CIFE CENTER FOR INTEGRATED FACILITY ENGINEERING

The Intelligent Real-Time Maintenance Management (IRTMM) System: Support for Integrated Value-Based Maintenance Planning

By
John C. Kunz, Yan Jin, Raymond E. Levitt, Shouh-Dauh Lin, Paul Teicholz

CIFE Technical Report #100
January, 1995

STANFORD UNIVERSITY

If you would like to contact the authors please write to:

c/o CIFE, Civil Engineering,
Stanford University,
Terman Engineering Center
Mail Code: 4020
Stanford, CA 94305-4020

SUMMARY
CIFE TECHNICAL REPORT #100

Header:
Title: The Intelligent Real-Time Maintenance Management (IRTMM) System: Support for Integrated Value-Based Maintenance Planning

Authors: John C. Kunz, Yan Jin, Raymond E. Levitt, Shouh-Dauh Lin, Paul Teicholz

Publication Date: January 1995

Funding Sources:
- Name of Agency: EPRI, Shimizu, PG&E, Southern California Edison
- Title of Research Project: IRTMM

1. Abstract: The overall objective of the Intelligent Real-Time Maintenance Management (IRTMM) project is to help process plant owners perform value-based plant maintenance, i.e., to perform maintenance as needed for engineering and business reasons, rather than periodically (i.e., need-it-or-not) or following a breakdown. The system includes three coupled functions, each implemented as an independent software module: Situation Assessment (SA), planning and Value Analysis (VA). Users interact with the system using a graphic user interface that displays interactive Process and Instrumentation Diagrams (P&ID's) and various tables, charts and graphs. The modules share a symbolic plant model that describes plant components, their attributes and their connectivity. Since the plant processes include pumping, steam generation and fluid flow, the process model is general, and the planning and business models are also very general. The system has been tested successfully with test cases provided by power company utilities and from a large process plant.

2. Subject:

- What is the report about in laymen's terms? Use of multiple integrated computer-based techniques to support value-based plant maintenance, i.e., techniques to help an owner to perform maintenance as needed for engineering and business reasons, rather than periodically (i.e., need-it-or-not) or following a breakdown.
- What are the key ideas or concepts investigated? Specific techniques for situation analysis, planning, value analysis; use of an integrated product and process model.
- What is the essential message? The approach can work.

3. Objectives/Benefits:

- Why did CIFE fund this research? It is broadly important for and relevant to process operators;
- What benefits does the research have to CIFE members? Shows the way for specific approaches and needed developments
- What is the motivation for pursuing the research? Apply and test modern computer methods with real industrial test cases.
- What did the research attempt to prove/disprove or explore? Could value-based maintenance be performed?

4. Methodology:

- How was the research conducted? CIFE developed software given problem specification from industry.
- Did the investigation involve case studies, computer models, or some other method? Both

5. Results:

- What are the major findings of the investigation? Software tools will support integrated value-based maintenance management.
- What outputs were generated (software, other reports, video, other): Video; this paper

6. Research Status:

- What is the status of the research? Supported initially by IRTMM, then by Intel, now by Kaman Sciences for USAF and EPRI.
- What is the logical next step? Continue research (with Kaman), try pilot test
- Are the results ready to be applied or do they need further development? Ready for pilot test and industry software product development, not for operational use.
- What additional efforts are required before this research could be applied? Pilot test, product development necessary. Research (especially on Situation Assessment) would be helpful.

December 22, 1994

The Intelligent Real-Time Maintenance Management (IRTMM) System: Support for Integrated Value-Based Maintenance Planning

John C. Kunz, Yan Jin, Raymond E. Levitt, Shouh-Dauh Lin, Paul A. Teicholz

Center for Integrated Facility Engineering

Stanford University

Stanford CA 94305

Abstract

The overall objective of the Intelligent Real-Time Maintenance Management (IRTMM) project is to help process plant owners perform value-based plant maintenance, i.e., to perform maintenance as needed for engineering and business reasons, rather than periodically (i.e., need-it-or-not) or following a breakdown. The system includes three coupled functions, each implemented as an independent software module: Situation Assessment (SA), Planning (PL) and Value Analysis (VA). The SA uses model-based diagnosis, heuristic classification and case-based reasoning to monitor collected plant data and identify the presence, causes and potential effects of problems. The planner constructs plans of activities to perform maintenance tasks. The VA identifies alternative times at which maintenance can be performed and shows the expected costs and dollar savings associated with each alternative. Users interact with the system using a graphic user interface that displays interactive Process and Instrumentation Diagrams (P&ID's) and various tables, charts and graphs. The P&ID's allow a user to inspect any selected subsystem, identify component operating parameters, and review and make notes regarding component performance, operational and maintenance history. The modules share a symbolic plant model that describes plant components, their attributes and their connectivity. IRTMM has been designed and implemented to support generic power and process plants. Since the plant processes include pumping, steam generation and fluid flow, the process model is general, and the planning and business models are also very general. The system is implemented using object-oriented software with associated objects, displays and systems interface utilities, rules and methods. The system has been tested successfully with test cases provided by power company utilities and from a large process plant.

Introduction

A simple statement of the maintenance objective for a plant is that plant systems always are available to support plant function, without ever limiting plant production. More precisely, the maintenance objective for the plant is that the cost of any maintenance activity should be less than the expected marginal value of production enabled by the planned activity. A managerial challenge is to allow the plant operating engineers and the owner to share information about current plant component status and the business situation and for them to plan maintenance to meet this time-varying objective.

Supporting this objective is difficult. It is difficult to assess the amount of risk posed by an observed non-critical problem to future production. There are multiple goals (e.g., high long-term availability, minimal short-term cost); goals change (e.g., between availability and cost concerns); goals conflict; indicator data are almost never completely reliable or adequate. The problem has multiple aspects, including interpretation of observed data, diagnosis of problems, repair and maintenance planning, and business evaluation of the value-added of different repair and maintenance options. Finally, significant judgment is needed to interpret both available engineering and business data, and clear business policy is needed to define the "value" of maintenance.

Maintenance can be performed following breakdown, periodically based on utilization metrics (e.g., calendar time or hours of operation), or predictively following condition monitoring (e.g., detection of warning conditions such as high vibration). In turn, repair can be partial or total.

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

- Situation Assessment (SA): interprets observed data as being normal or abnormal and diagnoses causes and effects of plant equipment problems. It analyzes system performance to identify indications for condition-based maintenance. Given specialized input data from instrumentation (e.g., pressures, flow rates) and expert diagnostic systems (e.g., vibration analysis), the SA focuses on systems diagnosis. It identifies root causes and effects of component problems, where some of those causes and effects are in the component with a problem, and others are in subcomponents or connected systems. The user selects problems to consider for planning.
- Planning: Given a set of problems to repair, provided by the SA or by a user who is considering maintenance for any reason, the planner builds plans of the activities needed to repair a diagnosed problem. The planner can also merge related plans for the same or different components that can be performed during the same plant outage. Specifically, it identifies required activities to perform particular repairs, suggests alternative start times, identifies plans that can be merged, and identifies dependencies among plans that imply ordering of the sequence of plans.
- Value Analysis (VA): identifies the dollar costs and predicted benefits of performing every selected repair plan at different times. The user can consider the repair for any reason. Using predicted power demand and plant operating and maintenance cost data, the VA assesses the net unit operating costs associated with performing plans at different possible future times. The VA uses a decision-analysis procedure: it considers possible choices of repair actions and the chance outcomes that can arise given any choice. The value of a choice depends both on the choices and chances and on the probabilities and costs of each occurrence. Since it is usually impossible to get good probabilities, the system identifies "break-even" probabilities of failure such that an owner is indifferent between two options. The user then judges whether actual probability of failure exceeds the break-even.

The IRTMM system provides interactive analyses to facilitate engineering decision-making, not automate it. Given some data from a data acquisition system, the system identifies candidate causes of problems 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, selecting both the desired repair activity and the planned repair time, after considering the system-generated options.

The system is designed to reside on a computer network. It can receive component status information from an on-line data acquisition system and 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. Recommended work could be logged in the CMMS.

Shared Plant Model

A fundamental part 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 (e.g., 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 (e.g., operating, startup, failed), engineering parameters such as vibration and temperature (e.g., high, normal, low) and relationships such as connected-to. As shown in Figure 1 , each module has a copy of the generic plant model. Control of IRTMM processing involves shifting system focus of attention from the SA to the Planner (PL) to the VA modules. Each module requests data from its information source.

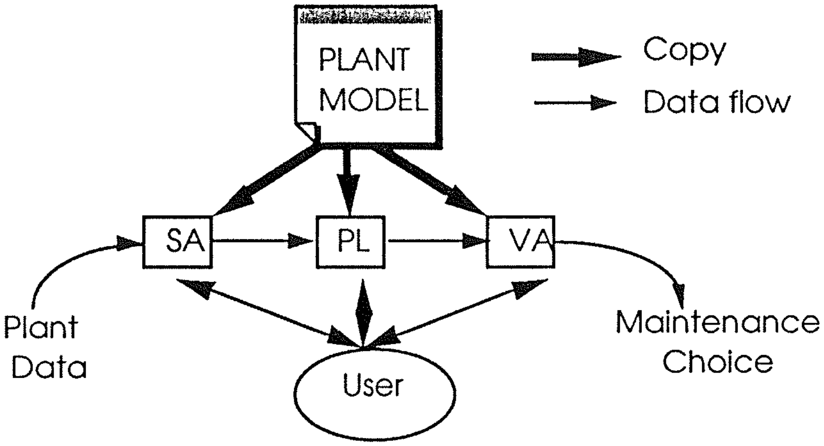

Figure 1: IRTMM Architecture

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. Requested data, specifically the values of attributes of instance objects, are received and stored by the querying module.

Table 1 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 model constituents are used by the different IRTMM modules, as shown by x's in the rightmost three columns. Hatching shows data items shared among two or more modules.

Form		**SA**	**Planner**	**VA**
	Components (e.g., 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 (e.g., Pumping): Method to:			
	- Propagate behavior	x		
	Tasks (e.g., 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 (e.g., 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 (e.g., Outlet Temperature)			
- Expected, Simulated values	x		
Plants (e.g., Unit-1)			
- Demand, price, cost profiles			x
Resources: Actions they can perform			x

Table 1: Entities defined in the shared plant model

The plant model represents the following classes of plant components: Filters; Headers; Heat exchangers; Instruments; Pipes; Process equipment; Pumps; and Valves.

Each parameter is represented explicitly in the model. Each has a value, which is often numeric. Numeric parameters also have a state. The value of the state attribute of parameters is, normally, one-of: High, Normal, Low, Off.

The SA and Planner use the model as a source of the components of the plant and their connectivity, engineering functions, and possible behaviors.

Situation Assessment

The Situation Assessment (SA) system diagnoses plant equipment problems. Given specialized input data from instrumentation (e.g., pressures, flow rates) and possibly from specialized expert diagnostic systems (e.g., vibration analysis), the SA provides a systems diagnosis. It identifies potential root causes and effects of component problems, where some causes and effects may be in the component with a problem, and others in subcomponents or connected systems. The SA uses a combination of methods to do its 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&ID's) that allow them to view P&ID's for any selected subsystem, identify component operating parameters, and review and make notes regarding component performance, operational and maintenance history.

The SA provides a systematic monitoring and component diagnosis capability for facility equipment and systems.

The first version of the SA was implemented by H. Sipma in the BB1 software environment [Sipma 92].

SA purposes

The SA includes the following capabilities:

- Check reported data for consistency: report alarm conditions for data that is out of 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. The out of expected range test is now largely performed by plant monitoring systems, but it is part of the SA 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, identify candidate causes and report evidence for and against hypothesized faults. Report causes, effects, supporting data, missing data, recommended actions.

- Show a shared, annotated P&ID to plant staff. Any IRTMM user is able to bring up a subsystem P&ID and view the following information provided by any SA user:
  - Measured data currently in the plant model system, e.g., temperatures;
  - Comments by staff, e.g., allow one user to store the fact that a component was repaired and is now back on-line and allow another interested user to discover the fact;
  - Inferred data, e.g., the computer concludes that a component is unreliable and should not be used except in emergency.
- 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 is designed to have two modes of operation:

- Periodic: every period (e.g., hour), it would query the monitoring system for the current status of all measured parameters, assess the diagnostic status of each component, and display a summary assessment of the status of all monitored subsystems, including trends when they are available. Plant maintenance supervisors will be principal periodic users. At least once a shift, they would review the situation assessment and prepare work orders as appropriate. Note that the SA is designed to operate as a maintenance advisor, not a control system.
- Demand: when initiated by a user or a monitored event reported by the data acquisition system, it would query the monitoring system for the current status of components in a selected subsystem, assess the diagnostic status of the subsystem, and display analysis results on an annotated subsystem P&ID. When requested by a user, it will query or update the Work Management System history file. Individual plant maintenance staff and management will be principal demand users. They will review outstanding work orders; review component status, history, trends and comments; plan maintenance activities; and log comments about their actions, observations and conclusions.

SA Reasoning

Knowledge systems now routinely do diagnostic reasoning using three methods: model-based diagnosis, heuristic classification and case-based reasoning. The SA uses a combination of each of these methods.

Model-based diagnosis (MBD) [deKleer] involves qualitative simulation of system behavior. First, a user (or an algorithm) sets up the plant model, selecting components to represent in the system, their connectivity, their states (e.g., on, off, open, closed), and their assumed behaviors (e.g., as designed, leaking). Next, a change is injected into a model of a system (e.g., a valve closes or a leak starts), and a simulation is run to propagate the change through the system to determine the complete system behavior given the assumed system parameter states. Finally, simulated and observed data are compared. If data are consistent, then the assumption is made that the state of the physical system and the state of the model are consistent. If the observed and simulated data are inconsistent, assumptions about the model are varied until simulated and observed data become consistent. Any abnormalities in the model state are sufficient to explain any abnormalities in the observed system. MBD was used in the SA to analyze the behavior of a process cooling water (PCW) system in a process plant, and the method generally applies to situations in which relatively simple rules describe the propagation of behavior from one component in a system to another.

Heuristic classification [Clancey] is the basis for classic expert systems. Evidence, i.e., measured data, is abstracted and related to a predefined potential problem. The problem is matched with a solution, and the solution is refined. Heuristic classification was used in the SA to analyze situations that have idiosyncratic causal behavior that were not readily amenable to the simulation approach of model-based diagnosis, e.g., the causes and effects of bearing failure and shaft imbalance.

Case Based Reasoning (CBR) [Kolodner] is the basis for some diagnostic systems and many recent help-desk applications. An expert creates a set of cases, each of which includes some descriptions of a situation (i.e., a case) and an associated statement of a causing problem and suggested repair. Thus, the CBR method aggregates the steps of the heuristic classification. Diagnosis involves simple matching of observed data with the data of each case. Relevant cases are reported as appearing similar to the observed situation. Note that MBD uses no notion of "fault." Systems simply have behavior. Observed and simulated behaviors do or do not compare. Separate from the MBD technique, a user may choose to modify a system (physical or model) to change its behavior. In the SA, the MBD technique was used to generate 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 expected outcome.

The SA model performs the following reasoning procedures:

- Set up a test case: allow a user to introduce a problem into the model and record both a description of the problem and the appropriate repair. This set up method then identifies consequences of the introduced problem (by invoking the Propagate! method) and collect these secondary parameter values into the case record (by invoking the Snapshot! method).
- Propagate qualitative behavior of a component to its downstream components (Propagate_Behavior!), i.e., check the state of component parameter inlets and determine the state of component discharge parameters. For a component that is functioning normally, the propagation method sets the state of discharge parameters for a component to be the same as the states of corresponding inlet parameters, e.g., (high, normal, low) temperature for the inlet gets propagated to (high, normal, low) temperature of the outlet. Components whose role is to heat (cool) set the discharge temperature to high (low) if fluid inlet is high (low).
- Snapshot values for a subsystem (Snapshot!), i.e., record the parameter state assignments made by a user and the associated states of all non-normal 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 snapshot is added to a library of cases.
- Situation assessment for a system (Diagnose!), i.e., compare a set of observed parameter data with the cases in the library and report those cases that have values that are the closest to the observed set of parameter status observations.

The SA has been developed and tested 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 reduces chilling capacity of a chilled water system, and blockage of a filter in a process flow system.

Figure 2 shows part of the graphic user interface to the Situation Assessment module. The interface is an interactive P&ID with the following functions:

- Record and report user notes, component history, measures of success;
- Show parameters of any selected component;
- Invoke a diagnostic routine to diagnose problems that can cause any observed parameter abnormality;

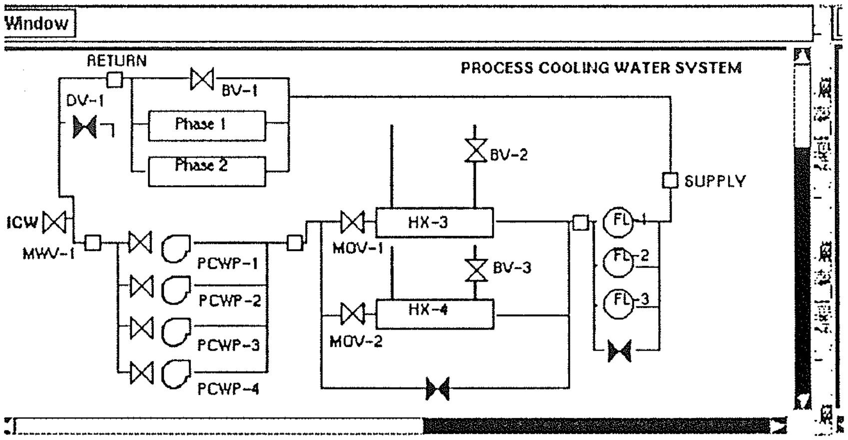

Figure 2 : Active P&ID for a Subsystem

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. Selecting any icon that shows in red because it is abnormal, the user can view the information discussed below in the SA Testing section.

SA Testing:

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 Chilled 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

The system identifies two possible causes for the observed low PCW discharge (return) pressure. In comparison with the selected explanation, the other has less significant evidence in favor and more evidence against it. Thus, between the possible causes shown above, the first is more likely and should be confirmed first.

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 Hx-3 inlet valve)
- MOV-2 Closed (hand-operated Hx-4 inlet valve)

For MOV-1 Closed,

Evidence +:	High Hx-4 discharge temperature
	High supply header discharge temperature
	High return header discharge temperature
	Hx-3 outlet flow off
	High Hx-4 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.

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

Planner

Engineering planning is a process of generating work procedures for workers and machines to follow to achieve an engineering goal and for planners to estimate costs and to manage projects. From a planning point of view, the engineering planning problem is likely to be complex because it is highly contextual and may involve many types of objects, actions and resources.

Early AI planners showed the possibility of using the computer to identify plan activities and to infer activity precedence [Fikes 1971]. While such early "STRIPS"-style planners evolved by the mid 1970s, they remain ill-suited to most engineering domains because of the limitation of the activity representation adopted in these planners [Wilkins 1988, Levitt 1989]. General purpose planners in the tradition of STRIPS still have not found broad use for generating realistic plans in engineering domains. On the other hand, narrowly scoped expert planning systems can be used for specific domains but have little applicability to even slightly different domains.

The research on engineering planning described in this section attempts to address the shortcomings of overly general or overly specific planning approaches by modeling the behavior of plan elements. OARPLAN, a model based planner, defines activities in a engineering plan by their constituents: *objects*, *actions*, and *resources* [Darwiche]. That is, an activity specifies an *(object, action, resources)* set, as well as their traditional attributes of times (e.g., start time, end time) and relationships (e.g., successors) It generates a plan by reasoning about objects, actions and resources of a specific engineering domain.

OARPLAN uses generic models of objects (e.g., Pumps, Valves) contained in the shared plant model. In addition, it uses models of actions (e.g., Close, Remove) and resources (e.g., Plumbers, Cranes).

Planner Introduction

Planning in OARPLAN involves two steps: activity generation and activity ordering. OARPLAN planning is hierarchical in the sense that it recursively generates activities at lower planning levels by elaborating those at higher levels, based on object-action-resource models. Activity ordering is performed 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 is obtained after all elaboration is performed and all constraints are examined and satisfied.

Planner Purposes

The Planner includes the following capabilities:

- Clear a component;
- Generate activities based on elaboration and refinement of a top level activity.
- Order planned activities;
- 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 in order to isolate a component and make it safe for workers. If a valve is disabled, the procedure searches upstream from the component to clear to find 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, and objects can have relationships that specify how they decompose into parts. As shown in Figure 3, for example, the activity generation procedure recursively generates detailed subactivities to elaborate a given top-level activity. 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: Clear, Examine, etc. The generation procedure takes a given top level task, (Repair UNIT_3_MBFP_Pump Some-Resources) and elaborates an activity tree as shown in Figure 3. 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 in 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, i.e., have the same object, action and resources. Merging plans is desirable when a merged plan can be performed faster than several independent plans and when opportunistic maintenance can be performed relatively inexpensively at the same time that another required maintenance activity is being perfomed.

Planner Testing:

The SA reported that the main boiler feedwater pump (MBFP) required maintenance because vibration indicated a bent shaft problem. Figure 3 shows the generated repair activity tree. The plan itself is a linearized list of boxed activities.

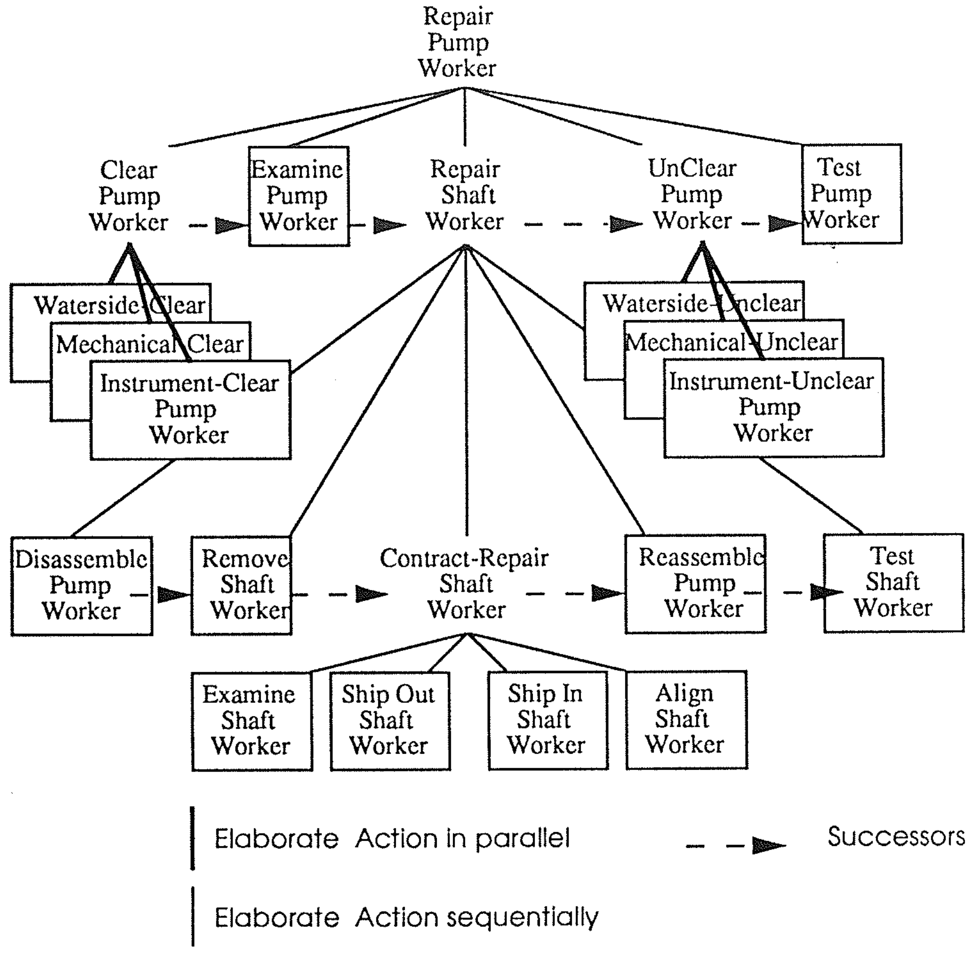

Figure 3: OARPLAN Plan Activity Tree

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 Analysis (VA)

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, at the next scheduled outage that is long enough, and defer maintenance until after the end of the study period.

Decision Analysis recognizes that when we make a choice, we take chances. Given a choice of time to maintain a component, for example, 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, i.e., decision analysis, to analyze the *Expected Value* of each choice, based on the likelihood and value of the outcomes of each possible chance.

Figure 4 shows the way that decision analysis computes the expected value of a choice 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 i-th hour, i = 1, ..., q; defer the repair until after the end of the study period. There are some number of chance outcomes for the each choice. For the alternative of repairing at the i-th hour, for example, the possible chance outcomes can be represented by the tree shown in Figure 4.

P_i = Probability of failure during the i-th hour;
B_j = Net benefit of repair and revenue loss if the component fails in the j-th hour;

Expected Benefit of Choice = __P_jB_j

Summation takes place over each of the chance occurrences, i.e., 1<=j<=i.

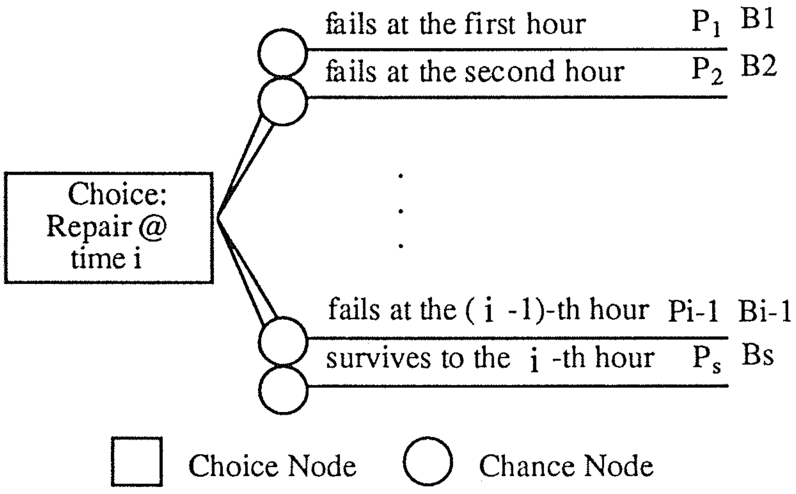

Figure 4: A choice-chance tree.

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.

The size of the decision space depends on the size of both the study period and the grain size of data: a 7-year study period must consider 7 outcome possibilities if the problem assumes an annual data grain size and it considers 84 outcomes if the problem assumes data precision of one month. The impact of each decision 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.

Current VA Assumptions

The VA considers only one failing component at a time, so only one maintenance activity is passed from the Planner at a time 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 thing that is random in the analysis is the failing component's life. Unplanned repair is started immediately following component failure, but the time to perform unplanned repair is assumed to include both a wait time to assemble parts and crew plus actual repair time. Components deteriorate only during operation. That is, when idle, component performance does not deteriorate. Once a component is repaired, it is restored 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 k-th hour means that the breakdown happens at the beginning of the k-th 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 life distribution of a failing component is assumed to be Weibull, the most widely used parametric family of failure distributions [Barlow and Proschan]. The Weibull distribution has the form:

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

where T: failing component's life.

λ: scale parameter.

α: shape parameter.

The VA gets the mean time to failure and standard deviation on the estimate from the model and it converts these parameters to the two parameters of the Weibull distribution.

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, e.g., whether the repair is planned or unplanned.
- cost of repair, C_{repair}, either planned or unplanned, during a time period.

The expected benefit of a chance occurrence is computed by summing the costs, weighted by probability of failure at each time period, to determine the expected net benefit of a choice:

$$\text{Benefit of a chance} = \overset{\text{end of study period}}{\underset{\text{time from now} = 1}{\bullet}} (S - C_{downtime} - C_{repair})$$

Suppose the component fails at the k-th hour. Total cost of failure would be:

Total Cost if the component fails at the k-th hour
= costs due to deterioration from the first to the (k-1)-th hour, e.g., cost of deration and marginal costs of operating an aged component, i.e., cost of production loss
+ costs due to shutdown (for repair) from the k-th 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 cost of repair at the k-th hour also considers the time value of money.

The user provides the following input to the VA module:

- A 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 the above 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 in comparison with 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;
- Break-even probability between two selected two timing options.

Test Case: Fan Bearing Maintenance

Investigating excessive fan motor vibration, an engineer recommends that a component needs its bearing oil changed. If oil is not changed, there is concern that the bearings will fail and bring down the plant. Arbitrarily, the engineer chose to analyze the economic value of maintenance options for a 1-week study period. The engineer identified the following choices:

1. Change oil off-line ASAP, bringing down the plant during the procedure;
2. Change oil on-line ASAP, keeping the plant operating during the procedure;
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;
3. Survival of the bearing until the next planned outage (after the end of the study period).

The engineers made a number of assumptions:

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

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

The table below 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 probability of an oil spill. For example, when the probability of oil spill is high and the probability of bearing failure is low, the best strategy is to defer the off-line oil change. When both the probability of oil spill and probability of bearing failure are high, the riskiest situation, we should immediately shut the system down and change the oil off-line. Otherwise, the best strategy is to change the oil on-line. The table below shows the optimal decision in terms of the bearing failure and oil spill probabilities. Also shown are relative expected benefit (K$) of each decision to that of the second best.

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

☐ defer ▓ on-line ░ ASAP

The patterns indicate three preferred repair alternatives. Numbers in boxes show relative expected benefit (K$) of the preferred alternative over second-best alternative.

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 Discussion

An important part of the power of the IRTMM system comes from the engineering content of the model, 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 suggested in Table 1, 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 implemented SA initially for a power plant and later for the water handling subsystem of a process plant, and we had to make only minor extensions to the functional definitions in the generic plant model. 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 in STEP. However, an important part of the total system power comes from the behavioral methods defined in the function models, and while such methods are standard capabilities of object-oriented technology, they are outside of the normal capabilities of STEP-style product models.

As shown in Figure 1, each application module has a copy of the generic plant model. Thus, each module shares the complete plant model ontology. In fact, as shown in Table 1, only very limited form and behavior data items are shared by two or more modules, 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 (i.e., the classes 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. A simplest-possible alternative is to share minimal data, i.e., share 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. One or the other module could report the same information under two names or make any other appropriate parametric transformation. An alternative integration architecture [Khedro] is for an external agent to do the transformation after one module registers that it can produce particular data and another module registers its interest in the data.

We are reluctant to generalize our limited results to some of the major efforts to share large ontologies, such as STEP [Schenck. A practical conclusion of our work is that it is probably very worthwhile to identify the engineering problem to be solved as clearly as possible, e.g., 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. It is our experience that the sharing mechanism can easily become far more complex than necessary to support the engineering objective.

SA Discussion

The SA uses a hybrid approach to diagnosis that includes model-based diagnosis, heuristic classification and case-based reasoning. The hybrid approach of the SA has a number of benefits:

- Uses knowledge of plant design, specifically of the form, function and behavior of the plant, because the Model-based Diagnosis (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 Case-Based Reasoning (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 MBR technique predicts behavior;
- Does not require staff to identify all failure modes, because the MBR technique can be invoked exhaustively to find the behaviors which emerge from various input conditions.

On-line data acquisition systems are entering broad use within the process industry. While engineers always want more sensed 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 components. Potentially, it would accept input from specialized expert system diagnostic routines, e.g., a vibration 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 trending of data or quantitative prediction of the severity or timing of predicted degradation.

Planner Discussion

OARPLAN represents the object, action and resources of planned activities. These entities and their attributes are considered in both planning and merging plans. 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* a gives generality to the planning procedure. Third, the action elaboration and refinement allows the planning to become as specialized as required 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 non-hierarchical 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, and the second pass allows it to accommodate specialized engineering details.

The effectiveness of the IRTMM planner will 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.

VA Discussion

The fan bearing problem fits the VA framework naturally. As evidenced by this and a number of 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 return the user with maintenance choices regarding timing, probability of survival, and the break-even analysis.

However, the calculation is based 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, the VA will return an over-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 the recurring failure and the 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

We expect that successful implementation of integrated maintenance systems such as the IRTMM system can favorably affect the following risk factors for plant downtime:

- "Procedure errors:" reduce their incidence because operators will have the opportunity to predict the effects of planned operations, since they represent direct risk of downtime;
- False alarms (i.e., proper response is to ignore): reduce their incidence, since 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 call-outs to technicians and engineers at home: reduce their incidence, since 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, since they indicate that staff lacks the access to knowledge or data to perform a job properly;
- Alarms that do not indicate production impact (i.e., proper response is to fix a component, but production is not affected): increase their incidence, since they indicate effective predictive maintenance;
- Rework: reduce its incidence, since it detracts from effective maintenance, and cost;
- Timeliness: reduce the time from decision to perform maintenance or repair to time that repair is successfully completed, since delayed definitive repair simply adds breakdown risk;
- Personnel and equipment utilization: by reducing equipment downtime and helping technicians to do the work right the first time with fewer call-outs for help, the system should help improve personnel and equipment utilization.

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, evaluation procedures;
- Reduced stress for engineers, technicians, especially on short-staffed 12-hour shifts;
- Qualitatively decrease time to design facility retrofit projects;
- Facilitate startup and help operators to bring new plant facilities to peak production capacity faster.

Plant engineers have suggested some possible extensions, including:

- Predict imminent faults based on statistical process control trend analysis and potential failure modes;
- 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;
- P&ID objects should be able to show component schematics and disassembly sketches;

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 analyses thus allow analysis from any of these perspectives, and they allow a user to do wholistic studies to explore the ways in which different perspectives interact. The symbolic plant model is largely non-numeric, 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 a number of properties:

- **One or a small number of users:** authority and decision-making responsibility are highly centralized. The decision-maker can obtain automated suggestions from multiple perspectives and can send a set of 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, e.g., 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 effects of change, and accept, modify or retract those changes. Interactive CAD and P&ID displays will be useful for many building and facility analysis applications.

The IRTMM system includedes some 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 are of interest as well as the final state. A time-dependent simulator shows those dynamics.
- **Open architecture:** in desktop publishing, a small set of vendors provides 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.

References

[Clancey 85] Clancey, W.J., "Heuristic Classification", *Artificial Intelligence* 27, pp. 289-350, 1985.

[de Kleer] de Kleer, J., Williams, B.C., "Diagnosis with Behavioral Modes", *Proceedings of IJCAI,* pp, 1324-1330, 1989.

[Fikes 1971] Fikes, R.E., Nilsson, N.J., "STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving," Artificial Intelligence, 2, 1971.

[Darwiche] Darwiche, A., Levitt, R.E., Hayes-Rotyh, B., "OARPLAN: Generating Project Plans by Reasoning about Objects, Actions and Resources," Artificial Intelligence for Engineering Design, Analysis and Manufacutring (AI/EDAM) 2(3), 1989.

[Hayes-Roth] Hayes-Roth, B., "A Blackboard architecture for control", *Artificial Intelligence*, 26: 251-321, 1985.

[Kolodner] Kolodner, J.L., Simpson, R.L., and Sycara-Cyranski, K., "A process model of case-based reasoning in problem-solving", in *Proceedings of the Ninth International Joint Conference on Artificial Intelligence*, Los Angeles, CA, IJCAI, Inc., 1985.

[Khedro] Khedro, T., Genesereth, M.R., Teicholz, P.A., "Agent-Based Technology for Facility Design Software Integration," the *Fifth International Conference on Civil and Building Engineering*, pp. 385-392, Anaheim CA, 1993.

[Schenck] Schenck, D., Wilson, P., *Information Modeling: the EXPRESS Way*, Oxford University Press, 1994.

[Sipma 92] Sipma, H., Hayes-Roth, B., "IRTMM: The Situation Assessment Module," unpublished IRTMM technical report, 1992.

[Wilkins] Wilkins, D.E., *Practical Planning: Extending the Classical AI Planning Paradigm*, Morgan Kaufmann, Los Altos CA, 1988.

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 was supported by 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.