user provides the following input to the VA module:

- Baseline case with predicted cost outcomes—typically an option to continue current operational mode throughout the study period.
- Choices that can be made immediately—modes of operation and corresponding cost savings.
- Chances—failure modes and corresponding relative probabilities following each choice.
- Failing component and its lifetime probability of failure distribution.
- Demand prediction over a study period.
- Durations and costs for planned and unplanned maintenance.
- Discount rate of money.
- Study period duration.
- Time options when the maintenance plan could be performed.

Using this information, the VA generates a decision tree with each branch representing a possible chance outcome. It then computes backwards to obtain the expected benefit for each choice option compared to the baseline option. For each choice, the returned values from VA include probability of survival to the start of maintenance; expected cost savings, replacement power cost, repair cost, and benefit; best- and worst-case benefits; and break-even probability between two selected timing options.

VA testing. While investigating excessive fan motor vibration, an engineer recommends that a component needs its bearing oil changed. If the oil is not changed, the bearings might fail and bring down the plant. Arbitrarily, the engineer chooses to analyze the economic value of maintenance options for a one-week study period. The engineer identifies the following choices: (1) change oil off line as soon as possible, bringing down the plant during the procedure; (2) change oil on line ASAP, keeping the plant operating during the procedure; or (3) defer to the end of the study period.

Each choice has a set of chance occurrences: (1) unexpected bearing failure prior to any maintenance; (2) oil spill during oil change, requiring immediate plant shutdown until the problem is corrected; or (3) survival of the bearing until the next planned outage (after the end of the study period).

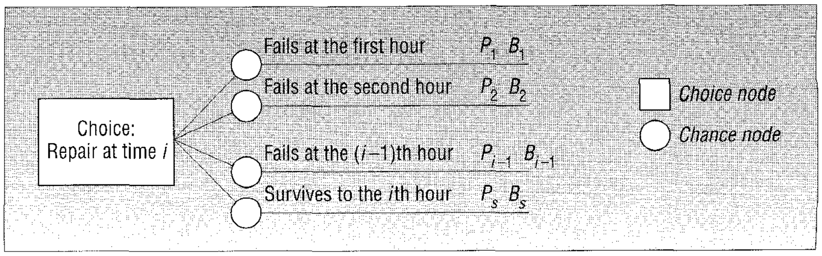

Figure 6. A choice-chance tree. The expected benefit of a choice, such as the choice to repair at time i, is the probability weighted sum of the benefits of the chances. P_j = probability of failure during the jth hour; B_j = net benefit of repair and revenue loss if the component fails in the jth hour, $j < i$; and $\Sigma P_j B_j$ = expected benefit of choice of repairing at the ith hour. Summation takes place over each of the chance occurrences: $1 \le j \le i$.

The engineer makes a number of assumptions:

- Oil spill and unexpected bearing failure are independent.
- Oil spill can only occur at the beginning of an on-line oil change, assumed to be planned for the first period, with a probability of P_{spill}.
- If oil spill occurs, engineers must shut down the system and change oil completely.
- After oil is changed, no failure occurs within the study period.
- The material costs for changing oil off line and changing oil on line are the same.
- The time to change oil on line is short enough so that no bearing failure will occur during the work.

The time unit for this test case was one hour, with a study period of one week and 168 periods to analyze during the study period.

We define the baseline case as the defer case: schedule oil change after the end of the study period. The baseline has no downtime cost, only the cost of the off-line oil change. The benefit for the baseline is zero.

Figure 7 shows the best alternative and associated expected benefit, relative to the ASAP/off-line alternative, as a function of the probability of bearing failure and the probability of an oil spill. For example, when the oil-spill probability is high and the bearing-failure probability is low, the best strategy is to defer the off-line oil change. When both probabilities are high—the riskiest situation—engineers should immediately shut the system down and change the oil off line. Otherwise, the best strategy is to change the oil on line. Figure 7 shows the optimal decision for the bearing-failure and oil-spill probabilities. It also lists each decision's expected benefit (K$) relative to the second best's expected benefit.

Discussion

This section comments on the use of the model, each of the modules individually, and then the system as an integrated whole.

Integrated system. An important part of the IRTMM system's power comes from the model's engineering content, specifically the representation of plant function, form, and behavior in the shared plant model, and from the reasoning about these issues by the applications. As Figure 2 suggests, the function, form, and behavior describes common engineering knowledge about a system. Thus, it is readily available from knowledgeable plant engineers. It is also very general, applicable to a broad class of process plants. For example, we initially implemented SA for a power plant and later for a process plant's water-handling subsystem, and we needed to make only minor extensions to the generic plant model's functional definitions.

Similarly, we have used the planner for both maintenance planning and building-construction planning, with only minor extensions to define actions for the new applications area. The form model represents relatively standard content of a product model, such as would be built using the ISO STEP standard. However, an important part of the total system power comes from the behavioral methods defined in the function models. While such methods are standard capabilities of object-oriented technology, they are outside the normal capabilities of STEP-style product models.

As Figure 1 shows, each application module has a copy of the generic plant model. Thus, each module shares the complete plant model ontology. In fact, as Figure 2 illustrates, two or more modules share only very limited form and behavior data items, and no functional methods are shared among modules. Rather than sharing the complete model ontology, it would have been sufficient for each module to share only a top-level ontology—the classes

	P_{spill}									
$P_{bearing\ failure}$	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
0.1	0.1	11.7	23.5	35.3	47.1	58.9	70.7	82.5	94.3	101.1
0.2	12.4	0.6	11.2	23	34.8	46.6	58.4	70.2	82	88.8
0.3	25.1	13.3	1.5	10.3	22.1	33.9	45.7	57.5	69.3	76.1
0.4	38.4	26.5	14.74	2.9	8.9	20.7	32.5	44.3	56.1	62.9
0.5	52.3	40.5	28.7	16.9	5.1	6.8	18.6	30.4	42.2	49
0.6	67.1	55.3	43.5	31.7	19.9	8.1	3.7	15.5	27.3	34.1
0.7	83.1	71.3	59.5	47.7	35.9	24.1	12.3	0.5	11.3	18.1
0.8	101	89.2	77.4	65.6	53.8	42	30.2	18.4	6.6	0.2
0.9	101.2	89.4	77.6	65.8	54	42.2	30.4	18.6	6.8	5
1.0	101.2	89.4	77.6	65.8	54	42.2	30.4	18.6	6.8	5

Defer On-line ASAP

Figure 7. Best alternative and associated expected benefit, relative to the ASAP/off-line alternative, as a function of bearing-failure and oil-spill probabilities. The patterns indicate three preferred repair alternatives. Numbers in boxes show relative expected benefit (K$) of the preferred alternative over the second-best alternative.

in which mutually referenced slots are defined.

One knowledge-sharing architecture is to share a complete ontology with the entire form, function, and behavior of every class and instance object. The simplest alternative is to share minimal data—values of selected slots of a few instances—with each module maintaining its own ontology of classes to describe generic form, function, and behavior in support of its own perspective. The current IRTMM implementation uses the former alternative, but we conclude in retrospect that the latter would have been adequate.

In general, two different applications can use different representations of the same information. An integration architecture needs to support appropriate transformation of the data generated by one application into the perspective and actual data structure expected by a successor application. For example, in IRTMM, the planner produces activity start and end times. The VA needs activity durations; the planner transforms these times into durations. Either module could report the same information under two names or make any other appropriate parametric transformation. In an alternative integration architecture, an external agent could do the transformation after one module registers that it can produce particular data and another module registers its interest in the data.[8]

We are reluctant to generalize our limited results to some of the major efforts to share large ontologies, such as STEP.[9] However, our work does let us conclude that it is probably very worthwhile to identify the engineering problem to be solved as clearly as possible—that is, our formulation of the maintenance problem—and then assess the information that needs to be shared and how different integration architectures can support the defined need. We have found that the sharing mechanism can easily become far more complex than necessary to support the engineering objective.

Situation assessment. The SA hybrid MBD, heuristic classification, and CBR approach offers a number of benefits; it

- Uses knowledge of plant design, specifically of the form, function, and behavior of the plant, because the MBD approach reasons about component definitions, processes, and topologies explicitly.
- Uses diagnostic knowledge of plant operators, because operators can add important cases at will, using the CBR technique or heuristic classification rules, and because they can annotate all cases with the proper engineering response to the situation, again using the CBR approach.
- Identifies implications of situations, because the MBD technique predicts behavior.
- Does not require staff to identify all failure modes, because the MBD technique can be invoked exhaustively to find the behaviors that emerge from various input conditions.

On-line data-acquisition systems are finding broad use in the process industry. While engineers always want more sensory data points, the monitoring systems often have valuable information that is not always easy to use for maintenance management. The SA supports maintenance management by integrating and interpreting the available data from multiple sensors and multiple components.

The SA is designed to assist with systems diagnosis, not perform specialized diagnosis of individual isolated components. Potentially, it can accept input from specialized expert system diagnostic routines, such as a vibration expert diagnostic system. The VA analyzes an entire system to identify the potential system-level causes and effects of problems with individual components. The SA does not now do any data trending or quantitative prediction of the severity or timing of predicted degradation.

Planner. Oarplan represents the object, action, and resources of planned activities. Both planning and merging plans consider these entities and their attributes. The number of such entities in the model is far fewer than the number of activities that would otherwise need to be represented—the sum of the numbers of objects, actions, and resources is far less than the product of those three numbers.

In addition, the descriptions of objects, actions, and resources and the topological and compositional relationships among objects describe fundamental knowledge about designed systems. Reasoning from such engineering principles gives generality to the planning procedure.

Also, the action elaboration and refinement lets the planning become specialized to support a particular planning purpose. Our experience indicates that Oarplan shows both power and generality not previously found in AI planning systems.

Oarplan does hierarchical planning. However, it is good engineering practice to aggregate some activities in a nonhierarchical order—for example, to do resource-leveling in support of scheduling and to avoid undoing useful setup activities. Rather than attempt to compromise the conceptual simplicity of hierarchical planning and consider special cases during the initial plan generation, the IRTMM planner uses a second pass to merge activities and to introduce efficiencies and remove conflicts that are possible with hierarchical planning. The first pass gives the generality to the planner; the second pass allows it to accommodate specialized engineering details.

The IRTMM planner's effectiveness is limited by the quality of the plant and action models on which it works. As with all model-based applications, the quality of the planning model limits the quality of the generated plans. Thus, the model builder controls both the object abstraction at which the planner works and the abstraction of the action details generated during plan elaboration.

Value analysis. The fan bearing problem fits the VA framework naturally. As shown by this and several other power industry test cases, the VA handles many power plant maintenance problems very well. Given the power demand prediction, monetary information, and component failure modes, the VA can provide maintenance choices regarding timing, probability of survival, and the break-even analysis.

However, the calculation rests on the simplifying assumption that the maintenance returns the failing component to a new condition such that it will not fail again by the same mechanism within the study period. That is, the maintenance is assumed to be perfect. This assumption is plausible for those (frequent) cases when the study period is short compared with the expected life of the repaired component. If it is not short, the VA will return an overly optimistic result, because it will underestimate the effects of recurrent failure.

The perfect repair assumption also implies that the VA does not accurately assess benefits of partial repairs. In both recurring failure and partial repair cases, the VA produces an overly optimistic net benefit because it normally ignores the cost of recurring repairs, the expected cost of failure introduced by an initial repair, and the cost of continued degraded performance following a partial repair.

The VA system has a forms-based graphic user interface that allows users to input required information. Thus, the VA user requires a knowledge of plant operational design and current status details. It is important to understand the concepts of decision analysis, but the system protects the user from both its theory and the details of tree generation and expected-value computation.

Implications for process plant operations. Successful implementation of integrated maintenance systems such as IRTMM can favorably affect the following risk factors for plant downtime:

- *Procedure errors*—reduce their incidence, so that operators can predict the effects of planned operations, which represent direct risk of downtime.
- *False alarms (proper response is to ignore)*—reduce their incidence, because they waste time, indicate a process that is not well managed, divert attention from real problems, and contribute to staff insensitivity to true alarms.
- *Phone calls to technicians and engineers at home*—reduce their incidence, because they indicate that staff lacks the access to knowledge or data to perform a job properly.
- *Phone trouble calls to technicians and engineers in the plant*—reduce their incidence, because they indicate that staff lacks the access to knowledge or data to perform a job properly.
- *Alarms that do not indicate production impact (proper response is to fix a component, but production is not affected)*—increase their incidence, because they indicate effective predictive maintenance.
- *Rework*—reduce its incidence, because it detracts from effective maintenance and is costly.
- *Timeliness*—reduce the time from the decision to perform maintenance or repair to the time that repair is successfully completed, because delayed definitive repair simply adds breakdown risk.
- *Personnel and equipment use*—by reducing equipment downtime and helping technicians to do the work right the first time with fewer calls for help, the system should help improve personnel and equipment use.

We expect the following collateral, qualitative benefits to facility operation following implementation of a system with the capabilities of IRTMM: improved continuous training in details of as-built design, diagnosis, planning, and evaluation procedures; reduced stress for engineers and technicians, especially on short-staffed 12-hour shifts; qualitatively decreased time to design facility retrofit projects; and startup and help operators to bring new plant facilities to peak production capacity faster. As the "Desktop engineering" sidebar describes, the IRTMM system also affords great ease of use.

Plant engineers have suggested some possible extensions. Imminent faults should be predicted on the basis of statistical process-control trend analysis and potential failure modes. Also, P&ID objects should be able to highlight themselves in layout diagrams, and components in layout diagrams should be able to highlight themselves in the P&ID. Finally, P&ID objects should be able to show component schematics and disassembly sketches.

SINCE COMPLETING THE ORIGINAL IRTMM research, we have done several studies of the potential feasibility of implementing some of the IRTMM ideas.

We did major detailed studies of current maintenance management practices and methods in two industrial and governmental agencies. Both organizations had plans to implement a CMMS, a database system that would collect plant operational data from a distributed control system and make it available in a database. The CMMS will provide the operational plant data needed for the SA operation, and it could hold generic plan and current demand and cost data for the Planner and VA. Thus, current CMMSs provide an enabling technology for IRTMM, and IRTMM provides services that are not included in any CMMS considered by our clients.

The IRTMM system requires an accurate symbolic plant model. In one case, a plant had reasonably accurate P&IDs, but because they were available only in CAD vector form, the IRTMM system could not interpret them as a symbolic model. In the other case, the owner did not have accurate P&IDs. Creating an accurate P&ID would have involved a significant cost. Some of us have since started a promising research project to build symbolic models from P&IDs in CAD format. Finally, most plant engineers and technicians were highly enthusiastic about adding IRTMM capabilities to their technology-support infrastructure.

References

1. B. Hayes-Roth, "A Blackboard Architecture for Control," *Artificial Intelligence*, Vol. 26, No. 3, July 1985, pp. 251–321.
2. J. de Kleer and B.C. Williams, "Diagnosis with Behavioral Modes," *Proc. Int'l J. Conf. AI (IJCAI)*, AAAI Press, Menlo Park, Calif., 1989, pp. 1324–1330.
3. W.J. Clancey, "Heuristic Classification," *Artificial Intelligence*, Vol. 27, No. 3, Dec. 1985, pp. 289–350.

Desktop engineering

The IRTMM system is an example of a desktop engineering system in that a single user can review and manage diverse issues concerning plant operations, including some aspects of engineering (using a diagnostic subsystem), management (through the planning subsystem), and business (in the value analysis subsystem). The user can do what-if analyses considering changes in engineering, management, and business operating assumptions. What-if analysis thus allows analysis from any of these perspectives. It lets a user do holistic studies to explore the ways in which different perspectives interact. The symbolic plant model is largely nonnumeric, describing components, their functions and interconnections, and their behavioral modes. The symbolic model could be complemented with a quantitative thermodynamic model.

Like a desktop publishing system, the IRTMM system has several properties:

- *One or a few users.* Authority and decision-making responsibility are highly centralized. The decision maker can obtain automated suggestions from multiple perspectives and can send recommendations to other human analysts.
- *Multiple integrated software applications.* Data can be taken readily from one to another, both automatically and under user control. Sometimes the applications will be built to be interoperable—for example, in the IRTMM system—while in other cases special effort will be used to make diverse systems interoperate.
- Presentation of a natural idiom (the P&ID) in a Wysiwyg interface. Users can view their products in a natural way; change them; review the effects of the change; and accept, modify, or retract those changes. Interactive computer-aided design and P&ID displays will be useful for many building- and facility-analysis applications.

The IRTMM system includes several features not normally found in desktop publishing systems:

- *System model.* In desktop publishing, the Wysiwyg interface changes the actual document. In desktop engineering, the Wysiwyg interface will normally change a model of the system, rather than the system itself, so that the user can safely use simulation to test a what-if outcome. In some cases, the user will also use the interface to control the modeled system.
- *Simulator.* In desktop publishing, a change propagates immediately to cause other changes without need for feedback. Engineering systems have feedback in their control. The dynamics of change, as well as the final state, are of interest. A time-dependent simulator shows those dynamics.
- *Open architecture.* In desktop publishing, a small set of vendors provide all the software applications. Users normally do not want to add proprietary applications to the tool suite. For desktop engineering, users often have useful legacy systems that they want to continue to use, and they will continue to want to develop and use specialized engineering applications.
- Integration management. Desktop engineering users need a manageable process to control passing of information among a changing set of applications.

4. J.L. Kolodner, R.L. Simpson, and K. Sycara-Cyranski, "A Process Model of Case-Based Reasoning in Problem-Solving," *Proc. Ninth IJCAI*, AAAI Press, 1985.

5. R.E. Fikes and N.J. Nilsson, "Strips: A New Approach to the Application of Theorem Proving to Problem Solving," *Artificial Intelligence*, Vol. 2, No. 2, 1971, pp. 189–208.

6. D.E. Wilkins, *Practical Planning: Extending the Classical AI Planning Paradigm*, Morgan Kaufmann, San Francisco,1988.

7. A. Darwiche, R.E. Levitt, and B. Hayes-Roth, "Oarplan: Generating Project Plans by Reasoning About Objects, Actions, and Resources," *Artificial Intelligence for Engineering Design, Analysis, and Manufacturing,* Vol. 2, No. 3, Sept. 1989, pp. 169–181.

8. T. Khedro, M.R. Genesereth, and P.M. Teicholz, "Agent-Based Technology for Facility Design Software Integration," *Fifth Int'l Conf. Civil and Building Engineering*, 1993, pp. 385–392.

9. D. Schenck and P. Wilson, *Information Modeling: The Express Way*, Oxford Univ. Press, 1994.

Acknowledgments

We appreciate the intellectual and funding support provided by Pacific Gas & Electric, Southern California Edison, the Electric Power Research Institute, and Shimizu Corporation of Japan. The original work received support from the industrial members of the Center for Integrated Facility Engineering at Stanford University. IntelliCorp provided the ProKappa system software used for the system implementation. We appreciate the comments and support of Alex Brousilovsky, John Chachere, David Garten, Fred Grote, Barbara Hayes-Roth, Mike Miller, Shuhei Murakoshi, Peter Rubino, Dan Rueckert, Felmir Singson, Henny Sipma, and Andrew Velline.

John C. Kunz is a senior research engineer at the Center for Integrated Facility Engineering (CIFE) at Stanford University. His interests include the theory and practice of symbolic modeling and its applications in a broad range of engineering domains. He was chief knowledge-systems engineer at IntelliCorp. He earned his BA, BE, MS, and PhD from Dartmouth College, Thayer School of Engineering, UCLA, and Stanford University, respectively. He is a member of the AAAI and the AAAS. Reach him at kunz@ce.stanford.edu.

Yan Jin is a senior research associate in the Department of Civil Engineering, Stanford University. He has done research on knowledge-based planning systems; distributed problem solving and multiagent systems; organization modeling; and its application to marine traffic control, computer-integrated manufacturing, collaborative design, and project management and control. He earned his PhD in naval and information engineering from the University of Tokyo. He is a member of the AAAI and Informs. Contact him at jin@ce.stanford.edu.

Raymond E. Levitt is a professor of civil engineering and associate director of Stanford's CIFE. His interests are in applications of knowledge-based systems to engineering and management of complex projects. Since 1990, he has led the development of the Virtual Design Team, a computational system that uses AI techniques to simulate decision making and communication in project organizations. He received a BSCE from the University of Witwatersrand in South Africa, and MS and PhD degrees in civil engineering from Stanford University. He is a member of the AAAI, the ASCE, and the Project Management Institute. Contact him at rel@ce.stanford.edu.

Shouh-Dauh Lin is a graduate research assistant in the Department of Engineering and Economic Systems at Stanford University. He earned his BS and MS from Stanford University. His interests include decision analysis and its application to maintenance of plant systems.

Paul M. Teicholz is the director of Stanford's CIFE and a professor (research) of civil engineering there. His interests include systems for collaboration and coordination of project teams; prediction of project performance (cost, schedule, quality), modeling the life-cycle information requirements of products and the use of these models for procurement, design, construction, maintenance and operation; and project-management systems. He received a BSCE from Cornell University and his MSCE and PhD from Stanford University. He is a member of the Operating Research Society of America (ORSA) and the Institute of Management Science (TMS), and is a fellow of the ASCE. Contact him at teicholz@ce.stanford.edu.

Contact all of the authors at the Center for Integrated Facility Engineering, Stanford Univ., Stanford, CA 94305-4020.