NIST Industry Forum
08 May 2018

Crafting Intelligent Systems Management Using Requirements-Driven Design

Mark Walker D2K Technologies
mark.walker@d2ktech.com
Agenda

- Intro
- PHM Overview
- Requirements Driven Design
  - Definitions
  - Generic Approach
- Re-usable OO Platforms
  - Overview
  - Architecture
A Solutions company established in 2014, D2K utilizes:

- reliability centered design methodologies
- state of the art OO AI software development platforms (we love reuse!)
- agile software engineering for on-time delivery of validated software solutions

Focus: to leverage system model-based reasoning for delivering “Situation Aware” software. SA software is “thinking” software that encapsulates insight and understanding regarding operation, availability, uncertainty, and adaptation.

...software that can intelligently and autonomously monitor, control, emulate, execute, or optimize actions that will successfully ensure safe, timely, and dependable results.
PHM Systems are evolving to meet higher expectations

What should PHM Systems do?

- Determination of Health and its impact on system functions
- Monitor early warning of incipient failures
- Predictions of Remaining Useful Life
- Leveraging of advanced “reasoners”
  - Signal processing for event detection
  - Algorithms for event correlation and sensor fusion
  - Expert Systems and rule-based architectures
  - Advanced neural and statistical classifiers
  - Real-time state estimators
  - Model-based Reasoning
- Supervisory-level intelligence / logic
- Estimation and understanding of system state within operational context
- Decision support to assist operators in maintaining operational availability
- Optimize scheduling of maintenance and corrective actions according to the principals of condition-based maintenance
How have PHM Systems performed?

- Expensive
  - Takes too long to develop and deploy
- Often ill specified
  - Limited access to existing design data
  - Incomplete (or non-existing) design data
- Often an afterthought - considered very late in design cycle
  - Often reduced in scope
  - Or involving small incremental improvements to legacy systems
  - Or eliminated altogether
- Excruciating test and validation cycles
  - How to qualify PHM system?
  - No false positives / no false negatives
  - Test and validation using Simulation vs. historical data vs. supervised learning
- Questionable performance
  - Is system availability increased (downtime minimized)?
  - Is MTTR decreased?
  - Are operators better equipped and informed?
  - Are overall lifecycle costs reduced?
System Management Design Considerations

- Early as possible derivation of requirements
- Design based on functional requirements and the mitigation of failures (Behavior Driven Design)
- Need to link failures to detectable events across subsystems, and diagnosis to maintenance and corrective actions
- Design should identify necessary instrumentation (and consequences of inadequate instrumentation)
- Design should consider reasoning over systems, subsystems, predictive models, usage, operational regimes, real-time and historical data – within operational context
- Design should offer immediate advantages for life-cycle management
Output of Design Methodology

- Crisp set of System Management requirements according to operational context, functional requirements, and mission objectives

- Preliminary definition of critical failure modes, associated instrumentation and algorithms required to detect them, and downstream consequences (as well as the intra-subsystem event propagations that drive them)

- Baseline system object model required for reasoning

- Baseline fault models for diagnosis and prognostics

- Simulation and initial validation of diagnostic approaches and understanding of underlying event propagation
Quest for Software Quality

- **Test Driven Design (TDD)**
  - Write a test that fails
  - Code until it passes
  - Refactor (re-coding if it breaks)

- **Behavior Driven Design (BDD)**
  - “BDD is about implementing an application by describing its behavior from the perspective of its stakeholders”
  - Requirements as User Stories
  - Pull vs. Push based

- **Automated Testing using philosophy of jUnit, TestNG (example tools)**
  - Automated Report Generation
  - Tests follow system through life-cycle
Agile Development Process

1. Step 1: Preprocessor (cpp)
2. Step 2: Compiler (gcc, g++)
3. Step 3: Assembler (as)
4. Step 4: Linker (ld)

Waterfall development vs. Agile development

Value delivered vs. Risk of failure
Reliability Centered Maintenance Design

PHM Design Methodology – Part 1
Design Analysis and Asset Definition

- Reliability Analysis, FMEA, PRA

- Review content and determine if appropriate

- Tools and analyses should support PHM objectives.

- Drawings, specifications, schematics. Model to detail required by critical failure modes

Reliability Centered Maintenance Design

PHM Design Methodology – Part 2
Functional Failure Modes and Effects

- Functional Failure descriptions ensure that the PHM system detects what users care about
- Analyses may not provide insight into event propagation.
- Consider deterioration, increased expectation, and applied stress.
- What happens as a result of the failure?
- Take the time to consider event propagation. What evidence is available? Consider subsystem interaction
Monitoring, Diagnostics, and Prognostics for Manufacturing Operations

- Did something break? Is the system down? Did something spill? Does anyone get hurt?
- What could be done to avoid the consequences?
- How serious are the effects?
- What insight is there for defining event detection logic?
- Ready to do fault modeling
• Fault detection and isolation vs. Fault prediction

• Define the usage monitoring requirements and parameters

• Published, estimated, and derived statistical fault likelihoods

• Can failure rate be used as specified (and PM scheduled)?

• If possible, prognosticate

Selecting the Best Architecture

- Capabilities Abound

- How to Decide?
Traditional Control System Design

- Process Design
- Customer Requirements

Control System Reference Design

Control System Specification(s)

Design Review

Vendor Detailed Design

Validation & Acceptance Testing

Potential Risks:
- Technical Quality Issues
- Schedule Delivery Problems
- Cost Management Problems
- Incomplete Documents
- Difficult to make Modifications

Problems:
- Cost Management
- Schedule Delivery
- Technical Quality Issues
- Incomplete Documents
- Difficult to make Modifications
Model Driven Design Improvements

Process Design

Customer Requirements

Control System Reference Design

Model-based Design

Interactive Design Review

Vendor Detailed Design

Validation & Acceptance Testing

Plant Simulator

Correct Technical Solution

Reliable Schedule Delivery

Cost Effective Process

High-Quality Documents

Life-cycle Support

Process Design

Customer Requirements

Control System Reference Design

Model-based Design

Interactive Design Review

Vendor Detailed Design

Validation & Acceptance Testing

Plant Simulator

Correct Technical Solution

Reliable Schedule Delivery

Cost Effective Process

High-Quality Documents

Life-cycle Support
Goal: Transform data into information and knowledge based on operational context, leveraging all available wisdom.
Object Oriented Development Platform

HealthMAP

- Advisory Generation
- Prognostics
- Health Assessment
- State Detection
- Data Manipulation
- Virtual Intelligent Sensor Environment (VISE)
- Data Acquisition

External data and knowledge

- Advanced Simulation and Prediction
- Enterprise data
- Failure Modes Effects Analyses (FMEAs)
- Reporting/Archiving
- Business Process Policies
- Data Acquisition/SCADA systems

Reference: General Atomics
Standards-based Layered Architecture

- Models of Expected Process Behavior
- Empricial Health Determination
- Fault Detection and Root Cause Analysis
- Standalone System
- FM

LEGEND
- Input Layer
- Config Data
- Event Detection
- Fault Management
- Health Management
- Prognostics

- Validated Sensor Measurements
- Measured Process Behavior
- State Monitor
- State Commands

- Usage Monitoring
- Stress Detection
- Designed Stress Resistance

- User Entered Maintenance Events
- Maintenance Monitoring
- Maintenance Requirements

- Maintenance Advisories
- Maintenance Advisories

- Health Estimation
- Updated Fault Likelihoods
- A Priori Fault Likelihoods

- Health Prediction
- Consequence Prediction
- Consequence Advisories
- Mitigation

Anticipated Usage
Preferred Platform Characteristics

- **Reasoning Execution Engine**
  - Scheduling, simulation, inferencing, trending, state estimation, situational awareness, model-based reasoning, and multi-threaded processing

- **Integrated graphical modeling tools**
  - Domain representation, state transition, fault modeling, neural networks, workflow models, bow-tie diagrams

- **Methodology guided implementation using re-usable libraries**

- **OSA application supporting standards-based interfacing**
  - Transducers, DACs, PLCs, DCSs, SCADA, data aggregation platforms, 3rd party management tools, dynamic modeling and simulation platforms, enterprise data, Plant Historian, end user notification
Complex Generic Objects with Supporting Methods

- HM-VALVE, a class-definition
  - Methods
  - Procedures
  - Rules
  - User Interfaces
  - Supporting Resources

- HM-BALL-VALVE, a class-definition
- HM-CHECK-VALVE, a class-definition
- HM-MANUAL-VALVE, a class-definition
- HM-ACTUATED-VALVE, a class-definition
- HM-NEEDLE-VALVE, a class-definition
- HM-BUTTERFLY-VALVE, a class-definition
- HM-PNEUMATIC-MANUAL-VALVE, a class-definition
- HM-ROT-VALVE, a class-definition
- HM-SOLENOID-VALVE, a class-definition
- HM-CONTROLLED-VALVE, a class-definition
- HM-VARIABLE-POSITION-VALVE, a class-definition
- HM-MOTOR-VALVE, a class-definition
- HM-3WAY-SOLENOID-VALVE, a class-definition
- HM-3WAY-VALVE, a class-definition
- HM-GLOBE-THROTTLE-VALVE, a class-definition
- HM-ELECTRICALLY-ACTUATED-3WAY-VALVE, a class-definition
Extensible Model Libraries w/ Palettes
Domain Representations
Domain Representations
Relational Modeling Support

Object Model Classes

Water cooling system topology

Object relationships
Flow Subsystem as a Concept
Flow Subsystem 1: Members (TK1, pp1, T1, P1, pp2, pp3, V2, pp6, pp9, T3, P3, V5, T2, P2, F1, TK2), Source: TK1, Sink: TK2.
Flow Subsystem 1: Members (TK1, pp1, T1, P1, pp2, pp4, V3, pp7, pp9, T3, P3, V5, T2, P2, F1, TK2), Source: TK1, Sink: TK2.

Note: AO-MDS incorporates the concept of Flow Subsystem and dynamically determines Flow Subsystems for any application and its current configuration.

In Contrast with a data/information driven approach:
Flow subsystem selected from a pre-defined list that considers all possible combinations of valve configurations for all schematics
• generally hundreds or thousands of valves are involved, becoming a complex combinatorial problem.
• Any changes in the system (e.g. adding a valve) will require extensive work to update the combinatorial list.
• Any new system will require its own combinatorial list.
## Failures Modes and Effects Analysis (FMEA)

Modeling based on MIL-STD-1629A(2)

<table>
<thead>
<tr>
<th>ID #</th>
<th>Item-Functional Identification</th>
<th>Function</th>
<th>Failure Modes and Causes</th>
<th>Mission Phase-Operational Mode</th>
<th>Failure Effects</th>
<th>Failure Detection Method</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Process Equipment</td>
<td>Fluid feed subsystem</td>
<td>Leak</td>
<td>Sealed subsystem maintaining pressure</td>
<td>Pressure leak</td>
<td>Decreasing pressure measurement</td>
</tr>
</tbody>
</table>

---

![Diagram of Pressure Subsystem](image-url)
### Fault Detection, Diagnosis, and Prognosis

**Generic Fault Models** + **Domain Map** = **Specific Fault Model**

<table>
<thead>
<tr>
<th>Generic Fault Models</th>
<th>Domain Map</th>
<th>Specific Fault Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plugged inlet strainer</td>
<td><img src="image1" alt="Domain map 1" /></td>
<td><img src="image2" alt="Specific fault model 1" /></td>
</tr>
<tr>
<td>Low flow</td>
<td><img src="image3" alt="Domain map 2" /></td>
<td><img src="image4" alt="Specific fault model 2" /></td>
</tr>
<tr>
<td>Pump impeller damage</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Domain map 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Domain map 2</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Test and repair actions**
- Check inlet pressure pump
- Unclog inlet strainer pump
- Test and repair actions
Model based reasoning: event detection and diagnosis

Domain model used to predict expected values

Pressure drop detected in water cooling system

Sensor data

Context-specific causal model used for diagnosis

Valve leak

Low pressure downstream of leak

Degraded cooling

Degraded system
Debugging Specific Fault Models

Debug Specific Fault Models: Sequential Mode

<table>
<thead>
<tr>
<th>Type</th>
<th>Event Name</th>
<th>Target Object</th>
</tr>
</thead>
<tbody>
<tr>
<td>CurrEvt</td>
<td>Low flow</td>
<td>pump-2</td>
</tr>
<tr>
<td>PrevEvt</td>
<td>Clogged inlet strainer</td>
<td>pump-2</td>
</tr>
<tr>
<td>IncEvt</td>
<td>Low suction pressure</td>
<td>pump-2</td>
</tr>
</tbody>
</table>

Debug Status: INFER-EVENT
Steps Left: 6
Mode: Sequential


Debug Display Workspace

Debug Specific Fault Models: Event Log

<table>
<thead>
<tr>
<th>Number</th>
<th>Type</th>
<th>Event Name</th>
<th>Target Object</th>
<th>Value</th>
<th>Status</th>
<th>Time Stamp</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>InEvt</td>
<td>High outlet temperature</td>
<td>heater-1</td>
<td>true</td>
<td>specified</td>
<td>3/4/2007 22:56:02</td>
</tr>
<tr>
<td>3</td>
<td>InEvt</td>
<td>Impeller damage</td>
<td>pump-2</td>
<td>suspect</td>
<td>upstream inferred</td>
<td>3/4/2007 22:56:02</td>
</tr>
<tr>
<td>5</td>
<td>InEvt</td>
<td>Low flow</td>
<td>pump-1</td>
<td>suspect</td>
<td>upstream inferred</td>
<td>3/4/2007 22:56:02</td>
</tr>
<tr>
<td>6</td>
<td>InEvt</td>
<td>Impeller damage</td>
<td>pump-1</td>
<td>suspect</td>
<td>upstream inferred</td>
<td>3/4/2007 22:56:02</td>
</tr>
<tr>
<td>7</td>
<td>InEvt</td>
<td>Clogged inlet strainer</td>
<td>pump-1</td>
<td>suspect</td>
<td>upstream inferred</td>
<td>3/4/2007 22:56:02</td>
</tr>
<tr>
<td>8</td>
<td>InEvt</td>
<td>Tube fouling</td>
<td>heater-1</td>
<td>true</td>
<td>downstream inferred</td>
<td>3/4/2007 22:56:02</td>
</tr>
<tr>
<td>9</td>
<td>InEvt</td>
<td>Low outlet temperature</td>
<td>pump-1</td>
<td>suspect</td>
<td>downstream inferred</td>
<td>3/4/2007 22:56:02</td>
</tr>
<tr>
<td>10</td>
<td>InEvt</td>
<td>Low suction pressure</td>
<td>pump-2</td>
<td>suspect</td>
<td>downstream inferred</td>
<td>3/4/2007 22:56:02</td>
</tr>
</tbody>
</table>