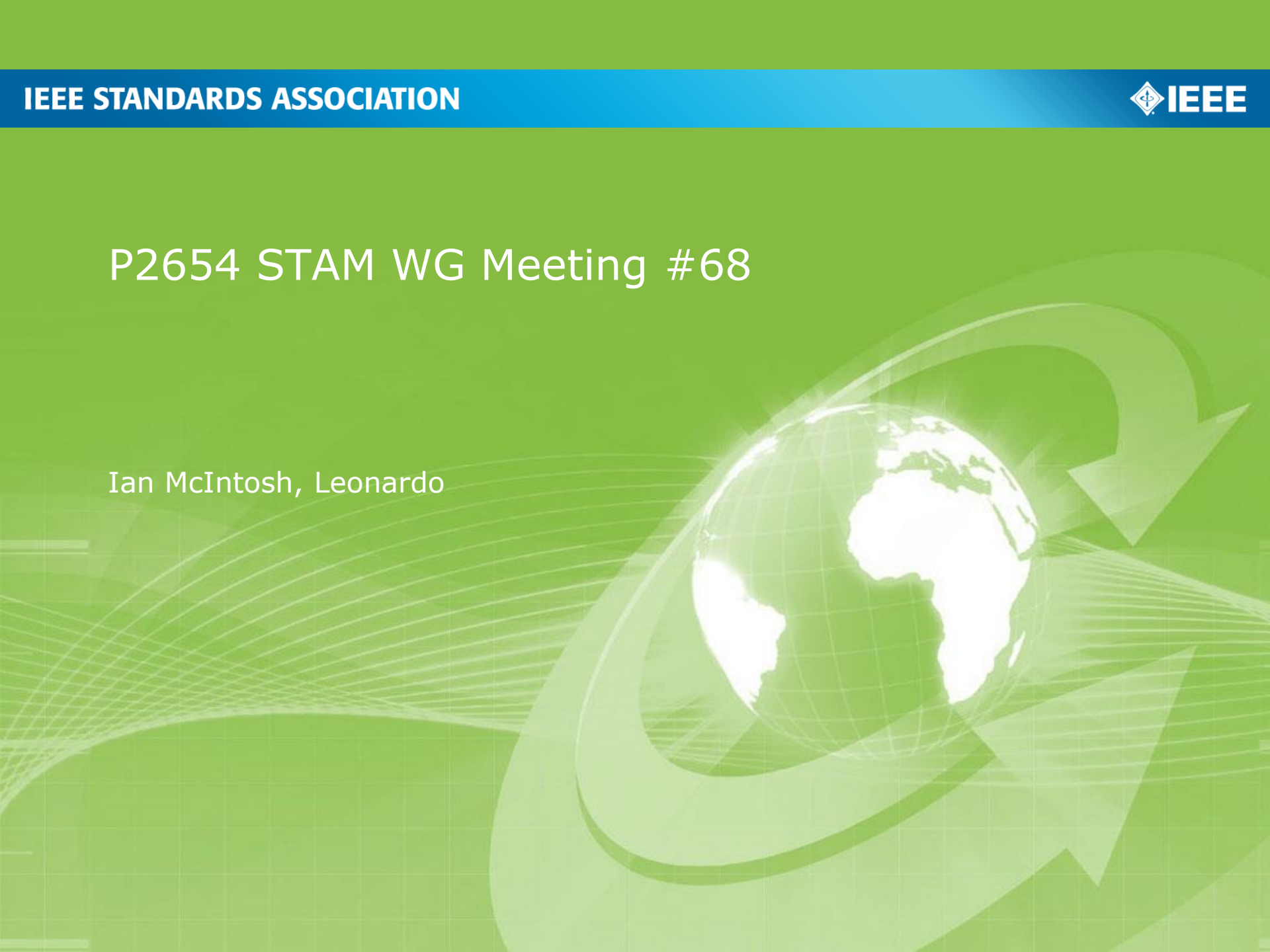
IEEE STANDARDS ASSOCIATION

IEEE

P2654 STAM WG Meeting #68

Ian McIntosh, Leonardo



Compliance with IEEE Standards Policies and Procedures

Subclause 5.2.1 of the *IEEE-SA Standards Board Bylaws* states, "While participating in IEEE standards development activities, all participants…shall act in accordance with all applicable laws (nation-based and international), the IEEE Code of Ethics, and with IEEE Standards policies and procedures."

The contributor acknowledges and accepts that this contribution is subject to

- The IEEE Standards copyright policy as stated in the *IEEE-SA Standards Board Bylaws*, section 7, http://standards.ieee.org/develop/policies/bylaws/sect6-7.html#7, and the *IEEE-SA Standards Board Operations Manual*, section 6.1, http://standards.ieee.org/develop/policies/opman/sect6.html
- The IEEE Standards patent policy as stated in the *IEEE-SA Standards Board Bylaws*, section 6, http://standards.ieee.org/guides/bylaws/sect6-7.html#6, and the *IEEE-SA Standards Board Operations Manual*, section 6.3, http://standards.ieee.org/develop/policies/opman/sect6.html

IEEE P2654
System Test Access Management
Ian McIntosh (chair)

Working Group Meeting #68			
Date: 2020-06-08			
Author(s):			
Name	**Affiliation**	**Phone [optional]**	**Email [optional]**
Ian McIntosh	Leonardo		

2. Agenda

1. Roll Call
2. Agenda
3. IEEE Patent and Copyright Slides
4. Review and approve previous minutes: June 1
5. Review open action items
6. Inter-group Collaboration
7. Discussion Topics:
    a. UTAP Development (Peter)
8. Any other business
9. Key Takeaways from today's meeting
10. Glossary terms from this meeting
11. Schedule next meeting
12. Topic for next meeting
13. Reminders
14. List new action items
15. Adjourn

Participants have a duty to inform the IEEE

- Participants shall inform the IEEE (or cause the IEEE to be informed) of the identity of each holder of any potential Essential Patent Claims of which they are personally aware if the claims are owned or controlled by the participant or the entity the participant is from, employed by, or otherwise represents

- Participants should inform the IEEE (or cause the IEEE to be informed) of the identity of any other holders of potential Essential Patent Claims

Early identification of holders of potential Essential Patent Claims is encouraged

Ways to inform IEEE

- **Cause an LOA to be submitted to the IEEE-SA (patcom@ieee.org); or**
- **Provide the chair of this group with the identity of the holder(s) of any and all such claims as soon as possible; or**
- **Speak up now and respond to this Call for Potentially Essential Patents**

If anyone in this meeting is personally aware of the holder of any patent claims that are potentially essential to implementation of the proposed standard(s) under consideration by this group and that are not already the subject of an Accepted Letter of Assurance, please respond at this time by providing relevant information to the WG Chair

Other guidelines for IEEE WG meetings

- All IEEE-SA standards meetings shall be conducted in compliance with all applicable laws, including antitrust and competition laws.
  - Don't discuss the interpretation, validity, or essentiality of patents/patent claims.
  - Don't discuss specific license rates, terms, or conditions.
    - Relative costs of different technical approaches that include relative costs of patent licensing terms may be discussed in standards development meetings.
      - Technical considerations remain the primary focus
  - Don't discuss or engage in the fixing of product prices, allocation of customers, or division of sales markets.
  - Don't discuss the status or substance of ongoing or threatened litigation.
  - Don't be silent if inappropriate topics are discussed … do formally object.

For more details, see *IEEE-SA Standards Board Operations Manual*, clause 5.3.10 and *Antitrust and Competition Policy: What You Need to Know* at http://standards.ieee.org/develop/policies/antitrust.pdf

Patent-related information

The patent policy and the procedures used to execute that policy are documented in the:

- *IEEE-SA Standards Board Bylaws*
  (http://standards.ieee.org/develop/policies/bylaws/sect6-7.html#6)
- *IEEE-SA Standards Board Operations Manual*
  (http://standards.ieee.org/develop/policies/opman/sect6.html#6.3)

Material about the patent policy is available at
http://standards.ieee.org/about/sasb/patcom/materials.html

If you have questions, contact the IEEE-SA Standards Board Patent Committee Administrator at patcom@ieee.org

IEEE-SA Copyright Policy

By participating in this activity, you agree to comply with the IEEE Code of Ethics, all applicable laws, and all IEEE policies and procedures including, but not limited to, the IEEE-SA Copyright Policy.

- Previously Published material (copyright assertion indicated) shall not be presented/submitted to the Working Group nor incorporated into a Working Group draft unless permission is granted.

  Prior to presentation or submission, you shall notify the Working Group Chair of previously Published material and should assist the Chair in obtaining copyright permission acceptable to IEEE-SA.
- For material that is not previously Published, IEEE is automatically granted a license to use any material that is presented or submitted.

IEEE-SA Copyright Policy

- The **IEEE-SA Copyright Policy** is described in the *IEEE-SA Standards Board Bylaws* and *IEEE-SA Standards Board Operations Manual*
  - IEEE-SA Copyright Policy, see
    Clause 7 of the *IEEE-SA Standards Board Bylaws*
    https://standards.ieee.org/about/policies/bylaws/sect6-7.html#7
    Clause 6.1 of the *IEEE-SA Standards Board Operations Manual*
    https://standards.ieee.org/about/policies/opman/sect6.html

- IEEE-SA Copyright Permission
  - https://standards.ieee.org/content/dam/ieee-standards/standards/web/documents/other/permissionltrs.zip
- IEEE-SA Copyright FAQs
  - http://standards.ieee.org/faqs/copyrights.html/

- IEEE-SA Best Practices for IEEE Standards Development
  - http://standards.ieee.org/develop/policies/best_practices_for_ieee_standards_development_051215.pdf

- Distribution of Draft Standards (see 6.1.3 of the SASB Operations Manual)
  - https://standards.ieee.org/about/policies/opman/sect6.html

4. Review and approve minutes

Working Group Meeting #67, June 1

Draft circulated June 1.

- Correction to Roll Call: Joel was a previously advised absence

Attendees:

Ian McIntosh (Leonardo)
Eric Cormack (DFT Solutions)
Heiko Ehrenberg (GOEPEL Electronics)
Brian Erickson (JTAG Technologies)
Peter Horwood (Digital Development Consultants Ltd)
Bill Huynh (Marvell Inc.)
Richard Pistor (Curtiss-Wright) (joined 11:09)
Jan Schat (NXP Semiconductors)
Jon Stewart (Dell)
Louis Ungar (A.T.E. Solutions)
Brad Van Treuren (VT Enterprises Consulting Services)
Carl Walker (Cisco Systems)

5. Review open action items

Action Item Register:

http://files.sjtag.org/PostStudyGroup/ActionItemRegister.xlsx

Format of action number is

[Meeting#.Action# within that meeting]

[38.1] Brad: Email brief instruction on using the simulation tool.

- Still ongoing.

6. Inter-group Collaboration

Nothing known prior to meeting.

7. Discussion Topics

7.a UTAP Development

- Presentation from Peter Horwood
- Live demo
  - May need follow-on (or ex-committee) session

Wrap-up items

8. Any other business
9. Today's Key Takeaways
10. Glossary terms from this meeting
11. Schedule next meeting
    - June 15
12. Topic for next meeting
    - Q&A for UTAP
13. Reminders
14. List new action items
15. Adjourn

January 6 Section Headings

DRAFT OUTLINE

Draft Outline

1. Overview
2. References
    1. Normative references
    2. Informative references
        1. IEEE Std. 1149.1-2013
        2. IEEE Std. 1687-2014
        3. IEEE Std. 1149.7-2009
3. Definitions
4. Concepts and Architecture/Technology (Small introduction of points/concepts)
    1. Introduction
    2. Relationship to other standards
    3. Architecture
    4. Interfaces
        1. Physical Layer (leveraged standards)
        2. Access and Data Link Layer
    5. Software Model
    6. Transformational Logic
    7. Retargeting
    8. Order of execution of commands
    9. Synchronization across interfaces
    10. Impediments
    11. Test Portability
    12. Security

Draft Outline (Continued 1)

5. Interfaces
    1. Physical Layer (leveraged standards)
        a) Formal standards (IEEE 1149.1, 1687, 1500, etc.)
        b) Proposed standards (IEEE P1687.1, P1687.2, P1149.7, etc.)
        c) Industry standards (I2C…, SPI, etc.)
        d) Ad hoc standards compliance via generic interface description (callbacks)
    2. Access and Data Link Layer
        a) Description of Access Link
        b) Description of Data Link
        c) Callbacks
        d) Retargeting for Access Link
        e) Illustration of concept
    3. Hand-off/Relation to other standards
        a) API vs. Streaming packet (function call vs. iSCSI message method) Why decision for method?
        b) Hand-off discussion
        c) Format of data messages (reference Software Data Model)
        d) Illustration of concept

Draft Outline (Continued 2)

6. Software Model
    a) Data Format shared between modules
        i. Detail the format used for messages between modules
    b) Circuit Description (how transformations interface)
        i. Something in the order of ICL and PDL as a descriptive language
    c) Transformation Description (the transform algorithm)
        i. How to get from point A to point B
            a. General transformation concept
            b. Retargeting impact
            c. Leveraging other standards as part of the transformation (e.g., JTAG to I2C Bridge where I2C Host is described with IEEE 1687)
    d) Structural Model
        i. How entities are wired together
            a. Netlist – Minimal topological information (interfaces) required (ex. Like what AI did with HSDL)
            b. Description of interfaces
            c. Etc.
        ii. Model components
            a. Transformation Maps
            b. Request Queues
            c. Response Queues
            d. Handlers (Callbacks)

Draft Outline (Continued 3)

7. Operational Overview
    a) Hierarchical Modules (Client to Host relationships)
        i. Describe a single module concept (like slides in presentations)
        ii. Describe role of client interface
        iii. Describe role of host interface
        iv. Software dependencies
            a. Host and Client Interfaces define the entry points to the module
            b. These interfaces have dependency on handlers being available to decode the messages through the interfaces
            c. The handlers have dependency on the model data for both configuration and state capture of hardware state for synchronization between software model and actual hardware entity
        v. Hint to description language format?
        vi. Definition of the P2654 host and client interface abstraction
            a. Message interface or do we use a class API to describe it (message with command code in header vs. a function/method described in the description language as the reference to what handler to call)
    b) Hierarchical Relationships
        i. Reliance on retargeting of host within bridge devices
        ii. Dynamic configuration of the path to open all AccessLinks to the target instrument. Do we want this to be automatic by the tooling or specified by the user as part of the algorithm?

Draft Outline (Continued 4)

8. System Concepts (More Detailed Descriptions)
    a) Assembly of Assemblies (vs discrete system, board, device, core)(connectivity is what matters)(Need to prevent pigeon hole one type from an aggregation – 16TB aggregate of 16-1TB memories)(Sub- reveals hierarchy)(Not using Sub- allows for pure aggregation and recursive nature relationship)(hierarchy vs. recursion)(Recursion – same thing/interface at various levels of the hierarchy)(SALT/NaCl – testing of SALT not Na and Cl separate necessarily)(STAM interested in stimulus and response not the function of the system. That is outside our scope.)
        i. Define what is common (attributes, behavior) in the assemblies part, but what is unique is the specialization of the assembly (e.g., system, board, device) defined in the specialized description
        ii. Form factor is different from connectivity. Need to define what is important and not just common attributes (e.g., addressing – geographical?, locality, instance)
        iii. What about distributed systems that may not be able to communicate all the time (wireless range, powered state)
    b) Spanning across physical boundaries (cable interfaces)
    c) Need simple diagrams of concepts
    d) Multi-drop vs. point to point vs. switch/router
    e) Available resources from sub-assembly? Discovery Activity
        i. Is this in STAM scope? Brought up multiple times.
        ii. Do we not support systems with subassemblies? We need discovery for sub-assemblies.
        iii. Is this an extended discovery feature: Dynamic query vs. static description in model. Absolutely need to have static discovery.
        iv. This is a discriminator between P1687.1 and P2654. P1687.1 deals bottom up with PDL. P2654 has some top down discovery needs.
        v. How does this relate with retargeter? Manually (separate) or integrated.

Draft Outline (Continued 6)

8. System Concepts (More Detailed Descriptions)
   - f) System (Proposed Definition): - A system is a structured hierarchy of subsystems and components. It has three fundamental properties:
     - i. Boundaries, which include not just an interface but also hand-off between the entities it binds
     - ii. Structure, which is a collection of interrelated and/or interdependent parts that often includes a graded hierarchy, and
     - iii. Synergy, which implies that the whole is greater than the sum of its parts and that the system possesses emergent properties that are not divisible to its subsystems.
   - g) Terry (NI – view): System is a unit that has a brain or controller. If it has application software then it is an application system. It does not need to have application software to become a system. A server does have a controller, so it is a system.
     - i. Jan: Electric systems vs. Electronic system
     - ii. Ian: Something that has an input and an output with something that gets done in between. A lever could than become a system. We are looking at electronic system.

Draft Outline (Continued 7)

8. System Concepts (More Detailed Descriptions)
    h) Ian (Leonardo – view): Radar is a system. Test is for the entire radar for DVT not for everyone that ships. Routinely test LRU to ensure they meet the qualifications of interface and thus will interoperate with other units. However, it has antenna, processor, receiver, power supply, etc. We don't sell radars. We sell components of a system.
        i. Terry: processor is the brain
        ii. Ian: each box has a controller for that unit
        iii. Joel: Do we really care what we call a system? Aren't we also testing sub-systems?
        iv. Ian: It comes down to what the user of STAM is desiring to do with it is what they care about.
        v. Joel: The key is then communications across interfaces is what is important.
        vi. Brad: Also, the transformation of what gets transmitted.
        vii. Louis: A system can make decisions. A memory card does not have control over what is written or read on it.
        viii. Ian: A system needs to have an awareness of its environment.
        ix. Bill: The control is really a state machine
        x. Joel: It can control and be controlled
        xi. A FLASH stick is an example of a non-system, but may be a sub-system/slave and not a master.
        xii. Ian: It is reactive instead of proactive.

Draft Outline (Continued 8)

8. System Concepts (More Detailed Descriptions)
   - i) Joel: USB PHY – Not a system, but a functional block. But this is what is tested as part of the module.
     - i. Jan: Is not USB PHY a sub-system?
     - ii. Ian: Perception of what constitutes a system depends on where you are looking at it in the overall hierarchy.
     - iii. Terry: It comes down to what you are trying to do as the function of your top level.
   - j) Ian: Individually, each board does not provide complete functionality on its own, but as an assembly they are able to perform the single function. A PCB is a practical decomposition and not always a functional decomposition. SRU is the module that gets shipped.

Draft Outline (Continued 9)

8. System Concepts (More Detailed Descriptions)
    k) Terry: Block diagram
        i) Parts and sub-parts
        j) Transformation functions
            i) Happens where the conversion to the other interface takes place
        k) Interfaces
            i) Terry: A tree structure does not capture the fact that communications could go between sub-assemblies. More a graph instead of a tree.
            j) Ian: Do alarms and simplistic flags represent this kind of communications we need to consider? This could be simpler than message based.
            k) Terry/Ian: Send a query to a system interface to tell the PSU to turn on its status outputs that can then be read as a simple state (analog or digital) and reported back as good or bad. Ian: It comes down to efficiency. At top level could send query and ask for analog value, but intermediate controller could just return a status whether it is good or bad to make the reporting more efficient. Ian: This matches what Louis was describing where a sub-system does not make decisions on its own, but responds to queries to provide information by some higher level. Brad: It is transforming the data into another form that is equivalent in representation for the purposes of the higher level. Ian: This is a different form of transformation from what we have talked about before with transformation. It is taking the information and moving it to a different form as a status instead of the same data. Brad: Is this transformation or translation? Louis: Need to describe transformation functions and translation functions.
            l) Ian: Transformation functions are reciprocal. Translations are not as it can be ambiguous going to the other direction.
            m) Terry/Brian: Translations are usually reciprocal and Transformation are not.
            n) Louis: Is translation a subset of transformation?
            o) Brad: Not really as in a transformation the data is preserved and nothing is lost. With translation, there is possibly loss of data in simplifying the data to a status.
            p) Joel: Lossless transformation vs. lossy transformation
            q) Peter: Encapsulation to different formats is what we were calling transformation. Consider retrieval of debug files and trying to decode their contents from the top level if not understanding the format of the file content at top level.
            r) Peter: May need to understand the use of a vector file at a different level.

Draft Outline (Continued 10)

8. System Concepts (More Detailed Descriptions)
    l) Brad shared forum discussion on UML to describe the blocks. SysML web site shown that was new to people. Ian uses SysML more for describing requirements. https://sysml.org/
    m) Ian: The reusable/generic block specification is what is important. Specialization is what makes it useful for that level. The generic/abstract is what makes it stick together.

Draft Outline (Continued 11)

8. System Concepts (More Detailed Descriptions)
    g) Available test targets
        i. Registers or Registers and 1687 ports (signals)
        ii. Overloading of term port: Port != port – Signal vs. I2C Address
        iii. IEEE 100-2000 Dictionary Port: port
            1) (1) (electronic devices or networks) A place of access to a device or network where energy may be supplied or withdrawn or where the device or network variables may be observed or measured. Notes:
                1. In any particular case, the ports are determined by the way the device is used and not by its structure alone.
                2. The terminal pair is a special case of a port.
                3. In the case of a waveguide or transmission line, a port is characterized by a specified mode of propagation and a specified reference plane.
                4. At each place of access, a separate port is assigned to each significant independent mode of propagation.
                5. In frequency changing systems, a separate port is also assigned to each significant independent frequency response.
            2) (10) An interface point connecting a communications channel and a device.
            3) (12) A conceptual point at which a cell or a hierarchical design unit makes its interface available to higher levels in the design hierarchy.
    h) Testability targets can be a feature. Realize functionality alone may not be ideal for a test application (e.g., BIST may require constraints, Signal generator may require a narrow range with higher resolution than factory test. BIST may only test a few frequencies that are not the same ones used in your application.)

Draft Outline (Continued 12)

~~9. Compliance~~/Conformance Concepts (Compliance Verification?) [Maybe move as last chapter before annex]

- ~~a) Degrees of~~ compliance vs compatible
- b) Compliance conforms to all essential rules of standard
- c) Compatible complies with some rules of standard to allow communications
- d) Must haves (mandatory), nice to have (recommendations, extension), can't do it(?)
- e) Black box support: Provisions for inputs that perform a particular response in the HW pins with stimulus
  - a) Discovery/declaration
  - b) Data in on digital interface – waveform on analog interface (example)
  - c) PDL like procedures supplied by the black box vendor
  - d) Discovery requires low level entity to supply something like a PDL procedure (e.g., by reference vs explicit)

Draft Outline (Continued 13)

10. Extended Concepts (How to plug into STAM)
    a) Perspective of view
    b) Leveraging lower level (standards or ad hoc)
        i. What an interface definition/specification needs to provide
        ii. Hand-off criteria
    c) Transformation reference between modules
        i. Data tracking (what bits are important)
        ii. Sequential vs. cached or queued processing of requests and responses (behavioral options)
    d) Hierarchies
        i. Scalability
        ii. Attributes – configurations?
        iii. Task assignments – Division of tasks
            i. High level – application
            ii. Single Test Sequence
            iii. Single Test Step
            iv. Instrumentation – Spectrum Analysis, Voltage Monitor, Power Supply, etc.
            v. Low level primitive
    e) Difference in stimulus/data
        i. Set up with digital stimulus and measure with analog output
        ii. Cluster tests (diagnostics inside cluster of logic accessible with stimulus and observation surrounding its boundary)
        iii. Tx/Rx loopback testing from DAC to ADC
            a. Requiring manual intervention to define stimulus and how to interpret observed

Draft Outline (Continued 14)

11. Advanced Concepts (Interesting but different use cases)
    a) System Interface (message based stream)
        i. STAM message packets over product communications interfaces
        ii. Queries of services available
        iii. Proxy (indirect test bus control via another interface) Interface
    b) System of Systems (complex perspective)
        i. How does this differ from 10.d Hierarchies
        ii. Racks of shelves of blades
        iii. Collections of LRUs
        iv. Interconnected by cable assemblies/harnesses
        v. Smart cables?
        vi. System (has a brain/CPU?)
        vii. System (able to operate autonomously)
        viii. Can we qualify what a system is in terms of being able to identify what can be replaced to make "System" operational again? How deep down do you go?
        ix. Repairable devices vs. soft repair of device (redundancy) – still need to mark as reduced availability

Draft Outline (Continued 15)

I. Education (Place holder about how to use and examples)
   a. Rule
   b. Permissions
   c. Preventative measures to ensure compliance
   d. Timing impact due to complexity of hierarchical interface

January 6 and 13 suggestions

CANDIDATE USE CASES

Use Cases

- TDR access between two end points
  - Interconnect Test between two end points
- SerDes Test between two end points
  - Same domain (IEEE Std. 1149.1-2013)
    - Device to Self
  - Different domain (IEEE Std. 1149.1-2013 and IEEE Std. 1687-2014)
    - Device to Device
    - Device to ATE
- JTAG to I2C Bridge
  - Protocol to Protocol transformation
  - IEEE 1687 or 1149.1 retargeting of I2C Host data
- System Test Message (via Ethernet stream interface)
- Host (top level) cause action in one sub-assembly causing action in another sub-assembly
  - Determining response in a different sub-assembly
  - Host asks for stimulus in one assembly that causes a reaction on another sub-assembly
- Top asks for temperature status (pass/fail). Sub-assembly asks for raw data to determine general status from other sub-assemblies.
- Need to be able to capture the vector representation at some level (e.g., COTS interface) to be able to pass to the vendor for further diagnosis (prune out dependency on rest of system being available). Isolate stream of data used. Visibility into representation.

Use Cases

- Concerns
  - Need to make sure it does not become a science project on it own
- Goals:
  - Show usage with 1687.1, 1687, 1149.1-2013
  - Show usage for interfaces outside of these standards and how AccessLink is able to accommodate these
- Support via annexes or illustration in body of standard?
  - Some illustration in body to convey concept
  - Specific illustrations/examples in annexes
- What is the difference between a use case and an example?
  - Use case drives to a feature
  - An example shows how to use a feature
  - 1687 uses examples
  - We are using use cases to drive what examples should be included
  - Use cases drive the rules for the standards
  - Use cases are the abstract problem

Recreating tests outside of the end-use hierarchy

- Vendor adding a value add by making the system test available to the vendor
  - Does vendor have other sub-assembly? May need stimulus from other sub-system to be able to recreate the test
- Goal is to have perspective from target sub-assembly instead of at the top. Need data into and out of the sub-assembly to be able to pass the test on to the third party vendor for diagnostic assistance.
- Not that vendor is recreating the entire test, but that the vendor is able to recreate the transactions occurring at their sub-system level. This is to resolve discrepancies where vendor claims it is not their problem and that of the test.
- Does not create entire test, but isolate portion of the test relevant to the target sub-system. Troubleshoot the sub-assembly itself instead compounding the complexity due to all of the intermediate transformations masking the real transaction.

Recreating tests outside of the end-use hierarchy

- Useful for determining test coverage (e.g., Load test code into on-board microcontroller to perform the test. Have software that is able to predict test coverage of the software running on microcontroller. Test coverage for each permutation.)
- Don't need full test capability, only the data stream for that target level. Tester agnostic if possible.
- Is this a core requirement of the standard or just an extension of it? There is a valid need from a debugging point of view. Case dependent for how you acquire the data. Value add by tool vendors?
- Brad's example of replaying the execution of a test off-line using captured stimulus from the embedded system by off-line debug station.
- Ian would prune down a test until as small a point where the failing point could be identified with a minimal test.

Recreating tests outside of the end-use hierarchy

- How can a tool vendor get the information from? Does some tool have the capability to apply the test and be able to preserve the information some place in memory.
- Joel suggests the model of the system is where the fabrication of the information happens and not in the real hardware. The data is then constructed from the model and not from the real hardware interface.
- This is probably an extension of the standard. However, Terry suggests there is always the case that the system test development does not work at first and requires some form of diagnostic at the levels, otherwise the standard is not real helpful.
- Are there scenarios with security where you cannot achieve the granularity of information. Yes, but need to acknowledge these constraints. Also true for cryptography.
- Indexed access to real data that sub-assembly uses internally to hide the detail of design of the black box. That may be sufficient. This is an important point for black box testing and better diagnostics.
- Example of Brad's plug 'n play example from ITC 2005.

Brainstorm on System

- Defining system in terms of characteristics is more useful than as physical form factors or similar domains
- Some form of hierarchy of parts
- System was different than sum of its component parts (e.g., PLL)
  - System has to be something beyond a primitive function
  - Needs to have some aspect of decision making to be concluded as a "System" – A controller present, the entity that receives instruction and sends out directives elsewhere
    - One of the attributes, but not only criteria
  - Not part of test access mechanism
  - Programmability – firmware does not change and the operation is strictly deterministic is not a system
    - What about ScanPathLinker? Behavior still is constant, but more complex than just pure JTAG TAP.
- System interactions need to be included in the discussion of the System definition

Brainstorm on System

- Then is a CPU a system? There is an aspect of power for it. Needs something else to be useful.
- Is a PXI Volt meter a system? Sub-system. It needs something else with it to be useful. This is an exception that lacks certain things to be able to be classified as a system
  - A system needs sub-systems
- ATE is a system.
- What is a part?
  - Could be a system as well
  - Provide some input, does some work, provides some output
  - Becomes a system when some higher level of control makes it

Brainstorm on System

- System (From forum)
  - Proposed Definition: - A system is a structured hierarchy of subsystems and components. It has three fundamental properties:
    - Boundaries, which include not just an interface but also hand-off between the entities it binds
    - Structure, which is a collection of interrelated and/or interdependent parts that often includes a graded hierarchy, and
    - Synergy, which implies that the whole is greater than the sum of its parts and that the system possesses emergent properties that are not divisible to its subsystems.
    - Need some aspect of control to be included as part of the definition

Discussion

- Jan sent email proposal for STAM System definition
  - Basic
  - More sophisticated
- Louis: Everything needs to be tested
  - Light bulb example
    - Bulb is not the system
    - Everything supporting the light bulb is the system
  - Jan system is something that is accessible
  - Louis: What portion is tested in a system (system part)?
  - Parts may be critical enough
  - Jan: A system is an entity containing a least one physical part that needs to be tested or should be tested.
  - Ian: STAM needs to be able to provide a way to test the system.

Discussion

- Ian: If you have a way to access the system through one of the existing standards, does that exclude the testing of the system by STAM?
- Ian: IEEE 1149.1 can tell you how to access the pins to stimulate the IO, but does not tell you how to leverage these pins to control another device connected to these pins needing to be programmed. STAM may be able to describe the mechanism/algorithm as to how to perform the programming from a file.
- Jon: How to target a test instead of just flipping on a light bulb.
- Ian: Difference between systems that need to be tested and throw away systems. Is that a distinguishing aspect?
  - Targeting of test is determined by customer. Pass rates per number of samples for example.
- Jan: Characterize and not just go/no-go?
- Jon: STAM deals with the process of test more than with the test.
- Could be a correlation between STAM and system test.
- STAM describes the process encompassing all the aspects and facilities required to perform the test on the system.
  - Ensure your test system is not going to be compromised during test
  - Safeguards ensuring test system continues even when a test injects a fault

Discussion

- Need to focus on constrains required during test for the rest of the system. Very difficult thing to communicate.
- System Test and Diagnosis – Simpson and Sheppard
  - Louis read 2 paragraphs 1994 version
- Computer Dictionary – Sippl & Sippl 1974 (SAMS book)
  - Several definitions for system that may be relevant
    - System
    - System Element
    - System Resources
- Radio Shack Dictionary – Rudolf F. Graf 1978 (shared by Jon)
  - System
  - System Element
- STAM System definition will most likely be broader than that for 1687.1

Discussion (quote read by Louis during meeting)

From: System Test and Diagnosis, William R. Simpson, John Sheppard

Page 3

"The complexity of modern systems is putting new demands on system maintenance. Every system, whether airplane, radio, or computer, has a mission to perform. The primary goal of system maintenance is to keep the system available for that mission. When the system fails, the job of maintenance is to diagnose and repair the system as rapidly as possible to return the system to correct operation. But diagnosing failures in complex systems requires analyzing characteristics of that system in great detail.

How do you reconcile the need for rapid repair with the need for in-depth analysis? Fault-tolerant systems approach this problem by limiting the need for diagnosis and repair, identifying failures as they occur on line, and reconfiguring the system to maintain functionality. Those in field maintenance have also tried to provide system-level diagnosis, incorporating ad hoc procedures based on field expertise, but this process is independent of design and manufacturing for the most part."

Discussion (quote from Ian during meeting)

From Sippl & Sippl:

System:

1. An assembly of components united to by some form of regulated interaction to form an organized whole.
2. A collection of operations and procedures, men and machines, by which business activity is carried out.
3. Any purposeful organization of resources or elements.
4. A collection of operations and procedures united to accomplish a specific objective.
5. A devised and designed regular or special method or plan or methodology or procedure. The organization of hardware, software, and people for cooperative operation to complete a set of tasks for desired purposes.
6. Any regular or special method or plan or procedure.

20200518 Discussion

- Thermostat example
  - Heater and A/C are subservient to the HVAC system
  - Thermostat is a controller, but may not be a computer – it makes decisions
  - IoT allows control to be remoted
  - IoT allows for aggregation of systems to be able to compile system of systems that may be hierarchical or distributed
  - Is there a requirement that something needs to make a prioritization of actions to be able to define what is a system?
  - Thermostat – human makes decision on the intent of the decision as input to the system decision process
  - As a system, there is some level of control that makes up what becomes a system. But that may not be a system for someone else (someone's system may be a sub-system to someone else)
  - Important is the perspective of the user
  - This is what makes System element important
  - Decision: If there is no control, it is not a system.

7. Discussion Topics

7.a Add Controller aspect to 'System' definition’

- Proposed Definition: - A system is a structured hierarchy of subsystems and components. It has three fundamental properties:
  - Boundaries, which include not just an interface but also hand-off between the entities it binds
  - Structure, which is a collection of interrelated and/or interdependent parts that often includes a graded hierarchy, and
  - Synergy, which implies that the whole is greater than the sum of its parts and that the system possesses emergent properties that are not divisible to its subsystems.
  - Control, which implies control/decisions over System Element and System Resource operation is required to constitute what is or is not a System
  - However, a P2654 System does require interaction of data between a higher level and the system/sub-system
  - Need some aspect of control to be included as part of the definition
- Additionally (from three references in May 4 meeting):
  - System Element
  - System Resources

7.b Select other definitions from forum

- http://forums.sitag.org/viewforum.php?f=40