IEEE P2848

Design review at 2024-01 SCC20 Meeting

Teradyne, North Reading, MA

2024-05-02

Legend

- Red: results of design review
- Green: remaining questions

Objective

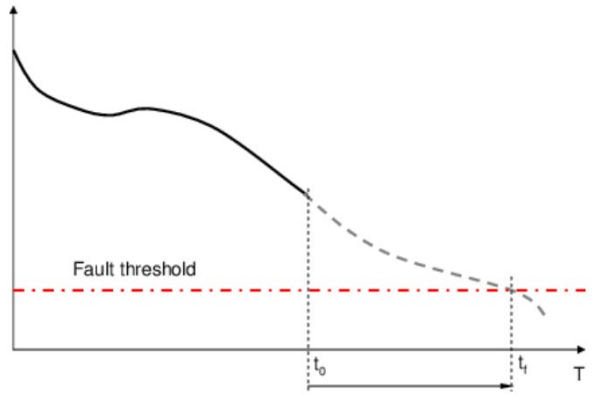
- Title: “Standard for Prognostics and Health Management in Automatic Test Systems”
- Two main use cases:
  - Prognose failures of UUTs, using test results produced by testing on ATE
  - Prognose failures of ATE components (instrumentation, switching, ITAs, …), using results of self-test and calibration tests
  - Observation: the accuracy of measurements on the UUT is influenced by the state of the ATE, so the UUT algorithms may need to also consider the state of the ATE; prognostics of the ATE could help, because it is degradation within tolerance.
- Create a normative standard that
  - Specifies the use of existing ATML and SIMICA component standards and of their elements
  - Specifies extensions to existing ATML and SIMICA schemas, for prognostic-specific data

Overview

- Assumes the existence of algorithms that can prognose failures using data collected during ATE testing
  - Does not attempt to specify the algorithms
  - Represents the algorithms as “(mostly) black box” **prognostic procedures**
- A prognostic procedure is assumed to have two **operating modes**:
  - Development & maturation – “tunes” the algorithm for a specific prognostic subject
  - Execution – runs the algorithm
- A prognostic procedure is assumed to have **parameters**
  - Calculated by development & maturation
  - Used by execution

Overview…

- Prognostic procedure, development & maturation
  - Prognostic subject:
    - Use case “UUT on ATE”
      - A UUT model that is tested on ATE
        - Optional: a specific UUT component, component failure mode, or function – see later slide
      - The ITA for that UUT
        - Optional: a specific ITA component, component failure mode, or function – see later slide
    - Use case “The ATE”
      - An instrument of the ITA
        - Optional: a specific instrument component, component failure mode, or function – see later slide
      - An ITA used for ITA self-test or calibration
        - Optional: a specific ITA component, component failure mode, or function – see later slide
  - Inputs:
    - Historic measurement results
    - Optional: Historic maintenance actions – ex. to determine when a component has failed and when it was replaced
      - To Do: add to concept diagram an indication that maintenance was performed as a result of running to failure
  - Outputs
    - Prognostic model parameters
      - Data series
      - Curve fitting (ex. polynomial, gaussian, exponential)
      - Other, defined through extension – see later slide



Overview…

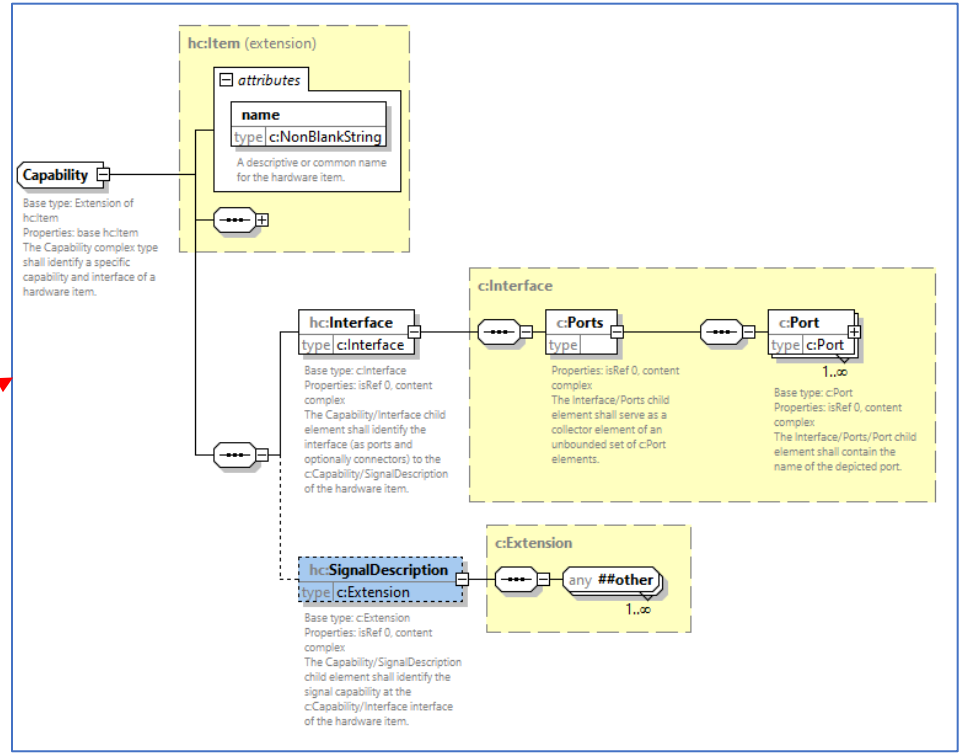
- Prognostic procedure, execution
  - Inputs:
    - Prognostic model parameters
    - Current measurement results
    - Optional: Past measurement results – ex. to determine trends
    - Optional: Past maintenance actions – ex. to determine when a component was last replaced
  - Outputs:
    - Prediction results for the subject:
      - Estimate that failure will occur within a preset time, with a specific confidence
      - Estimate that failure will occur before a specified time horizon, with a specific confidence
        - Could be multiple estimates, for different horizons (confidence increases as horizon moves farther) Note that “horizon” needs clarification. Erics sees it as the time to failure.
      - Estimated Remaining Useful Life (RUL), possibly with a confidence {this needs clarification}
      - Other, defined through extension – see later slide

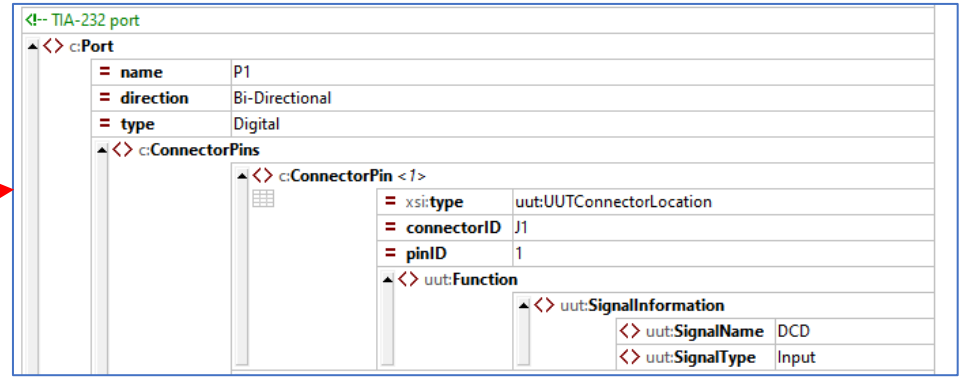
Design: Referenced Instance Documents

- Minimum requirements
  - Test Results
    - Store data analyzed by the prognostic procedures: current and past test results
    - Store prognostic results (through extension types)
  - Test Description
    - Describe prognostic procedures (through extension types)
    - Identify the prognostic targets
  - Applicable UUT Instance / Test Station Instance documents
- Optional
  - UUT Description
    - Identify the prognostic targets, in a document distinct from the Test Description
  - Test Adapter Description
    - Identify the prognostic targets, if ITA failures are being prognosed
  - MAI
    - Store data analyzed by the prognostic procedures: past maintenance actions [Chris] This should be a minimum requirement, along with Test Results; without that, trends in Test Results cannot be interpreted.

Design: Prognostic Subject

- References for UUT
  - **UUT Component**: reference a Component element by ID
  - **Failure mode of a UUT component**: reference a Fault element by ID
    - The Fault references a Component and specifies a Failure Mode
  - **UUT function, available at a UUT Port**: reference a Failure element by ID
    - The Failure references an Interface Port through an XPath attribute
    - ITA Component: reference the Test Adapter instance document by UUID and the Component (within the Test Station Description) by ID
  - **ITA Component** or **ITA Pin**: see next slide
  - Extensible by inheriting from abstract base type ***prog:AtsRefType***
  - Note that Component, Fault, and Failure elements can be specified in the Test Description instance document or a UUT Description instance document (if used)
- Alternatives for representing a prognosed UUT function (not shown in UML)
  - **Capability** elements - can have Signal Descriptions – Add this as a subject because it works better when extending the set of TPS measurements for the purpose of prognostics; is there a way to tie this to functional failures?
  - **Functions of Connector Pins**





Design: Prognostic Subject

- References for ATE:
  - **Instrument**: reference an Instrument (within the Test Station Description) by ID
  - **Instrument Component**: reference the Instrument by ID and the Component (within the Instrument Description) by ID
    - This also applies to switches within Switch instruments
  - **Instrument function** = Resource + Capability, available at Instrument Port: reference the Instrument by ID, the Resource (within the Instrument Description) by ID and the Capability (within the Instrument Description) by ID
  - **ATE Receiver Pin**: reference a Connector by ID and a Pin (descendant of a Port) by ID
  - **ITA Component**: reference the Test Adapter instance document by UUID and the Component (within the Test Adapter Description) by ID
  - **ITA Pin**: reference the Test Adapter instance document by UUID, a Connector (within the Test Adapter Description) by ID and a Pin (descendant of a Port, within the Test Adapter Description) by ID
  - Extensible by inheriting from abstract base type ***prog:AtsRefType***
  - Note: This design is different from that of the "UUT" use case because the Instrument Description schema does not allow the specification of Faults and Failures.
    - Instrument Failure Modes are not supported
    - Instrument functions at interface pins are described through Resource + Capability
  - This could change if support for Faults and Failures is added to Instrument Description (and Test Station Description for consistency). This addition would also help the ATE diagnostic use cases.
  - Chris: we could extend Instrument (ex. through xsi:type) also represent a UUT Description. That will be able to describe Faults and Failures.

Design: Prognostic Procedure Parameters

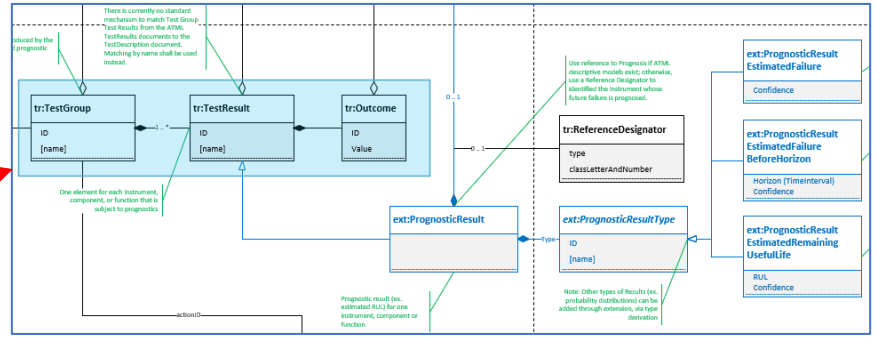
- Problem: Different classes of algorithms have different parameters
  - We are trying to remain algorithm-agnostic
  - At the same time, we want to offer some ready-to-use models
- Proposed solution:
  - Specify parameters for commonly used classes of prognostic algorithms {these need review by experts}
    - Data series
    - Curve fitting
      - Polynomial
      - Gaussian
      - Exponential
    - TBD
  - Provide extension mechanism
    - Abstract base type that can be inherited: ***prog:BehaviorType*** {is “behavior” a good term here? We are trying to identify the algorithm type / class}
    - For convenience: Derived type with a free-form description
    - For convenience: Derived type with a reference to an external document that contains the prognostic model (this may be an XML instance document or another type of document)

Design: Prognostic Procedure Results

- Problem: Different classes of algorithms have different results
  - We are trying to remain algorithm-agnostic
  - At the same time, we want to offer some ready-to-use models
- Proposed solution:
  - Specify results for commonly used classes of prognostic algorithms {these need review by experts}
    - Estimate that failure will occur (within a preset time), with a specific confidence
    - Estimate that failure will occur before a specified time, with a specific confidence
    - Estimated Remaining Useful Life (RUL), with a specific confidence
  - Provide extension mechanism
    - Abstract base type that can be inherited: ***prog:PrognosticResultType***
    - For convenience: Derived type with a free-form description
    - For convenience: Derived type with a reference to an external document that contains the prognostic model (this may be an XML instance document or another type of document)

Standard Contents

- For “use of existing ATML and SIMICA component standards”
  - UML diagrams – similar to those in Annex J of IEEE 1671
    - Will need to create multiple smaller diagrams, cross-referenced
    - Can make the large diagrams available as PDF downloads
  - Text describing the relationships:
    - Descriptive (statement of fact) for relationships between existing ATML and SIMICA entities {how detailed should this be? Would a UML diagram included in a normative clause be sufficient?}
    - Prescriptive (“shall”) for relationships between newly added Prognostic entities and existing ATML and SIMICA entities
- For “extensions to existing ATML and SIMICA schemas”
  - An XML schema with all the extension types
    - References the applicable ATML and SIMICA schemas {how to manage version changes?} In the text, reference the component standards without version, which means the latest version available. In the schema, reference current versions. Reference 2018 SIMICA version that uses ATML Common. A future amendment to reference revised ATML schemas will only impact the 2848 schema files.
  - Annotations converted automatically into normative text
- Conformance
  - Prescriptive (“shall”) for required components
  - Conditional prescriptive (“should” … “shall) for optional components:” – similar to that used to specify conformance in terms of ATML subframeworks, in Clause 11 of IEEE 1671



When the subject of a prognosis is a UUT component, the attribute prog:Prognosis/prog:TypeRef(xsi:type= prog:InstrumentComponentRef)/@componentId **shall** reference the attribute /td:TestDescription/td:UUT/ td:Description/c:Definition/hc:Components/ hc:Component/@ID identifying that component in the ATML Test Description instance document.

Instance documents conformant to the TestResults schema specified in IEEE 1636.1 **shall** be utilized to store the input test results for the prognostic procedure.

Should UUT maintenance data be provided as inputs to the prognostic procedure, instance documents conformant to the schema specified in IEEE 1636.2 **shall** be utilized to store these inputs.