Greg Stanley and Associates
Operations Management Automation

Pipeline Diagnosis Emphasizing Leak Detection: An Approach and Demonstration

White Paper

Prepared By: Greg Stanley

April 22, 2012

Table of Contents

1 Introduction

1.1 Purpose of this document

This white paper is an overview of an approach to pipeline diagnosis, with an emphasis on leak detection. It illustrates parts of the overall approach with a demonstration, and explains how the demo fits into the overall strategy.

1.2 Background

Pipeline companies need to detect leaks quickly, but avoid unnecessary shutdowns due to misdiagnosis. Elaborate dynamic simulators model the pipelines. The simulator packages also detect faults by looking at deviations between measured and simulated values. When a measured flow deviates significantly below the simulated value over a significant time period, this indicates a leak, so an alarm is generated. But this fault detection is sensitive to other errors. Those errors are either due to model errors, sensor errors, or human input errors. The errors might be in terms of incorrect values, or in the timing of changes in those values.

A separate diagnostic system is needed to determine the root cause errors, to rule out false alarms, and avoid unnecessary shutdowns. In choosing the root cause faults to be diagnosed, the emphasis is on detecting leaks, but this requires choosing additional root causes defined as needed to avoid false alarms about leaks

2 Description of the liquid pipeline

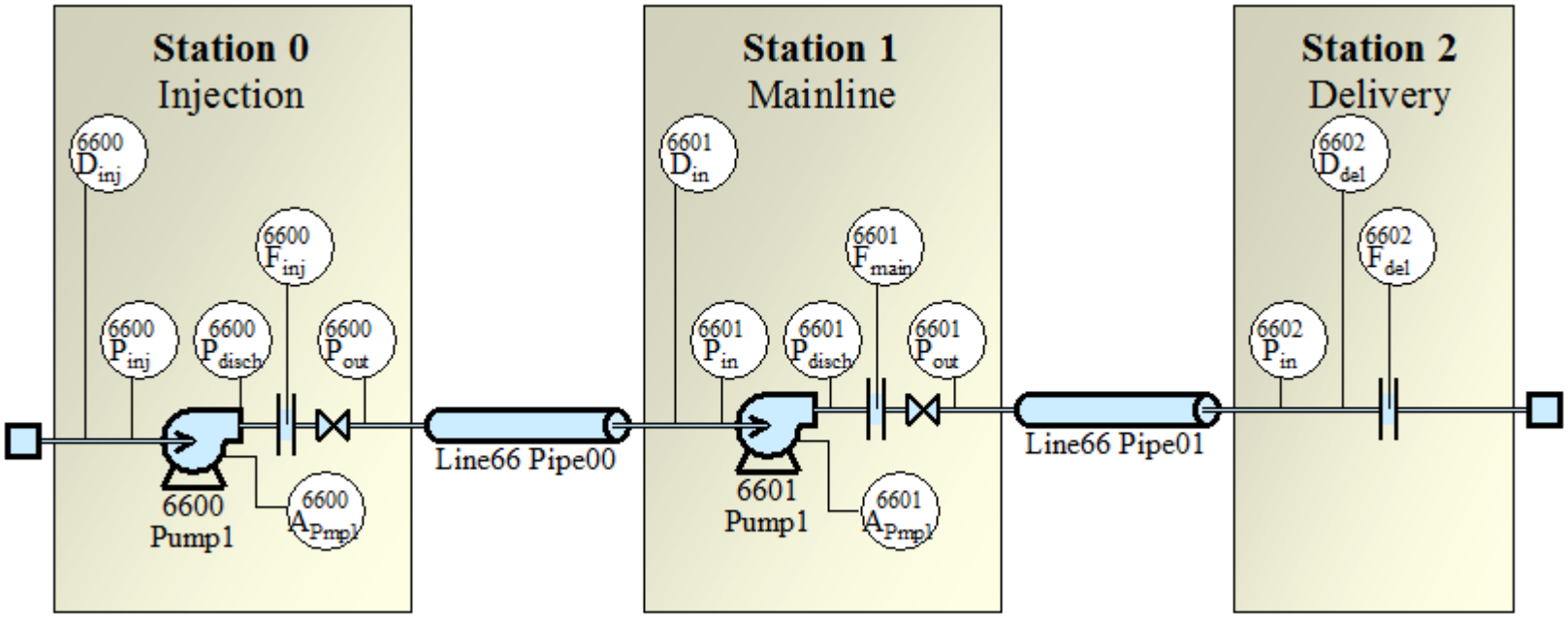
2.1 The process

The example process is a simplified version of a liquid pipeline with three stations. It includes measurements of flow, pressure, and density, sent from a SCADA system. A process flow diagram is shown on the next page. The symbol for orifice meters is used for flow meters, but their type is not important for the analysis.

An instrument tag naming convention is used. The first two digits represent the pipeline number. The next two digits represent the station number in that pipeline. F, P, D and A indicate flow, pressure, density, or amperes. Instead of using unique identifying numbers after that, we identify the sensors in subscripts based on their roles at a station: “inj” for injection (flow in); “main” for main line, “out” for outlet of a station, “del” for delivery (flow out); and “disch” for discharge of a pump.

The demo does not include many aspects ultimately needed in a full diagnostic system. A real system has to deal many additional issues such as valve positions and their changes for station bypass, startup and shutdown; suppressing alarms during instrument calibration or while using pigs, and so on.

2.2 Process flow diagram for demo example Line 66



3 The simulator

A detailed dynamic simulator would be running in a real system. A diagnostic system uses sensor values or event input, manual input, and values calculated by the simulator (called "sim value" for short in this demo).

In the demo, the diagnostic system just uses events. But some of those events are assumed to be generated based on the simulated values or their deviations from measured values.

For convenience in this discussion, we focus on just the parts of the simulation associated with the pipeline segments (called "pipes" in the demo). We assume that the measured inlet pressure and flow to each pipe are used directly (or at least strongly impact) the inputs used in the simulation. Then, an increase in the flow measurement at the input to a pipe eventually results in an increase in the simulated flow at the output of the pipe. Similarly, an increase in the pressure measurement at the input to a pipe eventually results in an increase of both the simulated flow and pressure at the output of the pipe.

The exact diagnostic logic may need to change if this is not the case, but this example illustrates the use of the numerical simulator as part of the diagnostic logic.

If there is a leak in a pipe, then material will be diverted to the leak, so that the actual and measured pipe outlet flow and pressure will decrease. But the simulation model is based on no leak being present. So, the measured flow will be lower than the simulated flow. A simulator package does more than simulation: it also checks the simulated versus measured outlet flow. When the deviation of measured outlet flow versus the simulated outlet flow stays too low for too long, it generates a material imbalance alarm event. In the demo, we just use a "low versus sim" event for flow meters.

4 Data reconciliation

4.1 What is data reconciliation

Data reconciliation provides estimates of process variables based on combining measurement information with process knowledge (such as mass or volume balances, or energy balances). The process knowledge is in the form of algebraic equations and inequality constraints. If the constraints are correct, and measurements fit assumptions about their noise, the resulting estimates will be better than those obtained just from raw measurements. When systems have significant dynamics, the variable values are integrated over a long enough period so that the steady state algebraic equations are a good approximation.

Data reconciliation has mainly been used for reconciling flow measurements, subject to material or volume balances. In this case, dynamics are often handled for inventory changes. For inventory measured by levels (or pressures), inventory change derivatives are treated as equivalent to additional flows. So the inputs are flow and level measurements, and the outputs are better estimates of the

level changes and flow, that are 100% consistent with the material or volume balances specified as constraints.

Data reconciliation is generally formulated as a least-squares optimization problem to minimize a weighted sum of the measurement adjustments subject to equality constraints. The weights are based on assumptions about instrument accuracy, expressed as a variance (square of the standard deviation). When reconciling just material or volume balances, the constraints are linear. The resulting solution (without inequality constraints) can be obtained by solving a set of linear equations.

The filtering strategy and averaging period are important factors. Filtering, or long averaging periods, filters out the higher frequency noise. What is left of instrument error after heavy filtering is, by definition, instrument bias. So, what data reconciliation is really doing is reconciling the instrument biases against each other.

For a more complete overview of data reconciliation, including some of the original published technical papers, see http://gregstanleyandassociates.com/whitepapers/DataRec/datarec.htm .

4.2 A strategy for using data reconciliation in pipeline monitoring

Data reconciliation will be used on the raw flow measurements over a period long enough to filter out most of the process dynamics variations, so that the steady state volume balances are a reasonable approximation. The adjustments made to the flow meter will be used to calculate an estimated bias for each meter. That bias estimate will be filtered over time, updated each time a new bias estimated.

The idea is modeling a measurement as a sum of the actual value, a bias term that can only change slowly, and random noise with zero mean. Instead of just using lightly filtered raw measurements in any analysis over the short term, you subtract the estimated bias for an “improved” estimate. The purpose of the data reconciliation is to explicitly estimate flow sensor biases. This use of data reconciliation emphasizes estimating values that change only slowly over the long term, which is the most appropriate use for steady state balances. And it does not second-guess the full-blown dynamic simulator over shorter time periods.

We also calculate an estimated loss at each pipeline segment, based on the difference of lightly filtered raw measurements, but corrected for the estimated biases, and corrected for dynamic imbalances calculated by the simulator.

We formulate the fault model to include the root cause faults “high bias” and “low bias”, because they affect other variables and events depending on its value.

There is a trade-off here between short-term accuracy and long-term accuracy. If a leak really persisted a long time, its effects would gradually get absorbed into the bias estimates. Using only steady state analysis, there aren’t enough flow measurements to estimate both biases and leaks – in fact there is really only enough to estimate the net effects of biases for sensors around each pipeline segment. That’s why we force biases to update only slowly over long time periods. But in trade, we get more accurate variable estimates when the leak first starts. The initial detection is far more important – then

the leak will never become a long-term leak. This part of the system would not be able to address small leaks that might go undetected for long periods, but any system will have trouble with that.

The bias strategy used is slightly different, but based on ideas used for an application at Exxon, as well as earlier work with steady state models and dynamics. See the technical paper "Online data reconciliation for process control", available at

http://gregstanleyandassociates.com/whitepapers/DataRec/AIChE-82-OnlineDataRecProcessControl.pdf

4.3 Data reconciliation is reflected in the demo as another event source

In the demo, we assume that the data reconciliation portion of the diagnostic system is running and generates events. The diagnostic logic in the demo just works on those events. The events are "bias estimate low" and "bias estimate high". "Bias estimate low" means that the bias estimate has deviated too far below a threshold. "Bias estimate high" means that the bias estimate has deviated too far above a threshold. The point of these root causes and alarms is not to indicate leaks. One purpose is to recognize alternate diagnoses instead of false leak alarms. Another purpose is to trigger maintenance work, because a lot of misdiagnosis could occur otherwise.

5 Diagnostic system

5.1 Domain object model

Domain objects are representations of what is being monitored. Domain objects include individual equipment, collections of equipment such as an entire pipeline, and locations such as stations. But domain objects could include users, and more abstract entities such as organizations.

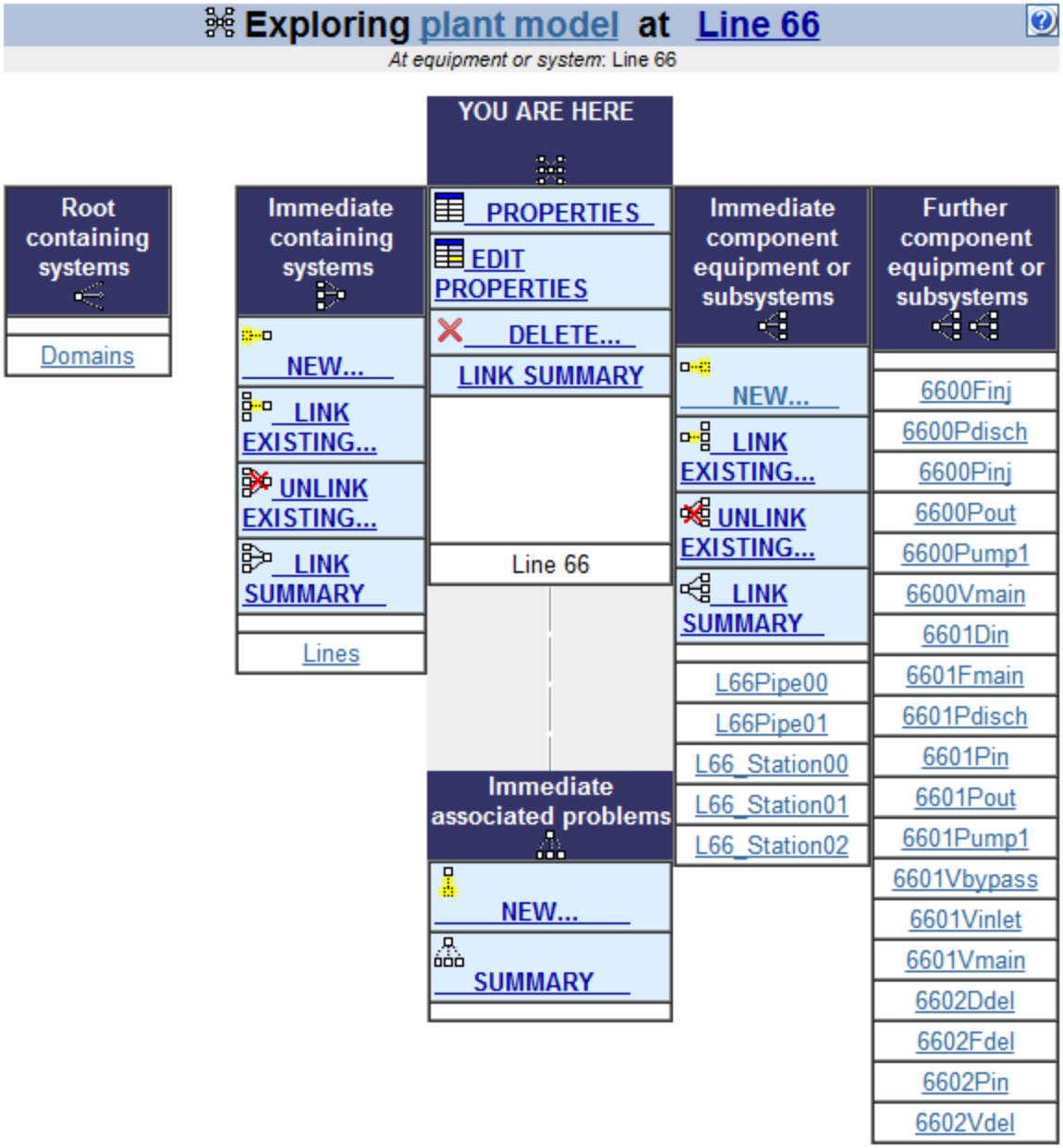
Domain objects have attributes such as active/inactive status, description, and so on.

Domain objects are organized in a containment hierarchy. For instance, the abstract object "Lines" contains Pipeline 66 and Pipeline 77. Pipeline 66 in turn contains 3 stations and 2 pipeline segments. Each station in turn contains objects such as pumps, sensors, and so on. This hierarchy is shown for Line66 in the screen shots in the appendices. One object can be contained by more than one object. For instance, both a location object and an overall-equipment object could contain a pump. This flexibility is available, but not yet used in the demo.

Events in the fault model, such as root causes, are associated with domain objects. For instance, certain kinds of faults, symptoms, and intermediate conclusions are associated with sensors, pipeline segments, and pumps. These faults for a flow sensor, pipe, and pump are shown in the screen shots in the appendices.

Organizing an application around domain objects and containment relationships is convenient for representation in the user interface, for inheriting properties and events when constructing the system, and for modularizing an application.

The domain objects in the demo represent pipes; pumps; valves; sensors for flow, pressure and density; stations, and pipelines. The following screen shot shows a view of the containment links up and down from Line 66:



The graph explorer is centered at “YOU ARE HERE”, looking multiple levels up and down the hierarchy. You navigate by clicking on any link, re-centering the display at that object. The user is in “expert mode”, so there are options for editing the containment hierarchy and also problems associated with the domain objects.

5.2 Event generation

A full diagnostic system receives some externally generated events, such as the material imbalance alarm events from the external simulation system. Based on raw data and its data history, it also generates its own events as needed to represent significant process disturbances.

This includes simple checks of measured values, simulated values, or their difference, as high or low versus limits. It also includes time series analysis – generating events based on the time history. For instance, standard deviation calculations for measurements result in events for zero or almost-zero standard deviation, or high standard deviation. Zero standard deviation, called "flat line", is useful to conclude a sensor is stuck, a common failure mode (including, but not limited to cases where a sensor is stuck at the top or bottom of its range). High standard deviation may indicate excessive noise in the sensor or the process. (Excessive sensor noise is a symptom of some instrument failure modes.) Excessive noise in the process is a useful symptom because it could indicate situations like pump cavitation (which in turn could indicate a major leak upstream of that pump).

Other events may be generated based on rates of change, occurrence of an event within a recent time period, and so on.

The demo assumes that events are available, and just focuses on processing the events. The events can be entered in arbitrary order. Additionally, the system asks for input on other events. In a manual system, this would be the main input method. When prototyping an online system, the questions are equivalent to acquiring data selectively.

5.3 Filtering

In a full diagnostic system, filtering would have to be carefully considered prior to event generation – using it to filter out higher frequency noise for typical variables, and avoiding it where time series analysis is used. Even aliasing should be looked at – avoiding the conversion of high frequency noise into lower frequency noise due to long data sampling intervals. We assume filtering is already addressed during the event generation. The demo just focuses on processing the events.

5.4 Fault model

The demo is based on an explicit graphical model of fault propagation from root causes through various symptoms and intermediate conclusions. (Conversely, it can be thought of as a representation of the possible causes for any event.) The fault propagation model is represented by a set of nodes ("problems") and a set of "is a possible cause of" links between them. The possible problems are associated with domain objects.

5.4.1 Problems

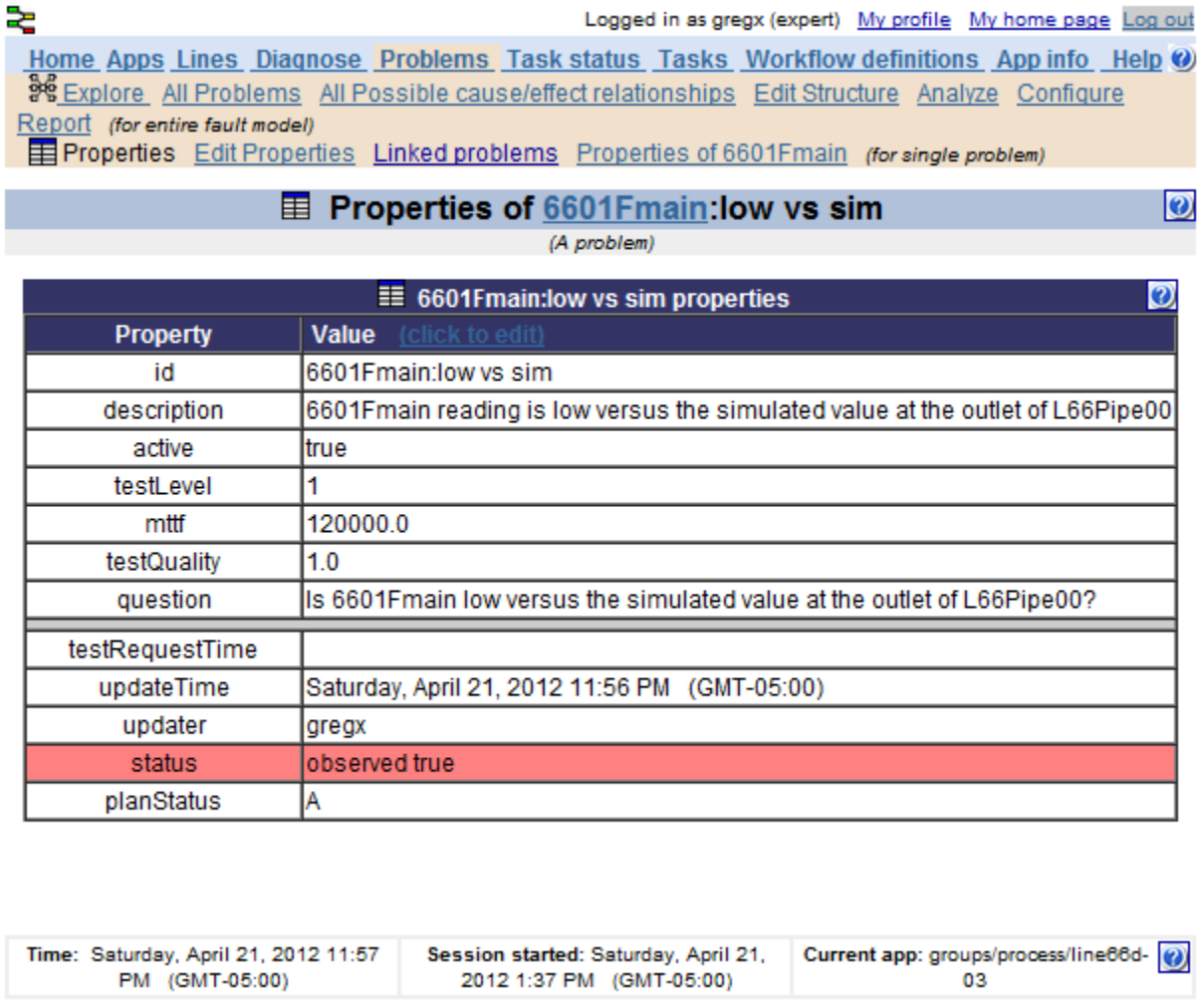
A problem is any condition that is abnormal, with undesirable consequences. A "true" status indicates the bad condition is present. False means "OK" - the problem is not present.

Problems include root causes, symptoms, and intermediate conclusions. In implementation, the distinction is really only an indication of the problem's location in the fault model. This is the most convenient for model development and a consistent user interface, because at any time during development, someone might ask "why might a root cause happen", generate some more fundamental root causes, and turn the previous root cause into an intermediate conclusion.

Problems are always associated with a domain object – a piece of equipment or subsystem. The naming convention for problems is to include the name of the equipment or subsystem, followed by a colon, followed by the name of the problem. For instance, 6601Fmain:flatlined is a problem associated with the flow meter 6601Fmain, that its value is unchanging.

In addition to the relationships to domain objects, problems also have a few properties (attributes) of their own. The following screen image shows these attributes for the mainline flow 6601FMain:

Properties of mainline flow 6601Fmain:low vs sim



Logged in as gregx (expert) My profile My home page Log out

Home Apps Lines Diagnose Problems Task status Tasks Workflow definitions App info Help

Explore All Problems All Possible cause/effect relationships Edit Structure Analyze Configure Report *(for entire fault model)*

Properties Edit Properties Linked problems Properties of 6601Fmain *(for single problem)*

Properties of 6601Fmain:low vs sim

(A problem)

6601Fmain:low vs sim properties

Property	Value (click to edit)
id	6601Fmain:low vs sim
description	6601Fmain reading is low versus the simulated value at the outlet of L66Pipe00
active	true
testLevel	1
mttf	120000.0
testQuality	1.0
question	Is 6601Fmain low versus the simulated value at the outlet of L66Pipe00?
testRequestTime	
updateTime	Saturday, April 21, 2012 11:56 PM (GMT-05:00)
updater	gregx
status	observed true
planStatus	A

Time: Saturday, April 21, 2012 11:57 PM (GMT-05:00) **Session started**: Saturday, April 21, 2012 1:37 PM (GMT-05:00) **Current app**: groups/process/line66d-03

The configuration properties (attributes) are above the grey line in the table above. Test quality is not yet used. The mttf is "mean time to failure", which affects the failure probability calculations. For this demo, it is always left at the default value. As long as all problem mttf values are at the same, they will have no impact. The properties below that grey line apply to the current diagnostic status. This dynamic diagnostic status is included for convenience, although it is actually stored in a "diagnostic instance". The Plan status of "A" is for "assertion" (a direct observation, based on manual input or data). Other possibilities include unknown, and true/false variations for predicted (future) values or inferred (past) values based on possible delays due to the direction of the cause/effect model links.

5.4.2 Root causes

A root cause is literally a problem with no "upstream" problems that could have caused it. The choice can be somewhat arbitrary, because for almost any existing root cause, you could ask another "why": why did that happen. People generally stop looking for more fundamental causes (stop asking "why") when they reach a point where corrective action can be taken, and where looking for more fundamental causes would cross an organizational boundary where they can't take corrective action.

We define the possible root causes for each type of equipment or other domain object. We try to define root causes that apply to equipment objects regardless of how they are connected to other objects. However, this simple approach often has to be modified for several reasons

(1) Significant equipment operation changes that affect downstream equipment can occur due to changes in input variables, without that equipment actually failing. But the effects have to be propagated further downstream. An example of this is pump cavitation caused by low inlet pressure. The cavitation leads to large, high frequency variations in flow and pressure, and low flow out. Those need to be propagated downstream, and to symptoms such as "excessive high frequency noise" for nearby flow, pressure, and amps sensors. From the standpoint of the pump by itself, low inlet pressure is a root cause problem. But in the overall system, there will always be reasons why the inlet pressure went low. "Low inlet pressure" is really more of an interface variable for assembling the system by modules. In the demo, we more specifically call this "inlet-pressure-low-NPSH" to highlight that the threshold for low pressure is near the NPSH (Net Positive Suction Head) for the pump.

(2) Sometimes we know the model will only explicitly cover some equipment, but we still have to account for failures in other equipment. Pumps illustrate this example as well. In this demo, we do not yet explicitly model valves. Yet a mainline valve with a "stuck shut" failure upstream of a pump will lead to low inlet pressure. Similarly, we do not model every short piece of piping in the immediate vicinity of the pump inlet. A major leak in any of those pipes will also lead to low inlet pressure. So instead, we provide catchall "root cause" failures like "upstream blockage" and "leak near inlet" for the pump. This provides a home to account for the myriad other possible failures. The effects for all these failures are the same. The test for all these failures is also basically the same – visit the site and perform a manual inspection near the pump inlet. So nothing is really lost by this aggregation of root causes. A diagnosability analysis

(discussed later) points to a need to combine root causes that can't be distinguished from each other, when they have the same test and effects. In the case of the demo, we moved the "leak near inlet" to a fault for a station called "leak between inlet and pump".

Root causes chosen for diagnosis should include faults that are common, and faults that have high impact. But, the demo is not a complete system. For the demo, we have included the following as representative root causes:

5.4.2.1 For sensors:

- Biased high
- Biased low
- Stuck high
- Stuck low
- Noisy sensor fault.
- Bad

"Bad" will cover other failures, missing data, and "bad PV" status from other systems or human entry. It is defined, but not yet actually used.

Bias errors are especially insidious, because the instruments appear to work normally, with values moving up and down over time. A bias may go undetected for a long time. Furthermore, a simulator needs information from sensors to drive it – at least some minimum set of independent inputs. So bias errors propagate through the simulation, and end up affecting other deviation calculations between simulated and actual. Estimates of leaks are extremely sensitive to errors in flow meters, so bias errors are especially significant for avoiding false alarms for leak detection.

"Stuck high" means that the sensor reading is stuck at a particular value, higher than the actual value. It might be associated with hitting the top of the instrument range, or it might be stuck in range, but at a high value. The direct tests and corrective action for stuck high and stuck low faults are the same – a field calibration check. So, one goal of the fault modeling is to ensure that enough effects of this failure are represented, so that automated tests can recognize it. (The "stuck" part is easy – testing for a flatlined reading by looking at standard deviation over time.)

"Stuck high" and "stuck low" are separate root causes because their impacts are quite different. For instance, in the case of a mainline flow meter, "stuck high" could result in an apparent loss of material in the downstream pipe, and an apparent gain of material in the upstream pipe. "Stuck low" generates apparent loss in the upstream pipe instead.

"Bias high" and "bias low" are separate root causes for the same reasons.

Bias faults are separate from "stuck" faults because the effects are different. This is especially the case when analyzing rate of change – you can notice recent changes even with a bias fault, but you cannot

when a “stuck” fault is present. Otherwise, they all have the same test and corrective action – a field calibration.

We took shortcuts for the ammeters in the demo. The ammeters and their root cause failure modes were not modeled. Instead, the “actual amps” events associated with the pumps were used, so that their values are only accessible by local field readings. There was enough SCADA data for the other variables that this local field testing wasn’t really needed for the key goal of classifying leaks versus other problems. In a real system, the ammeters would be included, because they can provide confirming data, especially needed if other data is inconsistent or missing.

For the density meters, their only fault was “recent inconsistency vs. sim”. This was a shortcut for demo purposes. The point has already been made about the importance of full sensor checking – for examples of the sort of checking needed in a real system, see the flow meters.

5.4.2.2 For pumps:

- Downstream blockage
- Upstream blockage
- Flashing material present
- inlet-pressure-low-NPSH (“interface” variable usually propagated from upstream for station leaks)
- Failed

In the fault model, “flashing material present”, and any faults relating to upstream blockage or major upstream leaks lead to a conclusion “inlet pressure low NPSH” which leads to another intermediate conclusion “cavitation”. Any of these “root causes” could in reality be effects of other failures – they should be considered part of an interface for connecting two modular fault models. The need for these sorts of interface variables was discussed earlier. And also as discussed earlier, these root causes provide homes for a myriad of other faults related to equipment not modeled, such as valves and short piping near the pump inlet.

For Station 01 in the demo, the immediate upstream leak fault is associated with the station rather than the pump, as a natural home for the collection of piping and valves involved. In the case of the initial feed Station 00, “upstream blockage” and “major leak near inlet” could not be distinguished by data, because there is no flow meter until after the pump. So a more generic “feed supply problem” root cause is associated with Station 00. This could the fault of pipeline equipment, or the fault of the customer supplying the feed for the pipeline.

For applications beyond leak detection, additional failure modes would likely be included. For instance, there would be one or more failure modes that lead to loss of efficiency compared to the normal pump curve, and perhaps failures that lead to vibration if vibration is monitored.

5.4.2.3 For pipeline segments:

- New leak

- Sustained leak
- Batch misalignment (transient timing errors related to the time a batch of new material with different density is present)

A "new leak" is distinguished from a "sustained leak" because a "new leak" is diagnosed only detecting recent sudden changes. Monitoring transient variable changes (such as rate of change or detecting step changes) is more sensitive for initial leak detection than depending on the magnitude of the variables themselves.

However, that transient data analysis is also sensitive to noise and tuning problems. More importantly, that data is transient and will disappear over time. For instance, consider variables like flows and pressures jumping from one steady state to another due to a sudden leak. Transient analysis calculations could include calculating rate of change, detecting step changes, or looking for the presence of other events within a time window. But, after some time has elapsed, all these symptoms disappear (return to false). So we need a root cause fault as an early indicator of a leak, and another one that can persist after the initial transient symptoms disappear.

Both the "new leak" and "sustained leak" problems lead to an intermediate conclusion called "leak" used for most downstream propagation.

5.4.2.4 For stations:

- Leak between inlet and pump
- Leak between pump and mainline flow meter
- Leak between mainline flow meter and outlet
- SCADA failed
- Recent shutdown
- Recent startup

The latter 3 were defined as placeholders, but not actually used yet. The leaks were defined for different zones in the station, because the effects of each leak on various sensors are different.

5.4.2.5 For valves:

Valves failures are not included in the demonstration. At a minimum, they would have separate failure modes for stuck closed, stuck open, or stuck mid-range, because the effects are so different. For valves used in control, an additional mode related to unstable operation (due to valve positioner problems, for instance) might be needed.

5.4.3 Symptoms and intermediate events/conclusions

Symptoms are events such as alarms or indications of high/low variables values that lead to diagnosing root causes. Additionally, there are many intermediate conclusions that are defined. These are mainly defined for convenience and ease in modularizing a system. The intermediate conclusions often represent significant conditions. For instance, for pumps, cavitation is significant. However, it is not a

root cause. Cavitation is caused mainly by low suction (inlet) pressure. That low pressure in turn is caused by root causes such as a major leak at the inlet, flashing material present, and so on.

Similarly, there are convenience events such as “reads high vs. actual” for sensors. It has root causes “biased high” and “stuck high”. When building fault propagation models, it is simpler to use the “reads high vs. actual” when linking to all uses of that sensor, than worry about all the reasons the sensor might be reading high.

The events associated with various objects are shown in screen shots later. The events not identified as root causes above are symptoms and intermediate events.

5.4.4 Examples of problems defined for specific objects in the demo

The following screen images show the Line 66 problems defined for pipe 00, the mainline flow 6601Fmain, pump 6001Pump1, and Station 01.

Most of these screen images were taken during or after a diagnosis, so some status information is indicated by background color. Reddish colors indicate true, meaning “yes we have a problem”. Green shades are for false, meaning “OK”. Yellow indicates “possible root cause”, a “suspect” with status otherwise unknown. No color means unknown status and never under suspicion of failure, or that there is no ongoing diagnosis.

There are variations in the shades of red and green. The darkest colors indicate direct observations (“Assertions”). The somewhat lighter shades indicate inferred values (values that must have occurred in the past – upstream of observations in the fault model). The lightest shades indicate predictions – these are downstream of observations. We have slightly less confidence in the current values for predicted status because there might be time delays associated with the cause/effect links in the fault model.

The user is in “expert” (developer) mode, so in most of the following displays there are web links for modifying the model, unless cut off during the screen capture.

Associated Problems for L66Pipe00

At equipment or system: L66Pipe00 (Line 66 pipe segment 00)

Problems directly associated with L66Pipe00

Problem	Description
batch-misalignment-transient-gain	L66Pipe00 has a transient condition only appearing to be a gain, due to batch misalignment
batch-misalignment-transient-loss	L66Pipe00 has a transient condition only appearing to be a loss, due to batch misalignment
inlet flow sensor high	Line 66 Pipe 00 inlet flow used as input for the simulator is high vs actual
inlet flow sensor low	Line 66 Pipe 00 inlet flow used as input for the simulator is low vs actual
inlet pressure to sim high vs actual	Line 66 Pipe 00 inlet pressure used as input for the simulator is high vs actual
inlet pressure to sim low vs actual	Line 66 Pipe 00 inlet pressure used as input for the simulator is low vs actual
leak	Leak in Line 66 Pipe00
new-leak	Line 66 Pipe 00 new leak
recent leak estimate increase	L66Pipe00:recent leak estimate increase
sim outlet flow high vs actual	Line 66 Pipe 00 simulated outlet flow is high vs actual value
sim outlet flow low vs actual	Line 66 Pipe 00 simulated outlet flow is low vs actual value
sim outlet pressure high vs actual	Line 66 Pipe 00 simulated outlet pressure is high vs actual
sim outlet pressure low vs actual	Line 66 Pipe 00 simulated outlet pressure is low vs actual value.
sustained-leak	Line 66 Pipe 00 sustained leak

CREATE NEW ASSOCIATED PROBLEM FOR L66Pipe00...

Associated Problems for 6601Fmain (mainline flow)

Problems directly associated with 6601Fmain	
Problem	**Description**
actual value erratic	6601Fmain actual value is erratic
bad	6601Fmain status has been set as bad
bias estimate high	6601Fmain mainline flow sensor bias is estimated as high based on repeated data reconciliation over a long time period
bias estimate low	6601Fmain flow sensor bias is estimated as low based on repeated data reconciliation over a long time period
biased high	6601Fmain reading is biased high vs actual - reads too high vs actual on average, but generally within the range of the instrument. Field calibration check. Try automated tests first.
biased low	6601Fmain reading is biased low vs actual - reads too low on average, but generally within the range of the instrument. Field calibration. Try automated tests first
erratic	6601Fmain:erratic - excessive high frequency noise is present in the reading
flatlined	6601Fmain flatlined - unchanging or nearly unchanging
high vs sim	6601Fmain reads high versus the simulated value from the L66Pipe00 outlet
low vs sim	6601Fmain reading is low versus the simulated value at the outlet of L66Pipe00
near-zero	Line 66 Station 01 mainline flow reads at or near zero
noisy sensor fault	6601Fmain has failed in a mode causing excessive noise. Field check. Try automated tests first
reads high vs actual	6601Fmain:reads high vs actual. Field calibration check
reads low vs actual	6601Fmain:reads low vs actual. Field calibration check
stuck high	6601Fmain is stuck at a high value. Field check. Try automated tests first
stuck low	6601Fmain is stuck at a low value. Field check. Try automated tests first.

Associated Problems for 6601Pump1

Associated Problems for 6601Pump1

At equipment or system: 6601Pump1 (Line 66 initial station injection pump)

Problems directly associated with 6601Pump1

Problem	Description
amps erratic	6601Pump1 actual amps erratic (actual value, not measurement). Field reading. Try automated tests first
amps high	6601Pump1 actual amps high. Local field reading. Try automated tests first
amps low	6601Pump1 actual amps value is low, but not near zero. Local field reading. Try automated tests first
amps near 0	6601Pump1 actual amps at or near 0. Local field reading. Try automated tests first
cavitation	6601Pump1 cavitation - violently changing pressures, flows. Flashing liquid forms bubbles. Field test listens to pump, observes local gauges. Try automated tests first
discharge pressure low and erratic	6601Pump1 actual discharge pressure low and erratic. Excessive high frequency process variations are present. Field test. Try automated tests first
discharge pressure very high	6601Pump1:discharge pressure very high
discharge pressure very low	6601Pump1 actual discharge pressure is very low - at or near 0 - usually only seen if the pump has failed or there is a major leak. Field test. Try automated tests first
downstream blockage	6601Pump1:downstream blockage - blockage somewhere downstream. Full field pressure survey - may take a long time. Try other tests first.
failed	6601Pump1 failed -trip, broken impellor, etc. Field test. Try automated tests first
flashing material present	6601Pump1 inlet flow includes flashing material - with a low NPSH. The field test requiring a lab sample. Try other tests first.
flow near zero	6601Pump1 Line 66 Station 01 actual flow at or near 0. Field test. Try automated tests first
inlet pressure low NPSH	Line 66 Station 01 Pump1 actual inlet pressure is low enough to cause pump cavitation (low NPSH). Field check of local gauge. Try automated tests first
upstream blockage	6601Pump1 inlet flow has blockage somewhere upstream. Full field pressure survey - may take a long time. Try other tests first.

Associated problems for Station 01

Associated Problems for L66_Station01

At equipment or system: L66_Station01 (Intermediate station 01 for Line 66)

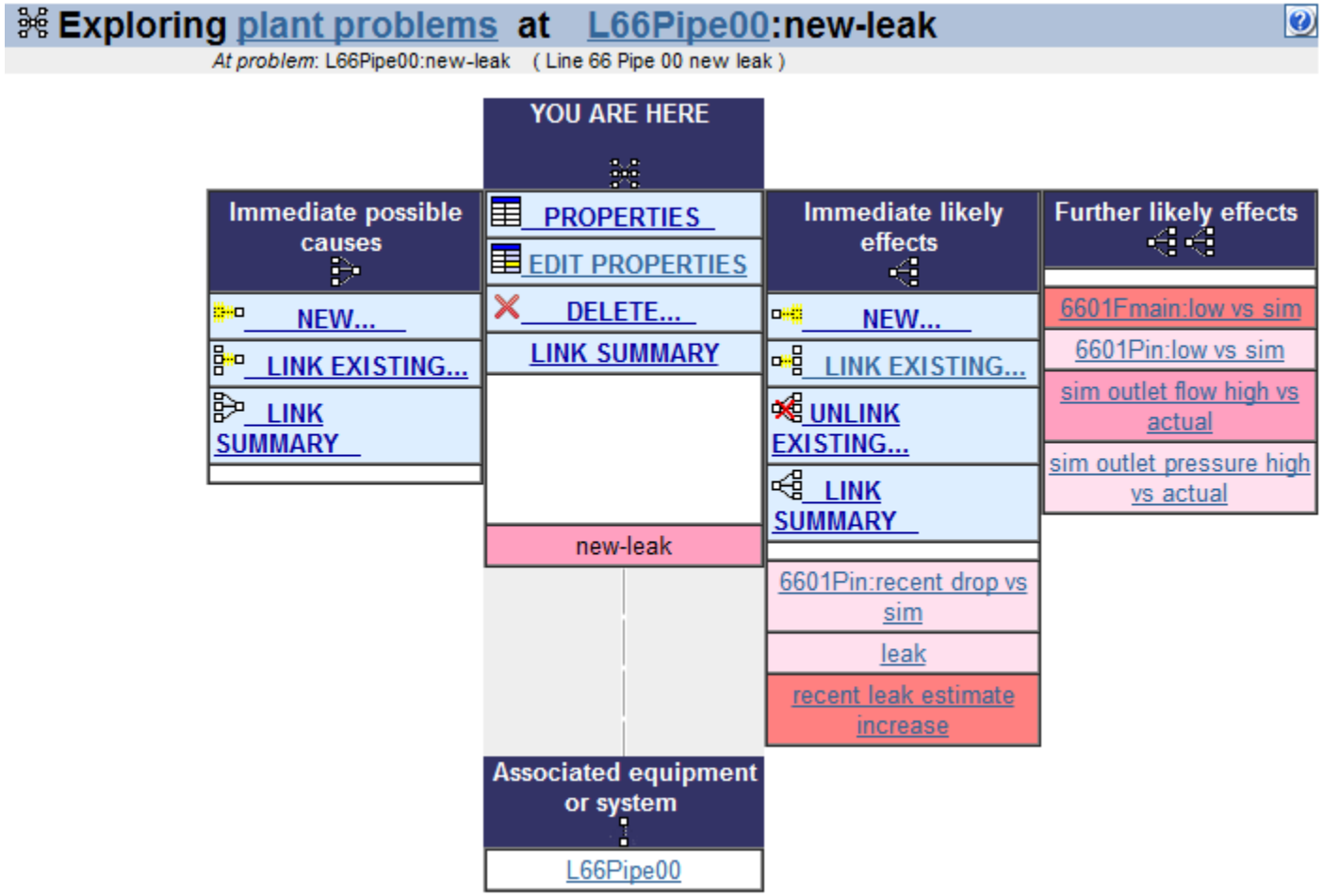
Problems directly associated with L66_Station01

Problem	Description
SCADA failed	Line 66 Station 01 SCADA failed
all sensors unavailable	Line 66 Station 01 all sensors are unavailable
leak between inlet and pump	Line 66 Station 01 leak somewhere between the inlet and the pump. Field check
leak between mainline flowmeter and out	Line 66 Station 01 leak somewhere between the mainline flowmeter and the station outlet. Field check.
leak between pump and mainline flowmeter	Line 66 Station 01 leak somewhere between the pump and the mainline flowmeter. Field check.
recent shutdown	Line 66 Station 01 recently shut down
recent startup	Line 66 Station 01 recently started up

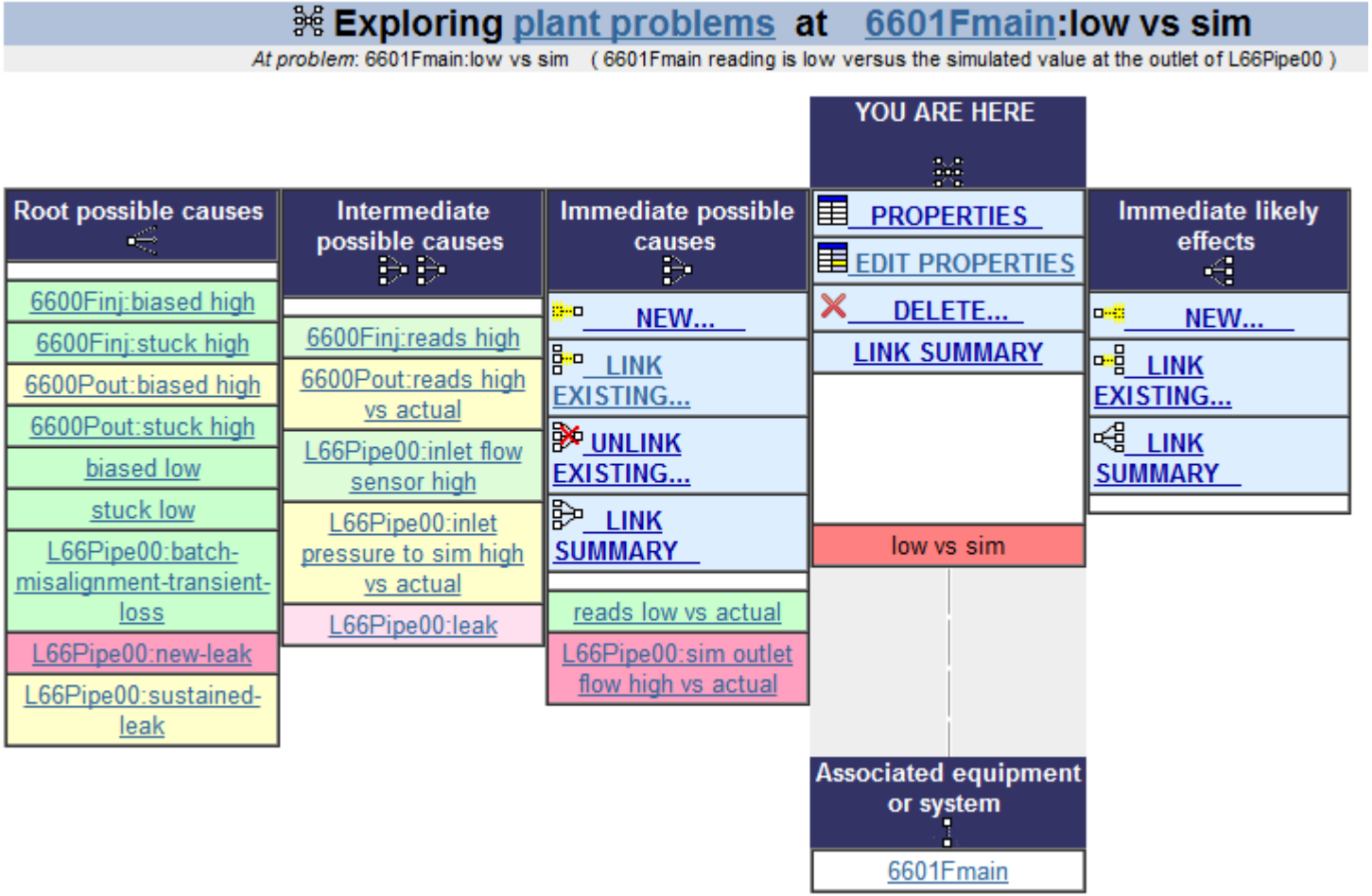
CREATE NEW ASSOCIATED PROBLEM FOR L66_Station01...

5.4.5 Navigating the fault model

You can use the "Graph Explorer" to trace the impacts of problems backwards or forwards through the causal links. The following screen image shows predictions of all the effects of new-leak fault. "YOU ARE HERE" indicates new-leak as the center of the display. The display drops the object L66Pipe00 part of the problem name when it is the same as the problem at the center of the display (e.g., as in "leak").



Conversely, you can see the possible causes of a problem. The following screen image shows possible causes for the true value of the flow vs. simulated value for 6601Fmain (the material loss alarm). This includes a complete list of all possible root causes on the left hand side. Again, since the user is in "expert" mode, there are web links to modify the possible causes and effects. The yellow status color indicates "suspects" that might be true, but not yet known (and possibly unknowable if they are masked by other faults).



5.4.6 Tests

Every problem can have an associated test. A test is a procedure directly associated with a problem, that determines if that problem is present (true), absent (false), or unknown. The simplest "tests" are just questions that can be answered through manual entry by a user. By default, the system generates a question, with the question text specified as an attribute of a problem. (If unspecified, if just defaults to the problem name followed by a question mark.) More generally, a test could also represent automated data collection. It could represent a request for data from another system. It also could represent the reception of events carrying a true/false/unknown status, whether requested or not.

For the pipeline demo, tests should be interpreted as requests for data from an external system, or as manual questions where that data would be unavailable.

5.4.7 Test level

Test (difficulty) level is an attribute of a problem. It indicates the difficulty or expense of directly testing for the existence of that problem. This could be an automated or manual test; ultimately answering the question "do you have this problem?" Bigger numbers indicate more difficult/expensive tests. Test level can factor in time, expense, skill level required, or whatever else is important in prioritizing the tests.

The test level guides the system in choosing which tests to request next. The "cheapest" tests are chosen first, so that other problems can be inferred without the need for direct testing. During diagnosis, you can set the maximum test level that you are willing to answer. The system supplies default test level descriptions, which may be overridden in applications.

The test levels specified for the pipeline demo are:

- Level –1: No test.
- Level 0: (Category only, used to automatically start diagnosis) - unused in this demo
- Level 1: Automated tests using SCADA data
- Level 2: Automated, transient analysis of SCADA data that will disappear over time, usually involving rate of change, and more sensitive to tuning errors
- Level 3: Field tests, hence leading to significant delays, cost, or risk of inaction
- Level 4: Difficult tests, requiring significant effort, field work, time delay, lab tests, expertise, customer interaction, or process impact

The following screen image shows the top part of a summary of the tests organized by test level for the pipeline demo:

Diagnostic tests by test difficulty level

Tests are questions answered true or false, or data requests. They are assigned a difficulty level called test level. This display organizes the tests in terms of test level. The higher the test level, the more effort or expertise is required to get the results. The numbers in parenthesis indicate the number of tests at a given test level.
Level 0: Category only, used to start diagnosis for non-automated diagnosis
Level 1: Automated tests using SCADA data
Level 2: Automated, transient analysis of SCADA data that will disappear over time, usually involving rate of change, and more sensitive to tuning errors
Level 3: Field tests, hence leading to significant delays, cost, or risk of inaction
Level 4: Difficult tests, requiring significant effort, field work, time delay, lab tests, expertise, customer interaction, or process impact

All tests (154)	Tests at level 0 (0)	Tests at level 1 (69)	Tests at level 2 (5)	Tests at level 3 (74)	Tests at level 4 (6)
6600Finj:bad		6600Finj:bad	6601Din:recent inconsistency vs sim	6600Finj:biased high	6600Pump1:downstream blockage
6600Finj:bias estimate high		6600Finj:bias estimate high	6601Pin:recent drop vs sim	6600Finj:biased low	6600Pump1:flashing material present
6600Finj:bias estimate low		6600Finj:bias estimate low	6602Ddel:recent inconsistency vs sim	6600Finj:noisy sensor fault	6600Pump1:upstream blockage
6600Finj:biased high		6600Finj:erratic	L66Pipe00:recent leak estimate increase	6600Finj:stuck high	6601Pump1:downstream blockage
6600Finj:biased low		6600Finj:flatlined	L66Pipe01:recent leak estimate increase	6600Finj:stuck low	6601Pump1:flashing material present
6600Finj:erratic		6600Finj:high vs sim		6600Pdisch:actual value erratic	6601Pump1:upstream blockage
6600Finj:flatlined		6600Finj:low vs sim		6600Pdisch:noisy sensor fault	
6600Finj:high vs sim		6600Finj:near-zero		6600Pinj:actual value erratic	
6600Finj:low vs sim		6600Pdisch:bad		6600Pinj:noisy sensor fault	
6600Finj:near-zero		6600Pdisch:erratic		6600Pout:actual value erratic	
6600Finj:noisy sensor fault		6600Pdisch:flatlined		6600Pout:biased high	
6600Finj:stuck high		6600Pdisch:very-high-value		6600Pout:biased low	
6600Finj:stuck low		6600Pdisch:very-low-value			
6600Pdisch:actual value erratic		6600Pinj:bad			

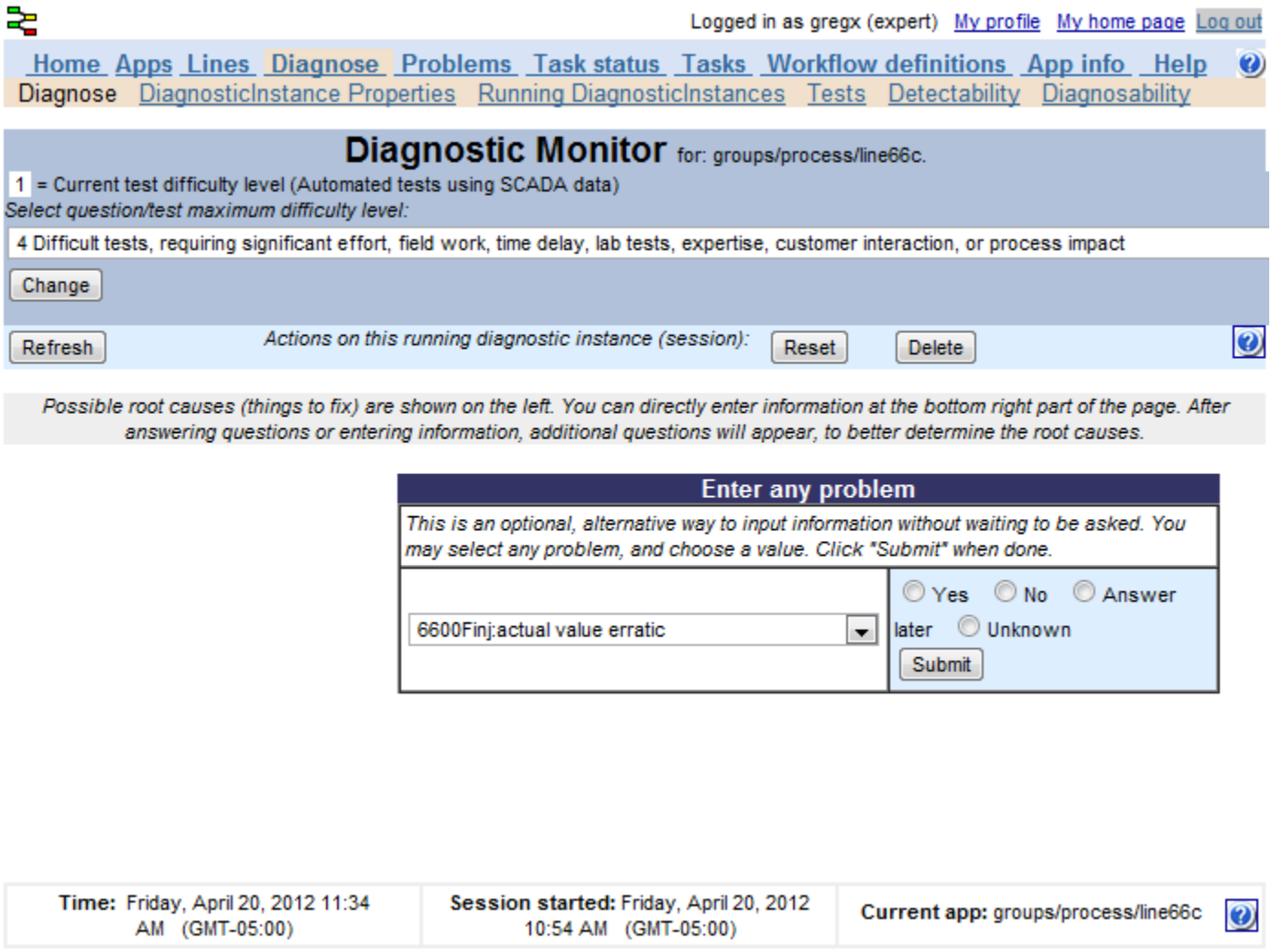
5.5 Diagnostic reasoning

The diagnostic system logic analyzes the events to arrive at a diagnosis. The basic approach taken is based on reasoning over the cause & effect models for events, as described in the "Guide to Fault Detection and Diagnosis" at

http://gregstanleyandassociates.com/whitepapers/FaultDiagnosis/Causal-Models/causal-models.htm

You could achieve similar diagnostic results through other approaches such as rules, logic gates, and so on. The demo is meant to focus on the elements needed in the fault models, rather than on the implementation.

You can input any event at any time to start or continue diagnosis, simulating receiving unsolicited events in any order when prototyping an online system. Alternatively, define a "level 0 test", for the system to automatically start by asking a particular question. Diagnosis and predictions can be initiated by selecting an event and value from the list box in the "Enter any problem" dialog as shown in the following screen image:



5.6 Test Planning

Test levels are significant in planning the tests to determine root causes. Most sensor problems can be tested by a field check (a direct check of equipment on site), but that's only a last resort for unattended sites. There could be a significant delay, so we do the best we can with SCADA data. However, even when a decision to shut down has to be made before the field check, the test requests should be useful for prioritizing the work following the shutdown, or for scheduling work for the next routine maintenance visit.

5.7 Detectability

When constructing complex models, it could be easy to make a mistake, or forget about some symptoms or the necessary test levels.

Detectability analysis checks to see if all defined root cause faults can detected given the set of tests that are defined. A problem is detectable for a given test level, if the occurrence of that problem leads to a change in an observable variable (symptom), using only tests at that difficulty level or easier. For instance, among all the symptoms affected by a root cause, suppose the only tests are at level 3 or higher. Then that root cause will be undetectable using just tests at level 1 or 2. But it will be detectable at level 3 and 4.

Since the system is based on causal models, this means there is a path of cause/effect links from the problem to some symptom with a test level less than or equal to the specified test level. This particular definition does not consider the effect of masking - it only guarantees that a problem can be detected if there are no other problems that mask it. So strictly speaking, it is fully applicable only in the case of a single fault assumption.

During fault model development, if a root cause is undetectable, either it should be removed from the model, or more symptoms with tests should be identified. Detectability is defined in terms of the test level because you often want to know if you can detect problems without using difficult/expensive tests.

This can point to the need for finding cheaper tests to detect particular root cause problems.

There is a display of undetectable root cause faults by test level. All faults in the demo are detectable at levels 1 and higher.

5.8 Diagnosability and ambiguity groups

Checking for diagnosability also helps in fault model development. Like detectability, it depends on the test level.

A root cause problem is diagnosable for a given test level, if that root cause can be isolated (uniquely determined) using only available tests at that test difficulty level or easier. Root causes that would remain in ambiguity groups after all tests have been run (at a given test level) are not diagnosable (at that test level).

Ambiguity groups are the result of diagnosis with inadequate diagnosability. They are sets of root cause problems with identical test results (for up to a given test level). So, given the test level, root causes within an ambiguity group can't be distinguished from each other without defining more tests/questions.

Diagnosability is defined in terms of the test level because you often want to know if you can diagnose problems without using difficult/expensive tests. This definition does not consider the effect of masking - it only guarantees that a root cause can be isolated if there are no other root causes that mask it. Strictly speaking, it is applicable only in the case of a single fault assumption.

The following screen image shows the top part of the page for diagnosability analysis by test difficulty level for the pipeline demo. For the pipeline demo, the analysis shows that some ambiguity groups are present at test level 1. For each pipe, we cannot distinguish the symptoms of a transient material imbalance from a station outlet pressure without resorting to level 2 tests. This isn't surprising, since the tests that deal with transient behaviour are defined at level 2. We don't have the same problem with flows, because we use the additional information available from data reconciliation. Luckily, this is not a problem, because we can derive the level 2 tests from SCADA data. We don't need to resort to local checks of conditions in the field.

Diagnosability analysis helps point to the need to combine root causes. Root causes are often aggregated when the effects are the same, and the tests and corrective actions are basically the same (e.g., sending someone to the field for a visual check for leaks). Sometimes root causes are not aggregated. An example is when there isn't enough data to distinguish between problems that must be checked or fixed by different organizations.

A diagnosability analysis for the demo is shown in the following screen image:

Diagnosability by test difficulty level (top of page)

Diagnosability by test difficulty level

Ambiguity groups are sets of root cause problems with identical test results (for up to a given test level).
So, given the test level, root causes within an ambiguity group can't be distinguished from each other without defining more tests/questions.

All root causes (91)	Ambiguity groups up through test level 1	Ambiguity groups up through test level 2	Ambiguity groups up through test level 3	Ambiguity group through test le
6600Finj:bad				
6600Finj:biased high				
6600Finj:biased low	Ambiguity group, size=2			
6600Finj:high vs sim	6600Pout:biased high			
6600Finj:low vs sim	L66Pipe00:batch-misalignment-transient-loss			
6600Finj:noisy sensor fault				
6600Finj:stuck high				
6600Finj:stuck low	Ambiguity group, size=2			
6600Pdisch:bad	L66Pipe00:new-leak			
6600Pdisch:flatlined	L66Pipe00:sustained-leak			
6600Pdisch:noisy sensor fault				
6600Pinj:bad	Ambiguity group, size=2			
6600Pinj:flatlined	6601Pout:biased high			
6600Pinj:noisy sensor fault	L66Pipe01:batch-misalignment-transient-loss			
6600Pout:bad				
6600Pout:biased high				
6600Pout:biased low	Ambiguity group, size=2			
6600Pout:noisy sensor fault	L66Pipe01:new-leak			
6600Pout:stuck high	L66Pipe01:sustained-leak			
6600Pout:stuck low				
6600Pump1:downstream blockage				

6 The platform on which the demo is built

The underlying platform is intended to support applications for diagnostics and workflow management. It is cloud-based (web-based) Java code running under the Google “App Engine”, so the interface is a web browser. (For workflows, the interface also includes e-mail and GoogleTalk, but that has not yet been extended to diagnosis).

The user interface for the diagnostics is oriented towards manual interactions. It can accept unsolicited arbitrary events from the set of possible events, and also has a test planner to ask questions once diagnosis is started. The underlying representation, diagnostic engine, and test planner are independent of the user interface and not necessarily restricted to manual interactions.

The underlying representation of the application is based mainly on directed graphs. For instance, the domain objects and their containment hierarchy form one directed graph. Similarly, the faults and symptoms and their causal relationships form another directed graph.

However, the user interface does not yet support construction and manipulation of these directed graphs directly through a drag-drop-connect graphical interface. Instead, alternate text-based displays are used. This was partly due to a desire to run on the simplest web browsers (including large smart phones), and mainly due to lack of time. The text-based “graph explorer” is a replacement for a full direct-manipulation graphical interface.

The platform is not yet available commercially.

7 Diagnosis examples

7.1 Example diagnostic view during diagnosis (top of page)

Please answer the new questions on the top right side of the page. Possible root causes (things to fix) are shown on the left. Also note the intermediate conclusions/working hypotheses below the questions. You can directly enter information at the bottom right part of the page. After answering questions or entering information, additional questions will appear, to better determine the root causes.

Top possible root causes - what to fix

Top possible faults (things to fix), arranged from most likely down to less likely. Numbers shown (probabilities) mean: 1 is the most likely, 0 is very unlikely. The order matters more than the numbers.

L66Pipe00:new-leak	0.25
L66Pipe00:batch-misalignment-transient-loss	0.25
L66Pipe00:sustained-leak	0.25
6600Pout:biased high	0.25

Full set of suspects, size=9

6600Finj:biased high	0
6600Finj:stuck high	0
6600Pout:biased high	0.25
6600Pout:stuck high	0
6601Fmain:biased low	0
6601Fmain:stuck low	0
L66Pipe00:batch-misalignment-transient-loss	0.25
L66Pipe00:new-leak	0.25
L66Pipe00:sustained-leak	0.25

New questions/tests

Answer any with "Yes" or "No" if you can. (Some answers will be predicted, so check each one carefully!)

Was 6601Din recently inconsistent vs its simulated value?	0.2653	Yes	No	Answer later	Unknown	Submit
6601Pin:recent drop vs sim?	0.25	Yes	No	Answer later	Unknown	Submit
L66Pipe00:recent leak estimate increase?	0.25	Yes	No	Answer later	Unknown	Submit

Top intermediate conclusions/working hypotheses (with probability estimate)

This helps you see what the system believes - working hypotheses that aren't directly testable.

L66Pipe00:sim outlet flow high vs actual?	1
L66Pipe00:leak?	0.5

Confirming questions/predictions (with expected answers)

These questions are redundant (unless they are conflicting, indicated by orange background for the probabilities). It helps to check the predicted values indicated below. Please change if appropriate.

6600Finj:biased high? (field test)	0	Yes	No	Unknown	Submit	(Difficulty: 3)
Is 6600Finj stuck at a high value? (field check)	0	Yes	No	Unknown	Submit	(Difficulty: 3)
Is 6600Finj stuck low (field test)?	0	Yes	No	Unknown	Submit	(Difficulty: 3)
6600Pout:stuck high? (field test)	0	Yes	No	Unknown	Submit	(Difficulty: 3)
Is 6600Pout stuck low? (field test)	0	Yes	No	Unknown	Submit	(Difficulty: 3)
6601Fmain:biased low (field check)?	0	Yes	No	Unknown	Submit	(Difficulty: 3)
6601Fmain:reads low vs actual? (field calibration						

7.2 Example diagnostic view during diagnosis (bottom of page)

Is 6601Fmain stuck low (field check)?	0	○ Yes	◉ No	○ Unknown	Submit	(Difficulty: 3)
6601Pin:low vs sim?	1	◉ Yes	○ No	○ Unknown	Submit	(Difficulty: 1)
6601Pout:biased low? (field test)	0	○ Yes	◉ No	○ Unknown	Submit	(Difficulty: 3)
6601Pout:stuck low? (field test)	0	○ Yes	◉ No	○ Unknown	Submit	(Difficulty: 3)
Is the 6602Fdel sensor bias estimated as high based on repeated data reconciliation over a long time period?	0	○ Yes	◉ No	○ Unknown	Submit	(Difficulty: 1)
6602Fdel:biased high? (field calibration)	0	○ Yes	◉ No	○ Unknown	Submit	(Difficulty: 3)
6602Fdel:reads high vs actual? (field calibration check)	0	○ Yes	◉ No	○ Unknown	Submit	(Difficulty: 3)

Questions already answered yes or no ("Assertions")

Questions you have already answered. You may change them at any time.

Is the 6600Finj sensor bias estimated as high based on repeated data reconciliation over a long time period?	0	○ Yes	◉ No	○ Answer later	○ Unknown	Submit
6600Finj:flatlined?	0	○ Yes	◉ No	○ Answer later	○ Unknown	Submit
6600Pout:flatlined?	0	○ Yes	◉ No	○ Answer later	○ Unknown	Submit
Is the 6601Fmain sensor bias estimated as low based on repeated data reconciliation over a long time period?	0	○ Yes	◉ No	○ Answer later	○ Unknown	Submit
6601Fmain:flatlined?	0	○ Yes	◉ No	○ Answer later	○ Unknown	Submit
Is 6601Fmain low versus the simulated value at the outlet of L66Pipe00?	1	◉ Yes	○ No	○ Answer later	○ Unknown	Submit
Is the 6602Fdel - Line 66 final station 02 delivery flow - high versus the simulated value at the L66Pipe01 outlet?	0	○ Yes	◉ No	○ Answer later	○ Unknown	Submit

7.3 Completion of a diagnosis of a new leak

Diagnosis is completed. The most likely faults are the ones with the highest probability, at the top of the 'possible root causes' list on the left (below). Please review the the confirming questions and their predicted answers (below). If you know that any of those answers are wrong, enter the correct answer and continue. Also, the following possible root cause(s) cannot be determined, because their failure would be masked by other predicted failures: L66Pipe00:sustained-leak, 6600Pout:biased high.

Possible root causes (things to fix) are shown on the left. Also note the intermediate conclusions/working hypotheses below the questions. You can directly enter information at the bottom right part of the page. After answering questions or entering information, additional questions will appear, to better determine the root causes.

Top possible root causes - what to fix	
Top possible faults (things to fix), arranged from most likely down to less likely. Numbers shown (probabilities) mean: 1 is the most likely, 0 is very unlikely. The order matters more than the numbers.	
L66Pipe00:new-leak	1

Full set of suspects, size=1	
L66Pipe00:new-leak	1

Top intermediate conclusions/working hypotheses (with probability estimate)		
This helps you see what the system believes - working hypotheses that aren't directly testable.		
L66Pipe00:leak?	1	
L66Pipe00:sim outlet flow high vs actual?	1	

Confirming questions/predictions (with expected answers)						
These questions are redundant (unless they are conflicting, indicated by orange background for the probabilities). It helps to check the predicted values indicated below. Please change if appropriate.						
6600Finj:biased high? (field test)	0	Yes	No	Unknown	Submit	(Difficulty: 3)
Is 6600Finj stuck at a high value? (field check)	0	Yes	No	Unknown	Submit	(Difficulty: 3)
Is 6600Finj stuck low (field test)?	0	Yes	No	Unknown	Submit	(Difficulty: 3)
6600Pout:stuck high? (field test)	0	Yes	No	Unknown	Submit	(Difficulty: 3)
Is 6600Pout stuck low? (field test)	0	Yes	No	Unknown	Submit	(Difficulty: 3)

7.4 Multiple faults

The following page illustrates a multiple fault situation. There is a leak at the first pipe, and a biased-low error for the delivery flow meter. Both are reported.

Diagnosis is completed. The most likely faults are the ones with the highest probability, at the top of the 'possible root causes' list on the left (below). Please review the the confirming questions and their predicted answers (below). If you know that any of those answers are wrong, enter the correct answer and continue. Also, the following possible root cause(s) cannot be determined, because their failure would be masked by other predicted failures: L66Pipe00:sustained-leak, 6600Pout:biased high.

Possible root causes (things to fix) are shown on the left. Also note the intermediate conclusions/working hypotheses below the questions. You can directly enter information at the bottom right part of the page. After answering questions or entering information, additional questions will appear, to better determine the root causes.

Top possible root causes - what to fix

Top possible faults (things to fix), arranged from most likely down to less likely. Numbers shown (probabilities) mean: 1 is the most likely, 0 is very unlikely. The order matters more than the numbers.

6602Fdel:biased low	1
L66Pipe00:new-leak	1

2 independent faults:

Full set of suspects, size=1

L66Pipe00:new-leak	1

Full set of suspects, size=9

6601Fmain:biased high	0
6601Fmain:stuck high	0
6601Pout:biased high	0
6601Pout:stuck high	0
6602Fdel:biased low	1
6602Fdel:stuck low	0
L66Pipe01:batch-misalignment-transient-loss	0
L66Pipe01:new-leak	0
L66Pipe01:sustained-leak	0

Top intermediate conclusions/working hypotheses (with probability estimate)

This helps you see what the system believes - working hypotheses that aren't directly testable.

L66Pipe00:sim outlet flow high vs actual?	1	
L66Pipe00:leak?	1	

Confirming questions/predictions (with expected answers)

These questions are redundant (unless they are conflicting, indicated by orange background for the probabilities). It helps to check the predicted values indicated below. Please change if appropriate.

Is the 6600Finj sensor bias estimated as low based on repeated data reconciliation over a long time period?	0	Yes / No / Unknown Submit	(Difficulty: 1)
6600Finj:biased high? (field test)	0	Yes / No / Unknown Submit	(Difficulty: 3)
6600Finj:biased low (field check)?	0	Yes / No / Unknown Submit	(Difficulty: 3)
Is 6600Finj stuck at a high value? (field check)	0	Yes / No / Unknown Submit	(Difficulty: 3)
Is 6600Finj stuck low (field test)?	0	Yes / No / Unknown Submit	(Difficulty: 3)
Is 6600Pout biased low? (field test)	0	Yes / No / Unknown Submit	(Difficulty: 3)
6600Pout:stuck high? (field test)	0	Yes / No / Unknown Submit	(Difficulty: 3)
Is 6600Pout stuck low? (field test)	0	Yes / No / Unknown Submit	(Difficulty: 3)
6601Fmain:biased high (field calibration)?	0	Yes / No / Unknown Submit	(Difficulty: 3)
6601Fmain:biased low (field check)?	0	Yes / No / Unknown Submit	(Difficulty: 3)
6601Fmain:reads high vs actual? (field calibration			

7.5 Conflicting data due to timing and threshold variations

In the following example, the initiating alarm was the low mainline flow versus its simulated value. A recent drop in the measured pressure 6601Pin versus its simulated value agreed with that, as did a recent increase in the estimated leak (based on recent flow imbalance). However, 6601Pin was not low versus the simulated value. That is, 6601Pin was dropping fast compared to simulation, but its absolute value had not yet crossed the low deviation threshold. So, the events were somewhat inconsistent.). Other root causes were ruled out by other tests, so there was only one root cause displayed, but it was now assigned a probability of about .53 instead of 1.0. The fact that there was conflicting evidence is emphasized by the orange background color for the probability estimates. This is immediately noticeable in the "Top Possible Root Causes", but also is highlighted in the "assertions" section (on the second page of the screen shot). Just the ones in orange are in conflict.

This is a fairly common situation, regardless of implementation in rules, causal models, or other systems. There are errors that could be related to timing. Timing errors have many possible origins, such as:

- Errors in the simulation model dynamics
- One variable having a little more filtering than another, so it takes longer to respond
- Inconsistent thresholds between variables or between variables and rate of change thresholds
- Difficulty of estimating rates of change or timing on changes
- One variable starting out much closer to its threshold than another when the fault occurs

In this case, it is likely that the symptoms will all agree soon, and continue to agree until the variables with "recent" in their title decide that the changes weren't recent enough. So, there is a window of time where the leak is most likely to be noticed.

In the meantime, the user still gets a warning that there is a suspected leak, with a lower probability.

Conflicting data part (top part of page)

Diagnosis is completed. The most likely faults are the ones with the highest probability, at the top of the 'possible root causes' list on the left (below). You may wish to increase the maximum test difficulty level (above) to see if more difficult tests can better pinpoint the root cause problem(s). Please review the the confirming questions and their predicted answers (below). If you know that any of those answers are wrong, enter the correct answer and continue. Some of the answers are not usually seen together. This may be due to information missing from this system, or may be due to an incorrect entry. This will result in more questions being asked, and lower probabilities for the answers. Diagnosis still provides a good ranked list of possible root causes, but please review questions already answered (on this page).

Possible root causes (things to fix) are shown on the left. Also note the intermediate conclusions/working hypotheses below the questions. You can directly enter information at the bottom right part of the page. After answering questions or entering information, additional questions will appear, to better determine the root causes.

Top possible root causes - what to fix

Top possible faults (things to fix), arranged from most likely down to less likely. Numbers shown (probabilities) mean: 1 is the most likely, 0 is very unlikely. The order matters more than the numbers.

L66Pipe00:new-leak	0.5278

Full set of suspects, size=1

L66Pipe00:new-leak	0.5278

Top intermediate conclusions/working hypotheses (with probability estimate)

This helps you see what the system believes - working hypotheses that aren't directly testable.

L66Pipe00:sim outlet flow high vs actual?	0.5278
L66Pipe00:leak?	0.5278

Confirming questions/predictions (with expected answers)

These questions are redundant (unless they are conflicting, indicated by orange background for the probabilities). It helps to check the predicted values indicated below. Please change if appropriate.

6600Finj:biased high? (field test)	0	Yes	No	Unknown	Submit	(Difficulty: 3)
Is 6600Finj stuck at a high value? (field check)	0	Yes	No	Unknown	Submit	(Difficulty: 3)
Is 6600Finj stuck low (field test)?	0	Yes	No	Unknown	Submit	(Difficulty: 3)
6600Pout:biased high? (field test)	0	Yes	No	Unknown	Submit	(Difficulty: 3)
6600Pout:stuck high? (field test)	0	Yes	No	Unknown	Submit	(Difficulty: 3)
Is 6600Pout stuck low? (field test)	0	Yes	No	Unknown	Submit	(Difficulty: 3)

Conflicting data (bottom part of page)

Questions already answered yes or no ("Assertions")		
Questions you have already answered. You may change them at any time.		
Is the 6600Finj sensor bias estimated as high based on repeated data reconciliation over a long time period?	0	○ Yes ◉ No ○ Answer later ○ Unknown Submit
6600Finj:flatlined?	0	○ Yes ◉ No ○ Answer later ○ Unknown Submit
6600Pout:flatlined?	0	○ Yes ◉ No ○ Answer later ○ Unknown Submit
Was 6601Din recently inconsistent vs its simulated value?	0	○ Yes ◉ No ○ Answer later ○ Unknown Submit
Is the 6601Fmain sensor bias estimated as low based on repeated data reconciliation over a long time period?	0	○ Yes ◉ No ○ Answer later ○ Unknown Submit
6601Fmain:flatlined?	0	○ Yes ◉ No ○ Answer later ○ Unknown Submit
Is 6601Fmain low versus the simulated value at the outlet of L66Pipe00?	0.5278	◉ Yes ○ No ○ Answer later ○ Unknown Submit
6601Pin:flatlined?	0	○ Yes ◉ No ○ Answer later ○ Unknown Submit
6601Pin:low vs sim?	0.5278	○ Yes ◉ No ○ Answer later ○ Unknown Submit
6601Pin:recent drop vs sim?	0.5278	◉ Yes ○ No ○ Answer later ○ Unknown Submit
Is the 6602Fdel - Line 66 final station 02 delivery flow - high versus the simulated value at the L66Pipe01 outlet?	0	○ Yes ◉ No ○ Answer later ○ Unknown Submit
L66Pipe00:recent leak estimate increase?	0.5278	◉ Yes ○ No ○ Answer later ○ Unknown Submit